DeFlow: Decoupling Behavior-Prior Modeling and Value Maximization for Offline Policy Extraction

Zhancun Mu **Chi Zhang**
School of Intelligence Science and Technology
Peking University
muzhancun@stu.pku.edu.cn chizhang.cz@pku.edu.cn

Abstract

We present DeFlow, a decoupled offline RL framework for extracting high-value actions from a learned multi-step flow prior. Directly optimizing iterative generative policies typically requires backpropagation through ODE solvers, while shortcut policies trade off expressivity, inference cost, and policy-improvement objectives. DeFlow keeps the flow model as a behavior prior and trains a lightweight action-conditioned residual module for value improvement under an adaptive trust-region penalty. This design bypasses solver differentiation and regularizes the refined policy toward the learned behavior prior. Empirically, DeFlow is competitive with recent generative offline RL baselines, with the clearest gains on selected multimodal manipulation and offline-to-online settings.

1 Introduction

Offline reinforcement learning (RL) must improve a policy while avoiding actions that are weakly supported by the static dataset [Levine et al., 2020]. This becomes harder on large multimodal benchmarks such as OGBench [Park et al., 2025a], where standard unimodal actors often cannot represent the diverse behavior distribution. Diffusion and flow-matching policies [Lipman et al., 2023, Sohl-Dickstein et al., 2015] are attractive in this regime because their iterative sampling procedures can model complex action distributions.

However, extracting a high-value policy from these generative models presents a stark *expressivity-optimizability dilemma*. Directly differentiating the Q-objective through iterative generation requires solver backpropagation, which can be costly; alternative objectives such as Q-score matching avoid this full differentiation [Psenka et al., 2024]. Shortcut methods such as Flow Q-Learning (FQL) [Park et al., 2025b] and SORL [Espinosa-Dice et al., 2025] reduce this burden through one-step or shortcut parameterizations, sometimes together with inference-time scaling. These methods are strong baselines, but they also motivate a complementary question: can we keep a multi-step behavior prior intact and perform value improvement through a small local extraction module?

In this work, we study this decoupled extraction route. We introduce DeFlow, a framework that separates behavior modeling from value-guided policy refinement.

Specifically, DeFlow comprises two distinct components: (1) a **Flow Matching Policy** optimized via supervised behavior modeling; and (2) a lightweight **Action Refinement Module** that maximizes Q-values via deterministic policy gradients (DDPG [Lillicrap et al., 2016]). Instead of updating the flow model through the ODE solver, the refinement module learns an action-conditioned residual shift on the generated action (Figure 1**c**).

A critical challenge in this decoupled setting is discouraging large residual moves that exploit unreliable critic values far from the learned prior. We address this with an adaptive Lagrangian

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

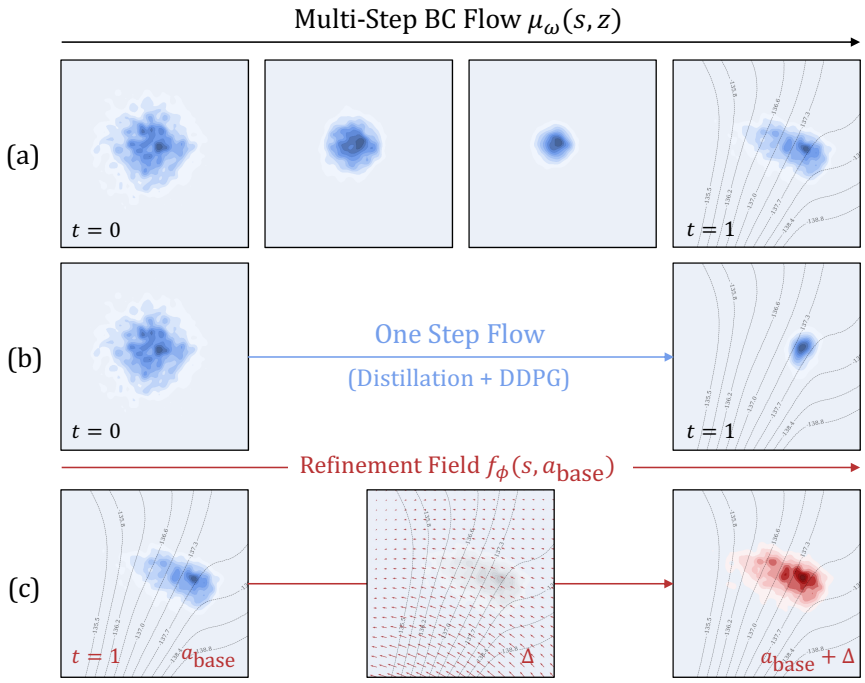


Figure 1: **DeFlow Framework. (a) Iterative Flow Prior:** A multi-step flow policy models multimodal behavior. **(b) Shortcut Extraction:** One-step distillation enables direct Q-optimization but may compress the generated action distribution. **(c) DeFlow:** We keep the iterative prior (Blue) and train a lightweight residual module (Orange) to perform local value improvement while penalizing large deviations from the prior.

penalty on the residual norm. The constraint should be interpreted as a trust-region surrogate around the learned flow prior, not as an exact guarantee of support preservation.

Some recent works [Xiao et al., 2025, Dong et al., 2025a] also decouple large expressive policies and small RL heads for policy improvement. DeFlow instantiates this pattern for offline policy extraction with a flow-matching prior and an instance-level residual conditioned on the sampled action.

Our contribution is the specific combination of a multi-step behavior prior, proposal-conditioned residual extraction, and adaptive residual regularization for offline RL. Flow matching, residual learning, dual adaptation, and BPTT avoidance are established ingredients rather than separate novelty claims. We study how this combination behaves relative to shortcut extraction, with benchmark comparisons and diagnostics of residual attribution, capacity, and budget sensitivity.

The results support a scoped claim: DeFlow is competitive with recent generative offline RL methods, while its strongest gains appear in selected multimodal manipulation tasks and in offline-to-online settings where the same residual extraction protocol is reused.

2 Preliminaries

We consider the standard Markov Decision Process (MDP) [Puterman, 2014] defined by the tuple $(\mathcal{S}, \mathcal{A}, r, p, \gamma)$, where $\mathcal{S}$ and $\mathcal{A}$ denote the state and action spaces, $r(s, a)$ the reward function, $p(s'|s, a)$ the transition dynamics, and $\gamma \in [0, 1)$ the discount factor.

2.1 Offline RL as Constrained Optimization

In offline RL, we are given a static dataset $\mathcal{D} = \{(s, a, r, s')\}$ collected by a behavior policy π_β. The goal is to learn a policy π_θ that maximizes the expected cumulative return. However, due to the distributional shift, simply maximizing values on out-of-distribution (OOD) actions leads to extrapolation errors. Following Levine et al. [2020], offline RL is often cast as a constrained

optimization problem:

$$\begin{aligned} &\max_{\pi_\theta} \mathbb{E}_{s\sim\mathcal{D}, a\sim\pi_\theta(\cdot|s)}[Q(s,a)] \\ &\text{s.t.} \quad \mathbb{D}(\pi_\theta(\cdot|s)||\pi_\beta(\cdot|s)) \leq \epsilon, \end{aligned} \tag{1}$$

where $Q(s,a)$ is the state-action value function and $\mathbb{D}$ is a divergence measure (e.g., KL divergence or Wasserstein distance) used to discourage unsupported policy shifts.

Critic Learning. To evaluate the policy, we learn a critic Q_ξ by minimizing the standard Bellman error [Mnih et al., 2013]. The objective is:

$$\mathcal{L}_Q(\xi) = \mathbb{E}_{(s,a,r,s')\sim\mathcal{D}}\Big[\big(Q_\xi(s,a) - r - \gamma\hat{Q}_{\bar{\xi}}(s',a')\big)^2\Big]. \tag{2}$$

where $a' \sim \pi_\theta(\cdot|s')$ and $\hat{Q}_{\bar{\xi}}$ utilizes a target network for stability. Critic parameters ξ, prior parameters ψ, and residual parameters ϕ are distinct.

The Actor's Dilemma. Standard methods typically convert the constrained problem in Eq. (1) into an unconstrained one via a Lagrangian relaxation with a fixed penalty coefficient α:

$$\mathcal{L}_\pi(\theta) = \mathbb{E}_{s\sim\mathcal{D}}\Big[-\mathbb{E}_{a\sim\pi_\theta}[Q_\xi(s,a)] + \alpha\,\mathcal{L}_{\mathrm{BC}}(\pi_\theta,\mathcal{D})\Big]. \tag{3}$$

Gaussian policies allow efficient reparameterized actor updates but have limited ability to represent multimodal behavior. Iterative generative policies offer greater expressivity, while direct differentiation through their sampling chains increases the optimization cost. Alternative training objectives can avoid this differentiation. A fixed regularization coefficient also introduces a task-dependent trade-off between value improvement and prior proximity [Park et al., 2025b, Agrawalla et al., 2025].

2.2 Conditional Flow Matching

To address the modeling challenge, we leverage Conditional Flow Matching (CFM) [Lipman et al., 2023], which offers a simulation-free alternative to diffusion training. CFM learns a time-dependent vector field $v_\psi(t,x) : [0,1]\times\mathbb{R}^d \to \mathbb{R}^d$ that pushes a simple prior distribution p_0 (e.g., $\mathcal{N}(0,I)$) to the data distribution p_1 (e.g., $\pi_\beta(a|s)$).

The generative process is defined by an Ordinary Differential Equation (ODE):

$$\frac{\mathrm{d}}{\mathrm{d}t}\Psi_t(x) = v_\psi(t,\Psi_t(x)), \quad \Psi_0(x) = x \sim p_0, \tag{4}$$

where Ψ_t is the flow map. We adopt the straight conditional path $x_t = (1-t)x_0 + tx_1$ [Lipman et al., 2024]. Noise and data endpoints are sampled independently; this does not assert an optimal coupling between the endpoint distributions. The vector field is trained via the regression objective:

$$\mathcal{L}_{\mathrm{FM}}(\psi) = \mathbb{E}_{t\sim\mathcal{U}[0,1],\, x_0\sim p_0,\, x_1\sim\mathcal{D}}\big\|v_\psi(t,x_t) - (x_1 - x_0)\big\|_2^2. \tag{5}$$

Flow as a Policy. In our context, the flow model serves as a state-conditional generator. We denote the generated action as a deterministic function of the initial noise z and state s:

$$a_{\text{flow}} = \mathrm{ODESolve}(v_\psi, s, z), \quad z \sim \mathcal{N}(0,I). \tag{6}$$

In our framework, this flow model acts as the learned behavior prior, decoupling distribution modeling from the policy improvement step described next.

3 DeFlow: Disentangling Fidelity and Optimality

DeFlow disentangles behavior modeling from value-guided extraction. Instead of training one generative policy with both behavior-cloning and Q-maximization gradients, we keep the multi-step flow as a proposal distribution and learn a small residual extractor on top of its samples. Algorithm 1 provides an overview of our method.

Algorithm 1 DeFlow

function $\mu_\psi(s, z)$
  for $m = 0, \dots, M-1$ **do**
    $z \leftarrow z + v_\psi(m/M, s, z)/M$
  return $\mathrm{clip}(z, -1, 1)$
function $\textsc{Act}(s)$
  $z \sim \mathcal{N}(0, I_d), \quad a_b \leftarrow \mu_\psi(s, z)$
  return $\mathrm{clip}(a_b + f_\phi(s, a_b), -1, 1)$
while not converged **do**
  Sample minibatch $\{(s, a, r, s')\} \sim \mathcal{D}$
  Update Q_ξ with Bellman targets using $a' = \textsc{Act}(s')$
  Update v_ψ on $\|v_\psi(t, s, x^t) - (x^1 - x^0)\|_2^2$
  $a_b \leftarrow \mathrm{sg}[\mu_\psi(s, z)], \quad \Delta \leftarrow f_\phi(s, a_b)$
  Update f_ϕ on $-Q_\xi^{\mathrm{norm}}(s, \mathrm{clip}(a_b + \Delta, -1, 1)) + \mathrm{sg}[\alpha]\|\Delta\|_2^2/d$
  Update $\eta = \log \alpha$ on $-\eta\ \mathrm{sg}(\|\Delta\|_2^2/d - \delta)$
return μ_ψ, f_ϕ

We formulate the policy as a composite function, where the final action a is derived from a **Base Flow** μ_ψ adjusted by a **Refinement Module** f_ϕ:

$$a = \underbrace{\mathrm{sg}[\mu_\psi(s, z)]}_{\text{Fixed Proposal}} + f_\phi(s, \underbrace{\mathrm{sg}[\mu_\psi(s, z)]}_{\text{Context}}), \quad z \sim \mathcal{N}(0, I) \tag{7}$$

Here, $\mathrm{sg}[\cdot]$ denotes the stop-gradient operator, and the displayed action is before clipping to the action bounds. The flow μ_ψ provides a behavior-prior proposal, while f_ϕ, implemented as an MLP, learns a residual shift toward higher critic values.

The role of Stop-Gradient: The stop-gradient operator cuts the computational graph between the refinement module and the flow model. During the update of f_ϕ, the base flow μ_ψ is treated as a fixed stochastic generator rather than a differentiable layer. This bypasses backpropagation through the ODE solver and keeps the two optimization loops separated.

3.1 Revisiting Existing Coupling Paradigms

To better contextualize DeFlow, we briefly analyze the limitations inherent in two prevalent strategies for integrating flow models with Q-learning.

Paradigm I: End-to-End Policy Steering. Ideally, one might update a flow model π_ψ by back-propagating the Q-gradient through the ODE integration chain:

$$\nabla_\psi \mathcal{L}_{\mathrm{RL}} = \mathbb{E}_{s,z}\left[\frac{\partial Q(s, a)}{\partial a} \cdot \frac{\partial \mathrm{ODESolve}(v_\psi, s, z)}{\partial \psi}\right]. \tag{8}$$

While conceptually direct, this "steering" approach requires BPTT or the Adjoint Method across numerous integration steps. This is memory-intensive and can suffer from unstable or weak gradients through the solver.

Paradigm II: Single-step Shortcut Distillation. Alternatively, methods like FQL [Park et al., 2025b] attempt to bypass the ODE complexity by distilling the iterative flow into a single-step policy $\pi_\theta(s, z)$. The actor is typically trained with a composite objective:

$$\mathcal{L}_\pi(\omega) = \underbrace{\mathbb{E}_{s,a^\pi}[-Q_\xi(s, a^\pi)]}_{\texttt{Q maximization}} + \underbrace{\alpha \mathcal{L}_{\mathrm{Distill}}(\omega)}_{\texttt{BC regularization}}. \tag{9}$$

While efficient at inference, this paradigm can compress the complex, multimodal distribution represented by the original flow. Furthermore, the scalar α introduces a sensitive hyperparameter trade-off: a large α limits improvement, while a small α allows larger deviations from the behavior prior.

3.2 The DeFlow Approach

DeFlow addresses these challenges by maintaining the *Multi-step Flow* as a behavior prior while employing a *Residual Refinement* for optimization.

The Base Flow μ_ψ is updated solely via the Flow Matching objective $\mathcal{L}_{\text{FM}}(\psi)$, so Q-learning gradients do not directly alter the prior.

Inspired by trust-region methods, we formulate refinement using a fixed, task-level budget δ on the mean squared residual per action coordinate. Let d be the action dimension:

$$\max_{f_\phi} \mathbb{E}_{s,z}[Q(s, a_{\text{flow}} + \Delta a)] \quad \text{s.t.} \quad \mathbb{E}_{s,z}[\|\Delta a\|_2^2/d] \leq \delta, \tag{10}$$

where $\Delta a = f_\phi(s, a_{\text{flow}})$ and $a_{\text{flow}} = \text{sg}[\mu_\psi(s, z)]$. Here s is sampled from replay. The division by d matches the implementation's coordinate-averaged squared error; δ is a squared-displacement budget, not a per-action radius. We apply action clipping when evaluating the critic and executing the policy. From an optimal-transport view [Villani et al., 2009, Peyré and Cuturi, 2019], the residual map $T_s(a_b) = a_b + f_\phi(s, a_b)$ is a Monge-style transport from the learned flow prior to the refined policy. Under the coupling induced by sampling $a_b \sim \mu_\psi(\cdot|s)$ and mapping it through T_s, $\mathbb{E}\|\Delta a\|_2^2$ upper-bounds the squared W_2 distance to the learned prior; we state this formally in Section A.

Relation to residual policies and SORL. Classical residual policy learning [Silver et al., 2019] usually adds a state-dependent correction to a base controller. DeFlow instead uses an instance-level residual $f_\phi(s, a_b)$, allowing different sampled actions from the same state to receive different local corrections. SORL [Espinosa-Dice et al., 2025] is an important step-size-conditioned shortcut-flow baseline: its actor is trained with flow matching, shortcut-consistency, and Q-maximization losses, and its evaluation can sweep shortcut step counts with optional critic-ranked candidates. We include it in the main benchmark, while DeFlow focuses on local residual extraction around a prior held fixed during each actor update. The prior continues to receive its separate behavior-modeling updates. Action conditioning is not a mode classifier: different proposals can receive different corrections, which may be mode-specific when modes are separated in action space.

To solve Equation (10), we employ the method of Lagrange multipliers. We first apply **Q-Normalization** to the RL objective to mitigate the scale variance of Q-values across different environments. We then introduce a learnable Lagrangian multiplier α (where $\alpha > 0$). The loss function for the refinement module f_ϕ is defined as:

$$\mathcal{L}_{\text{Refine}}(\phi) = \mathbb{E}_{s,z}\left[-\frac{Q(s, \text{clip}(a_{\text{flow}} + \Delta a, -1, 1))}{\text{sg}(\mathbb{E}_{\text{batch}}|Q|)} + \text{sg}[\alpha]\|\Delta a\|_2^2/d\right]. \tag{11}$$

We adapt the positive penalty $\alpha = \exp(\eta)$ using a log-space dual update. In the implementation, the loss minimized with respect to η is:

$$\mathcal{L}_{\text{dual}}(\eta) = \eta\left(\delta - \text{sg}\left[\mathbb{E}_{s,z}\|\Delta a\|_2^2/d\right]\right). \tag{12}$$

When the empirical residual exceeds δ, gradient descent increases η and hence α; otherwise it decreases the penalty. This resembles SAC's log-temperature adaptation [Haarnoja et al., 2018], but controls residual movement rather than entropy. It is a surrogate dual update, not a claim of solving the nonconvex constrained problem exactly. Adaptive α reduces fixed-coefficient tuning; δ still requires selection. Minibatch regularization neither enforces a hard pointwise bound nor guarantees true behavior-support membership.

4 Experiments

We evaluate DeFlow on 73 offline RL tasks and 15 offline-to-online (O2O) tasks. The goal is to test whether local residual extraction around a multi-step flow prior is competitive with recent generative offline RL methods, and where it helps or underperforms.

4.1 Experimental Setup

Benchmarks. We use OGBench [Park et al., 2025a] as the primary benchmark, including 50 state-based reward-conditioned `single-task` tasks and 5 visual manipulation tasks. We also evaluate on 18 D4RL tasks [Fu et al., 2020].

Table 1: **Offline RL results on OGBench and D4RL.** We report DeFlow against Gaussian-policy and flow-based/generative policy-extraction baselines. SORL results are included where aligned OGBench numbers are available; missing entries are marked by "-" and are not used when identifying row-wise best scores. Results are averaged over 8 seeds for state tasks and 4 seeds for visual tasks when applicable. Bold entries are at least $0.95\times$ the best reported score in their row. See Table 15 for the full task-level version of the non-SORL baselines. Counts in parentheses indicate the number of tasks.

	Gaussian Policies			Flow / Generative Extraction					
Task Category	BC	IQL	ReBRAC	FAWAC	FBRAC	IFQL	SORL	FQL	DeFlow
antmaze-large (5)	11 ±1	53 ±3	81 ±5	6 ±1	60 ±6	28 ±5	**89** ±2	79 ±3	81 ±3
antmaze-giant (5)	0 ±0	4 ±1	**26** ±8	0 ±0	4 ±4	3 ±2	9 ±6	9 ±6	12 ±5
humanoidmaze-medium (5)	2 ±1	33 ±2	22 ±8	19 ±1	38 ±5	60 ±14	**64** ±4	58 ±5	48 ±4
humanoidmaze-large (5)	1 ±0	2 ±1	2 ±1	0 ±0	2 ±0	**11** ±2	5 ±2	4 ±2	5 ±2
antsoccer-arena (5)	1 ±0	8 ±2	0 ±0	12 ±0	16 ±1	33 ±6	**69** ±2	60 ±2	**67** ±3
cube-single (5)	5 ±1	83 ±3	91 ±2	81 ±4	79 ±7	79 ±2	**97** ±1	**96** ±1	**96** ±2
cube-double (5)	2 ±1	7 ±1	12 ±1	5 ±2	15 ±3	14 ±3	25 ±3	29 ±2	**40** ±5
scene (5)	5 ±1	28 ±1	41 ±3	30 ±3	45 ±5	30 ±3	**57** ±2	**56** ±2	51 ±3
puzzle-3x3 (5)	2 ±0	9 ±1	21 ±1	6 ±2	14 ±4	19 ±1	–	30 ±1	**43** ±4
puzzle-4x4 (5)	0 ±0	7 ±1	14 ±1	1 ±0	13 ±1	**25** ±5	–	17 ±2	11 ±2
D4RL antmaze (6)	17	57	78	44 ±3	64 ±7	65 ±7	–	**84** ±3	**83** ±2
D4RL adroit (12)	48	53	**59**	48 ±1	50 ±2	52 ±1	–	52 ±1	52 ±1
Visual manipulation (5)	-	42 ±4	60 ±2	-	22 ±2	50 ±5	–	65 ±2	**69** ±2

[1] Pixel tasks selectively benchmark methods that are strong on state-based OGBench tasks.
[2] SORL numbers are taken from the aligned OGBench comparison used in DROL; SORL uses step-size-conditioned shortcut flows and inference-time shortcut-step scaling.

Baselines. We compare against Gaussian-policy baselines (BC, IQL [Kostrikov et al., 2022], ReBRAC [Tarasov et al., 2023]) and flow-based extraction baselines (FAWAC, FBRAC, IFQL, FQL [Park et al., 2025b]). We also include SORL [Espinosa-Dice et al., 2025] wherever aligned OGBench results are available. SORL is not an apples-to-apples matched-inference baseline because its shortcut-flow implementation conditions on step size and evaluates multiple shortcut step counts, with optional best-of-N critic selection; nevertheless, it is an important recent reference point for efficient generative offline RL.

Offline-to-online Experiments. For O2O experiments, we further compare against **Cal-QL** [Nakamoto et al., 2023] and **RLPD** [Ball et al., 2023], which are specifically engineered for online adaptation. We adapt DeFlow for this setting by simply augmenting the offline dataset $\mathcal{D}$ with an online replay buffer during training.

Evaluation Protocol. We follow the OGBench protocol [Park et al., 2025a], reporting performance after a fixed number of gradient steps. Results are averaged over 8 seeds for state-based tasks and 4 seeds for pixel-based tasks. We **bold** scores that are within 95% of the best reported performance in each row. The main results use a fixed four-layer residual; we do not select residual depth using the additional diagnostics. Their four-seed protocols and final-checkpoint evaluation budgets are reported separately in Section C.4.

4.2 Results and Analysis

Offline results. Table 1 shows a mixed but informative comparison. SORL is very strong on navigation and scene-style tasks, including `antmaze-large`, `humanoidmaze-medium`, `antsoccer`, and `scene`. DeFlow is most competitive on manipulation tasks where local multimodal precision matters: it improves over FQL and the reported SORL numbers on `cube-double`, and it remains strong on `puzzle-3x3` and visual manipulation. On D4RL, DeFlow is close to FQL on AntMaze and Adroit rather than clearly dominating. These results support a scoped claim: residual extraction around a multi-step prior is a useful alternative to shortcut extraction, especially for selected multimodal manipulation regimes.

Table 2: **Offline-to-online RL results.** The results are averaged over 8 seeds unless otherwise mentioned. Entries show offline → online performance; OGBench rows use the default single-task environment.

Task	IQL	ReBRAC	Cal-QL	RLPD	IFQL	FQL	DeFlow
humanoidmaze-medium	21 ± 13 → 16 ± 8	16 ± 20 → 1 ± 1	0 ± 0 → 0 ± 0	0 ± 0 → 8 ± 10	56 ± 35 → **82** ± 20	12 ± 7 → 22 ± 12	13 ± 5 → 65 ± 12
antsoccer-arena	2 ± 1 → 0 ± 0	0 ± 0 → 0 ± 0	0 ± 0 → 0 ± 0	0 ± 0 → 0 ± 0	26 ± 15 → 39 ± 10	28 ± 8 → **86** ± 5	44 ± 8 → **86** ± 3
cube-double	0 ± 1 → 0 ± 0	6 ± 5 → 28 ± 28	0 ± 0 → 0 ± 0	0 ± 0 → 0 ± 0	12 ± 9 → 40 ± 5	40 ± 11 → **92** ± 3	48 ± 13 → **93** ± 4
scene	14 ± 11 → 10 ± 9	55 ± 10 → **100** ± 0	1 ± 2 → 50 ± 53	0 ± 0 → **100** ± 0	0 ± 1 → 60 ± 39	82 ± 11 → **100** ± 1	58 ± 14 → **100** ± 1
puzzle-4x4	5 ± 2 → 1 ± 1	8 ± 4 → 14 ± 35	0 ± 0 → 0 ± 0	0 ± 0 → **100** ± 1	23 ± 6 → 19 ± 33	8 ± 3 → 38 ± 52	4 ± 4 → **100** ± 0
antmaze-umaze-v2	77 → **96**	98 → 75	77 → **100**	0 ± 0 → **98** ± 3	94 ± 5 → **96** ± 2	97 ± 2 → **99** ± 1	96 ± 3 → **99** ± 2
antmaze-umaze-diverse-v2	60 → 64	74 → **98**	32 → **98**	0 ± 0 → 94 ± 5	69 ± 20 → 93 ± 5	79 ± 16 → **100** ± 1	87 ± 6 → **99** ± 1
antmaze-medium-play-v2	72 → 90	88 → **98**	72 → **99**	0 ± 0 → **98** ± 2	52 ± 19 → 93 ± 2	77 ± 7 → **97** ± 2	76 ± 5 → **97** ± 2
antmaze-medium-diverse-v2	64 → 92	85 → **99**	62 → **98**	0 ± 0 → **97** ± 2	44 ± 26 → 89 ± 4	55 ± 19 → **97** ± 3	61 ± 9 → **98** ± 2
antmaze-large-play-v2	38 → 64	68 → 32	32 → **97**	0 ± 0 → **93** ± 5	64 ± 14 → 80 ± 5	66 ± 40 → 84 ± 30	76 ± 6 → **95** ± 4
antmaze-large-diverse-v2	27 → 64	67 → 72	44 → **92**	0 ± 0 → **94** ± 3	69 ± 6 → 86 ± 5	75 ± 24 → **94** ± 3	76 ± 9 → **96** ± 2
pen-cloned-v1	84 → 102	74 → 138	−3 → −3	3 ± 2 → 120 ± 10	77 ± 7 → 107 ± 10	53 ± 14 → **149** ± 6	64 ± 13 → **142** ± 7
door-cloned-v1	1 → 20	0 → **102**	−0 → −0	0 ± 0 → **102** ± 7	3 ± 2 → 50 ± 15	0 ± 0 → **102** ± 5	0 ± 0 → **99** ± 2
hammer-cloned-v1	1 → 57	7 → **125**	0 → 0	0 ± 0 → **128** ± 29	4 ± 2 → 60 ± 14	0 ± 0 → **127** ± 17	7 ± 3 → 106 ± 10
relocate-cloned-v1	0 → 0	1 → 7	−0 → −0	0 ± 0 → 2 ± 2	−0 ± 0 → 5 ± 3	0 ± 1 → **62** ± 8	**1** ± 0 → 35 ± 8

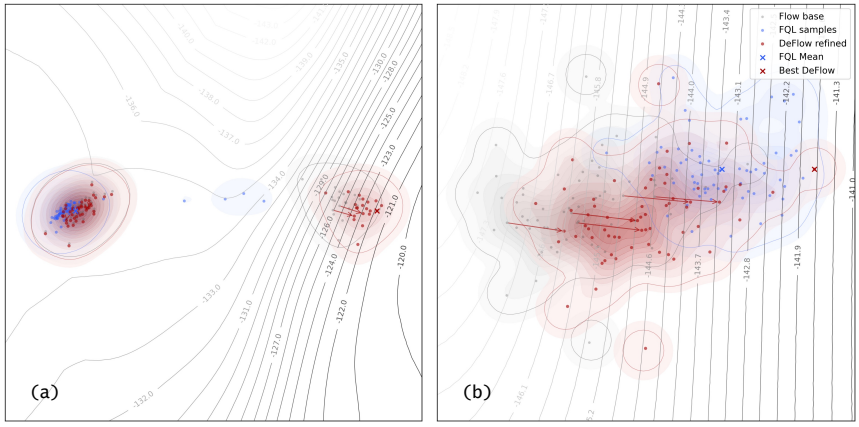


Figure 2: **Visualization of Action Landscapes.** Overlay of Q-value contours and action samples. **Gray**: Base Flow samples (data proxy); **Blue**: FQL (one-step); **Red**: DeFlow (ours). Arrows indicate the refinement trajectory. DeFlow performs local residual updates around base-flow proposals, illustrating smaller local shifts in this representative state.

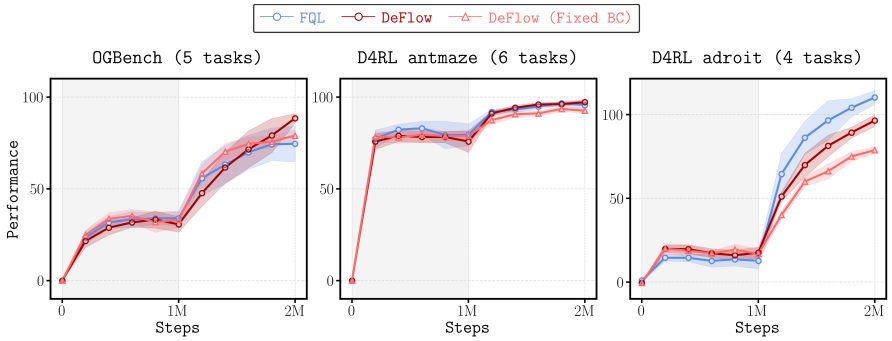


Figure 3: **Offline-to-online RL results (8 seeds).**

Mechanism visualization. Figure 2 visualizes critical states in `cube-double-play`. In these visualized states, FQL samples can fall between nearby modes or toward high-Q regions, while DeFlow starts from multi-step flow samples and applies a regularized residual. This diagnostic is not a formal support guarantee, but it illustrates the intended role of the residual penalty: keep value improvement local to the learned proposal distribution.

Offline-to-online and diagnostics. Table 2 and Figure 3 summarize O2O results. Using the same offline configuration, DeFlow is competitive with FQL on many tasks and is especially strong on `cube-double` and `puzzle-4x4`; it is weaker on some cloned Adroit tasks, highlighting dependence

on prior coverage and critic quality. Additional diagnostics in Sections C.1 and C.2 report δ sensitivity, an imbalanced human/expert dataset, action-conditioned refinement ablations, and compatibility with floq-style value modeling. Sections C.3 and C.4 add prior-only evaluation and residual-capacity sensitivity; Section D documents the runtime comparison.

4.3 Interpreting the Empirical Pattern

The SORL comparison changes how the results should be read: DeFlow is not uniformly stronger, and the table points to regimes where different extraction mechanisms are useful. SORL and shortcut-style methods can allocate extra evaluation compute across shortcut step counts or critic-ranked candidates, which fits several navigation and scene-style tasks. The strong SORL numbers on `antmaze-large`, `humanoidmaze-medium`, `antsoccer`, and `scene` are consistent with this view.

DeFlow is intended for a narrower regime: the learned prior already produces useful local modes, but value extraction must make small, sample-specific corrections without collapsing those modes into a single shortcut output. This is a possible explanation for the manipulation results, not an experimentally isolated cause. For example, `cube-double` and `puzzle-3x3` require choosing among nearby multimodal behaviors and then making precise local adjustments. A residual that is conditioned on both the state and the sampled action can move one proposal without forcing all proposals from the same state to share the same correction. This is also why the landscape visualization is included as a diagnostic rather than as a proof: it illustrates the intended local mechanism in a representative state, while the quantitative table determines where that mechanism is actually useful.

The O2O results provide a complementary stress test: we reuse the same residual architecture and budget mechanism while appending online transitions to replay. This transfers well when the offline prior already covers useful behaviors, as in `cube-double` and `puzzle-4x4`, but is weaker when prior coverage or critic quality is less reliable. The appendix diagnostics probe sensitivity and component attribution rather than serving as additional headline benchmarks.

Attribution and residual capacity. Disabling the residual at the same trained checkpoints gives 10 ± 2 versus 62 ± 12 with refinement on cube-double task1, and 0.5 ± 0.9 versus 35 ± 11 on puzzle-3x3 task2 (Table 10). These evaluations isolate the residual's inference-time contribution conditional on the trained prior, not robustness to arbitrary prior quality. The capacity sweep (Tables 11 and 12) also shows task dependence: one layer underperforms two on puzzle-3x3 and several navigation tasks, while greater depth does not consistently help on cube-double. A possible explanation is a balance between insufficient correction capacity and fitting optimistic critic errors. We have not directly measured this causal mechanism. The fixed four-layer main benchmark remains unchanged.

4.4 Runtime and Compute Trade-offs

DeFlow separates behavior-prior sampling from value improvement. The base policy still uses a multi-step flow solver, so DeFlow does not claim the same inference profile as one-step shortcut policies; the practical question is whether the residual module adds much overhead beyond the flow proposal itself. Let M denote the number of flow-solver steps, C_v the cost of one vector-field evaluation, C_f the cost of the residual module, and C_Q the cost of one critic evaluation. A direct BPTT actor update through an iterative generator retains the solver trajectory and backpropagates through M vector-field calls, with memory proportional to the unrolled solver. Shortcut policies reduce this to a one-step or few-step extraction cost, and SORL can spend test-time compute on shortcut step counts or critic-ranked candidates. DeFlow still pays MC_v to sample the prior, but its actor-gradient path only traverses f_ϕ and Q because $a_{\text{flow}} = \text{sg}[\mu_\psi(s, z)]$.

Table 3: Compute trade-offs for generative policy extraction. M is the number of flow-solver steps; C_v, C_f, and C_Q denote vector-field, residual-module, and critic costs.

Family	Update path	Eval. budget	Proposal use
Iterative BPTT	solver + critic	MC_v	update generator
Shortcut / FQL	shortcut + critic	1–few steps	compress actor
SORL-style	shortcut + critic	k steps; optional ranking	scale inference
DeFlow	residual + critic	$MC_v + C_f$	local residual

Table 4 reports wall-clock measurements on the same NVIDIA 3090 setup used for our main experiments. The measured overhead is modest in this implementation: DeFlow is slightly slower than FQL in offline training and inference because it preserves multi-step proposal sampling, while the residual MLP adds little extra cost relative to the flow and critic updates. These timings are implementation-specific and do not establish a speed advantage over SORL or a scaling law for larger models. DeFlow can also refine several proposals and rank them by the critic, at additional sampling and critic cost; that variant is not evaluated here. Applying the same separation to a large VLA-style prior is a motivation for future work, not an empirical result of this study.

Table 4: Training and inference time comparison in milliseconds per step.

Method	`Offline train`	`Online train`	`Inference`
`FQL`	2.2 ± 1.0	5.1 ± 0.5	4.2 ± 0.7
`DeFlow`	2.4 ± 1.0	5.3 ± 0.4	4.5 ± 0.5
`DeFlow (frozen prior)`	–	5.2 ± 0.5	–

5 Related Work

Offline RL under behavior constraints. Offline RL often improves value under constraints that keep the policy close to the static dataset [Levine et al., 2020]. Conservative and behavior-regularized methods use value penalties or actor regularization [Kumar et al., 2020, Fujimoto and Gu, 2021]; IQL and SQL avoid explicit out-of-distribution maximization [Kostrikov et al., 2022, Xu et al., 2023]. OGBench [Park et al., 2025a] exposes a harder version of the same problem: the policy class must represent multimodal behavior, but the extraction step must still avoid critic exploitation. DeFlow follows this constrained-improvement view with a residual budget around samples from a learned flow prior.

Diffusion and flow policies for offline RL. Diffusion policies and flow-matching policies are attractive for offline RL because they can model complex conditional action distributions. Value guidance and weighted policy objectives offer several ways to incorporate critic information into expressive policies [Lu et al., 2023, Kang et al., 2023, Zhang et al., 2025a, Peng et al., 2019]. Other methods backpropagate value gradients through the iterative denoising or sampling process [Wang et al., 2023, Ding and Jin, 2024], which provides a more direct policy-improvement signal but introduces substantial memory and stability costs. Q-score matching [Psenka et al., 2024] instead connects the policy score to the action gradient of the Q-function and differentiates through the denoising model rather than the full sampling chain. Thus, avoiding BPTT is an established design property; DeFlow's distinction is to apply value gradients to a proposal-conditioned action residual while training the prior only through behavior modeling. Flow matching offers straighter generation paths and has recently been used for policy and value modeling in FQL, floq, SAC Flow, MeanFlow-QL, SORL, and Value Flows [Lipman et al., 2023, Park et al., 2025b, Agrawalla et al., 2025, Zhang et al., 2025b, Wang et al., 2025, Espinosa-Dice et al., 2025, Dong et al., 2025b]. DeFlow is closest to this flow-based line, but keeps the iterative flow model as a behavior prior and applies critic gradients only to a small residual extractor.

Shortcut extraction and SORL. FQL and related shortcut methods reduce the optimization burden by distilling or parameterizing a one-step policy, making direct Q optimization cheaper than

differentiating through a full solver [Park et al., 2025b, Frans et al., 2025]. SORL extends this efficient extraction direction with step-size-conditioned shortcut models, shortcut-consistency training, and inference-time evaluation over shortcut step counts, with optional critic best-of-N selection; its strong results on several OGBench navigation and scene tasks make it an important comparison point [Espinosa-Dice et al., 2025]. DeFlow explores a different trade-off: retain a multi-step learned prior and regularize a local residual instead of using a shortcut policy. The empirical pattern in Table 1 is consistent with different extraction mechanisms being useful in different regimes; the unmatched inference budgets do not isolate the cause of this pattern.

Residual policy learning and refinement. Residual learning adds learned corrections on top of existing models, from residual networks [He et al., 2016] to residual policy learning [Silver et al., 2019]. Recent work trains small refinement modules on larger policies, often with online interaction [Ankile et al., 2025, Xiao et al., 2025, Dong et al., 2025a]. DeFlow uses the same broad decomposition in offline policy extraction, with an action-conditioned residual so different samples from the same state can receive different local corrections.

6 Discussions and Conclusion

We presented DeFlow, a BPTT-free framework for extracting policies from multi-step flow priors through action-conditioned residual refinement. The empirical picture is deliberately scoped: DeFlow is competitive with recent flow-based offline RL methods, improves selected multimodal manipulation and O2O tasks, and underperforms strong shortcut or inference-scaling methods in some navigation regimes. This suggests that offline generative policy extraction should be evaluated along several axes at once: behavior-prior fidelity, local value improvement, inference-time compute, and robustness to critic error.

Limitations. DeFlow depends on coverage of useful proposals: a local residual has no reliable mechanism for recovering distant behaviors absent from the data or missed by the prior. Proximity to an imperfect prior does not certify true data support. Both the budget δ and residual capacity affect performance, and adaptive α does not remove these choices or prevent critic exploitation. Our attribution tests do not systematically vary prior quality, and the critic-optimism explanation remains a hypothesis. Multi-step sampling retains a cost proportional to the solver budget, and the SORL comparison is not compute-matched.

Acknowledgments and Disclosure of Funding

Zhancun Mu thanks his former advisor, Yitao Liang, for his support. This work was supported by the Shanghai Artificial Intelligence Innovation Center under Grant No. 2025SHZDZX026D22. The authors declare no competing interests relevant to this work.

References

Bhavya Agrawalla, Michal Nauman, Khush Agrawal, and Aviral Kumar. floq: Training critics via flow-matching for scaling compute in value-based rl. *arXiv preprint arXiv:2509.06863*, 2025.

Lars Ankile, Anthony Simeonov, Idan Shenfeld, Marcel Torne, and Pulkit Agrawal. From imitation to refinement – residual rl for precise assembly. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 01–08. IEEE, 2025.

Philip J Ball, Laura Smith, Ilya Kostrikov, and Sergey Levine. Efficient online reinforcement learning with offline data. In *International Conference on Machine Learning (ICML)*, 2023.

Zihan Ding and Chi Jin. Consistency models as a rich and efficient policy class for reinforcement learning. In *International Conference on Learning Representations (ICLR)*, 2024.

Perry Dong, Qiyang Li, Dorsa Sadigh, and Chelsea Finn. Expo: Stable reinforcement learning with expressive policies, 2025a. URL `https://arxiv.org/abs/2507.07986`.

Perry Dong, Chongyi Zheng, Chelsea Finn, Dorsa Sadigh, and Benjamin Eysenbach. Value flows, 2025b. URL https://arxiv.org/abs/2510.07650.

Nicolas Espinosa-Dice, Yiyi Zhang, Yiding Chen, Bradley Guo, Owen Oertell, Gokul Swamy, Kiante Brantley, and Wen Sun. Scaling offline rl via efficient and expressive shortcut models. *arXiv preprint arXiv:2505.22866*, 2025.

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

Justin Fu, Aviral Kumar, Ofir Nachum, G. Tucker, and Sergey Levine. D4rl: Datasets for deep data-driven reinforcement learning. *ArXiv*, abs/2004.07219, 2020.

Scott Fujimoto and Shixiang Shane Gu. A minimalist approach to offline reinforcement learning. In *Neural Information Processing Systems (NeurIPS)*, 2021.

T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *arXiv*, 2018. URL https://arxiv.org/pdf/1801.01290.pdf.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

Bingyi Kang, Xiao Ma, Chao Du, Tianyu Pang, and Shuicheng Yan. Efficient diffusion policies for offline reinforcement learning. In *Neural Information Processing Systems (NeurIPS)*, 2023.

Ilya Kostrikov, Ashvin Nair, and Sergey Levine. Offline reinforcement learning with implicit q-learning. In *International Conference on Learning Representations (ICLR)*, 2022.

Aviral Kumar, Aurick Zhou, G. Tucker, and Sergey Levine. Conservative q-learning for offline reinforcement learning. In *Neural Information Processing Systems (NeurIPS)*, 2020.

Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint*, 2020.

Timothy P. Lillicrap, Jonathan J. Hunt, Alexander Pritzel, Nicolas Manfred Otto Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. In *International Conference on Learning Representations (ICLR)*, 2016.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *International Conference on Learning Representations (ICLR)*, 2023.

Yaron Lipman, Marton Havasi, Peter Holderrieth, Neta Shaul, Matt Le, Brian Karrer, Ricky T. Q. Chen, David Lopez-Paz, Heli Ben-Hamu, and Itai Gat. Flow matching guide and code. *ArXiv*, abs/2412.06264, 2024.

Cheng Lu, Huayu Chen, Jianfei Chen, Hang Su, Chongxuan Li, and Jun Zhu. Contrastive energy prediction for exact energy-guided diffusion sampling in offline reinforcement learning. In *International Conference on Machine Learning (ICML)*, 2023.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin A. Riedmiller. Playing atari with deep reinforcement learning. *ArXiv*, abs/1312.5602, 2013.

Mitsuhiko Nakamoto, Yuexiang Zhai, Anika Singh, Max Sobol Mark, Yi Ma, Chelsea Finn, Aviral Kumar, and Sergey Levine. Cal-ql: Calibrated offline rl pre-training for efficient online fine-tuning. In *Neural Information Processing Systems (NeurIPS)*, 2023.

Seohong Park, Kevin Frans, Benjamin Eysenbach, and Sergey Levine. Ogbench: Benchmarking offline goal-conditioned rl. In *International Conference on Learning Representations (ICLR)*, 2025a.

Seohong Park, Qiyang Li, and Sergey Levine. Flow q-learning. *ArXiv*, abs/2502.02538, 2025b.

Xue Bin Peng, Aviral Kumar, Grace Zhang, and Sergey Levine. Advantage-weighted regression: Simple and scalable off-policy reinforcement learning. *ArXiv*, abs/1910.00177, 2019.

Gabriel Peyré and Marco Cuturi. Computational optimal transport: With applications to data science. *Foundations and Trends in Machine Learning*, 11(5–6):355–607, 2019.

Michael Psenka, Alejandro Escontrela, Pieter Abbeel, and Yi Ma. Learning a diffusion model policy from rewards via q-score matching. In *Proceedings of the 41st International Conference on Machine Learning*, 2024.

Martin L Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

Tom Silver, Kelsey Allen, Josh Tenenbaum, and Leslie Kaelbling. Residual policy learning, 2019. URL `https://arxiv.org/abs/1812.06298`.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning (ICML)*, 2015.

Denis Tarasov, Vladislav Kurenkov, Alexander Nikulin, and Sergey Kolesnikov. Revisiting the minimalist approach to offline reinforcement learning. In *Neural Information Processing Systems (NeurIPS)*, 2023.

Cédric Villani et al. *Optimal transport: old and new*. Springer, 2009.

Zeyuan Wang, Da Li, Yulin Chen, Ye Shi, Liang Bai, Tianyuan Yu, and Yanwei Fu. One-step generative policies with q-learning: A reformulation of meanflow. *arXiv preprint arXiv:2511.13035*, 2025.

Zhendong Wang, Jonathan J Hunt, and Mingyuan Zhou. Diffusion policies as an expressive policy class for offline reinforcement learning. In *International Conference on Learning Representations (ICLR)*, 2023.

Wenli Xiao, Haotian Lin, Andy Peng, Haoru Xue, Tairan He, Yuqi Xie, Fengyuan Hu, Jimmy Wu, Zhengyi Luo, Linxi "Jim" Fan, Guanya Shi, and Yuke Zhu. Self-improving vision-language-action models with data generation via residual rl, 2025. URL `https://arxiv.org/abs/2511.00091`.

Haoran Xu, Li Jiang, Jianxiong Li, Zhuoran Yang, Zhaoran Wang, Victor Chan, and Xianyuan Zhan. Offline rl with no ood actions: In-sample learning via implicit value regularization. In *International Conference on Learning Representations (ICLR)*, 2023.

Shiyuan Zhang, Weitong Zhang, and Quanquan Gu. Energy-weighted flow matching for offline reinforcement learning. In *International Conference on Learning Representations (ICLR)*, 2025a.

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, 2025b. URL `https://arxiv.org/abs/2509.25756`.

A Residual Trust Region as an OT Surrogate

We use the standard Wasserstein formulation of optimal transport, where a Monge map deterministically pushes one distribution to another and W_2^2 is the minimum squared Euclidean cost over all couplings [Villani et al., 2009, Peyré and Cuturi, 2019].

Proposition A.1. *Fix a state s and let $\nu_s = \mu_\psi(\cdot|s)$ be the learned flow prior. Assume that ν_s and the residual have finite second moments. Define the refined policy by the deterministic map $T_s(a_b) = a_b + f_\phi(s, a_b)$ with $a_b \sim \nu_s$, i.e., $\pi_\phi(\cdot|s) = (T_s)_\# \nu_s$. Then*

$$W_2^2(\pi_\phi(\cdot|s), \nu_s) \le \mathbb{E}_{a_b \sim \nu_s} \left[\|f_\phi(s, a_b)\|_2^2 \right]. \tag{13}$$

Proof. The joint distribution induced by $(a_b, T_s(a_b))$ is a valid coupling between ν_s and $\pi_\phi(\cdot|s)$. The squared transport cost under this coupling is $\mathbb{E}_{a_b \sim \nu_s} \|a_b - T_s(a_b)\|_2^2 = \mathbb{E}_{a_b \sim \nu_s} \|f_\phi(s, a_b)\|_2^2$. Since W_2^2 is the minimum over all couplings, the inequality follows. □

This proposition only relates the refined policy to the learned flow prior. It does not imply exact preservation of the true data support, which depends on the quality and coverage of μ_ψ. With the implementation's coordinate-averaged penalty, the upper bound is $d\,\mathbb{E}\|f_\phi(s, a_b)\|_2^2/d$. Averaging over replay states gives $\mathbb{E}_s W_2^2 \le d\delta$ if the population constraint is satisfied. This is an average-state statement, not a per-state guarantee; a minibatch estimate alone is not a certified population bound. For bounded actions, the base proposal is clipped before refinement. Projection of the refined action onto the same box cannot increase its distance to that base proposal, so the same bound holds after action clipping.

B Implementation Details

We implement DeFlow based on the FQL [Park et al., 2025b] and floq [Agrawalla et al., 2025] codebases. The flow model architecture follows the same design as FQL, using a 4-layer MLP with 512 hidden units. The refinement module uses the same architecture for clear comparison. We use Q-Normalization as in FQL to stabilize Q-value scales across different tasks. For other hyperparameters except the target divergence δ, we follow the FQL/floq's default settings. The experiments use public OGBench and D4RL benchmarks and the FQL/floq implementation bases under their respective terms and licenses.

B.1 Hyperparameter Settings

For the target divergence δ, we set it based on the estimated Intrinsic Action Variance (IAV) of the dataset. Specifically, we compute the average variance of actions among k-nearest neighbors ($k = 5$) for each state in the following tasks and normalize it by the action dimension:

	`cube-double`	`antmaze-large`	`puzzle-3x3`	`antsoccer`
`IAV`	0.013	0.018	0.007	0.026

Leveraging prior knowledge of task complexity [Park et al., 2025a], we use the Intrinsic Action Variance (IAV) as a rough guide for choosing the constraint δ:

- **Fine Manipulation** (e.g., `cube-double-play`): High precision requires a tighter constraint, $\delta \approx 0.1 \times$ IAV.
- **Navigation/Locomotion** (e.g., `antsoccer`): Higher tolerance allows a looser constraint, $\delta \approx 1.0 \times$ IAV.
- **D4RL Benchmarks**: A generic setting suffices based on data quality (e.g., $\delta = 10^{-3}$).

IAV is a rough scale heuristic, not a validated predictor of an optimal budget. The actual settings include exceptions: humanoidmaze uses 0.001, puzzles use 0.0005, D4RL AntMaze uses 0.015, and pen uses 0.01. The complete task table, rather than the generic heuristic above, specifies the

reported configuration. We maintain the same δ for online fine-tuning without further adjustment. This strict protocol may limit asymptotic performance on specific outlier tasks, but it tests whether the residual constraint can transfer without task-specific retuning. Detailed hyperparameter settings for all experiments are provided in Table 14.

Selection protocol and reporting scope. Table 5 separates inherited settings, task-dependent choices, and post-submission diagnostics. The main benchmark used a four-layer residual throughout; additional depth comparisons do not replace those scores. We retain the same task budget during O2O adaptation. The original records do not provide a complete count of candidate trials or environment evaluations used during hyperparameter development. Thus, the reported configuration and diagnostic budgets are reproducible, but they should not be interpreted as a complete audit of historical selection cost.

Table 5: Hyperparameter choices and diagnostic scope. The sensitivity grids are diagnostic settings, not a claimed search history for selecting the main results.

Quantity	Main benchmark protocol	Additional analysis / budget
Residual budget δ	Task-dependent IAV-guided choice; retained for O2O	$\{0.0005, 0.001, 0.005, 0.01, 0.02\}$ diagnostic in Table 6
Penalty α	Adapted in log space; not a fixed-coefficient grid	Original nominal fixed settings $\{0.03, 0.1, 0.3, 1, 3\}$ in Table 9
Residual architecture	Four hidden layers, width 512; no per-task depth tuning	One/two layers, width 512; four seeds per setting; Section C.4
Prior and critic	FQL-based configuration; prior learns by flow matching	Held at the same architecture within each capacity comparison
Inference budget	Multi-step proposal followed by one residual	Ten Euler steps in capacity runs; no candidate ranking
Evaluation selection	Fixed-step main evaluation	Final scheduled checkpoint for capacity; 20 or 50 episodes per seed

Implementation conventions. The residual penalty is the minibatch mean of squared coordinates, $\widehat{g} = (Bd)^{-1} \sum_{i=1}^{B} \|\Delta a_i\|_2^2$. The actor treats α as constant, and the dual update treats $\widehat{g}$ as constant. In the audited adaptive implementation, $\eta = \log \alpha$ starts at 1 and uses the shared Adam learning rate 3×10^{-4}. Q-normalization divides the actor objective by the stopped minibatch mean absolute Q-value. Clipping is applied to the base proposal and the refined action. These conventions matter when translating a reported δ into an action-space displacement.

C Additional Experimental Results

We report the full offline RL results across all 73 tasks in Table 15. The results are averaged over 8 seeds (4 seeds for pixel-based tasks) and we bold scores that are within 95% of the best performance as in FQL [Park et al., 2025b].

C.1 Additional Diagnostics

This section collects diagnostics that are referenced in the main text but are not used as primary benchmark claims. They are intended to probe failure modes and component attribution.

Table 6: Sensitivity to the residual budget δ on a representative manipulation diagnostic. Larger budgets permit larger residual moves and can quickly degrade performance.

δ	0.0005	0.001	0.005	0.01	0.02
Score	55 ± 8	50 ± 7	27 ± 4	8 ± 5	4 ± 2

Table 7: Diagnostic checks for prior coverage and action-conditioned refinement. The mixed dataset uses 50% human and 50% expert data; the refinement ablation removes action conditioning from $f_\phi(s, a_b)$.

Diagnostic	Setting	Baseline	DeFlow / Variant
Mixed human/expert data	score	FQL: 21 ± 3	DeFlow: 37 ± 5
Refinement module	full action-conditioned $f_\phi(s, a_b)$	-	40 ± 5
Refinement module	remove action conditioning	-	5 ± 3

Table 8: Success rates on manipulation tasks when combining DeFlow with floq's value function.

Task	FQL	floq	DeFlow	DeFlow+floq
cube-double-play	25 ± 6	47 ± 15	40 ± 5	**58** ± 5
puzzle-3x3-play	29 ± 5	37 ± 7	43 ± 4	**50** ± 3

C.2 Ablations on α Adaptation

We retain the original comparison of adaptive penalties with fixed-coefficient settings on cube-double-play and puzzle-3x3-play, with the implementation qualification below.

Table 9: Original fixed-penalty sweep versus adaptation, separated by training phase. The fixed columns retain the nominal sweep setting c; see the implementation qualification below.

Task	Phase	$c = 0.03$	0.1	0.3	1	3	Adaptive
cube-double	Offline	0 ± 0	0 ± 0	0 ± 0	44 ± 2	7 ± 2	48 ± 13
	Online	0 ± 0	0 ± 0	0 ± 0	10 ± 3	89 ± 5	93 ± 4
puzzle-3x3	Offline	0 ± 0	0 ± 0	7 ± 7	0 ± 0	35 ± 8	23 ± 8
	Online	100 ± 1	100 ± 0	100 ± 0	0 ± 0	0 ± 0	100 ± 0

The original sweep illustrates task- and phase-dependent sensitivity: small nominal settings perform poorly offline on cube-double, whereas a setting that works offline on puzzle-3x3 can fail online. Adaptation is competitive in these examples but does not win every offline comparison. These returns do not by themselves establish that failures are caused by OOD actions or that the adaptive constraint is always satisfied.

Implementation qualification. The current fixed-penalty code path multiplies the configured coefficient twice, giving an effective coefficient c^2. The historical code version for this original sweep has not been recovered, so we retain its nominal labels rather than assert a calibrated effective-α axis. This table is descriptive evidence about the reported settings, not a matched-effective-penalty comparison. The new capacity diagnostics all use the adaptive branch and are unaffected by that fixed-branch issue.

C.3 Prior-Only and Residual Attribution

We evaluate the same four trained checkpoints per task with and without the residual at inference, using 50 episodes per checkpoint and mode. No component is retrained. Scores in Table 10 are success percentages, reported as mean and standard deviation across checkpoints. This tests whether the residual changes performance for the particular learned priors; it does not isolate the effects of prior architecture, prior training quality, or dataset coverage.

Table 10: Inference-time residual attribution at the same trained checkpoints. These are separate evaluations, not replacements for the main benchmark scores.

Task	Base flow only	Base flow + residual
cube-double task1	10 ± 2	62 ± 12
puzzle-3x3 task2	0.5 ± 0.9	35 ± 11

The improvement shows that these scores do not arise from the flow prior alone. It is complementary to the action-input ablation in Table 7: disabling the residual tests its inference-time contribution, whereas removing action conditioning changes the correction function being learned. Neither establishes a guarantee of mode preservation. The residual is conditioned on the proposal's position in action space, without explicit mode labels.

C.4 Residual-Capacity Sensitivity

We vary only the residual hidden-layer configuration within each new comparison, holding the flow prior and critic at four hidden layers of width 512. Each configuration uses four seeds, a fixed task budget, ten flow steps, batch size 256, and learning rate 3×10^{-4}. We report the last scheduled evaluation: one million offline updates for OGBench and 500,000 for pen-cloned. Cube-double uses seeds 2026–2029 and 20 evaluation episodes per seed; the other capacity runs use seeds 1996–1999 and 50 episodes. Means and population standard deviations are computed across the four final seed scores and rounded to whole points.

Table 11: Residual-depth diagnostics. All hidden layers have width 512. Cube's four-layer entry is a matched four-seed rerun (†); puzzle and scene four-layer entries are the original eight-seed benchmark references ($*$), not matched reruns.

Task	One layer	Two layers	Four layers
cube-double task1	81 ± 9	74 ± 23	69 ± 10 †
puzzle-3x3 task2	6 ± 3	55 ± 7	38 ± 11 $*$
scene task2	52 ± 12	70 ± 13	59 ± 17 $*$

The original cube-double task1 main-table result is 65 ± 6 across eight seeds; it is unchanged. The four-seed cube rerun above is distinct. Across these tasks, the results do not support a monotonic benefit from increasing residual depth. Comparisons to the original puzzle and scene references also differ in seeds and evaluation provenance, so they should not be treated as paired estimates of a depth effect.

Table 12: Broader one- versus two-layer sensitivity diagnostic. OGBench entries are success percentages; pen-cloned uses D4RL normalized return. No best-depth selection is applied to the main benchmark.

Task	One layer	Two layers
antmaze-large task2	48 ± 14	73 ± 6
humanoidmaze-medium task2	43 ± 18	73 ± 10
antsoccer-arena task2	28 ± 12	76 ± 3
cube-single task2	98 ± 1	97 ± 3
puzzle-4x4 task1	22 ± 2	24 ± 7
pen-cloned	63 ± 20	64 ± 8

Interpretation and limits. Two layers improve over one on several navigation tasks and puzzle-3x3, whereas cube-single is nearly saturated and the pen-cloned and puzzle-4x4 differences are small relative to their variability. A plausible explanation is a trade-off between representational underfitting and optimization against an imperfect offline critic [Levine et al., 2020]. Insufficient capacity may miss useful proposal-dependent corrections. Greater capacity may fit sharper directions of optimistic Q-error under a DDPG-style actor objective; a smaller network could sometimes provide useful

implicit regularization. These experiments do not measure Q-calibration, residual smoothness, or optimization conditioning directly, so they do not establish that explanation or show that smaller networks are intrinsically smoother. The same δ sets the same target penalty budget, but actual residual norms can differ because the constraint is only encouraged during training.

Table 13: Task-specific settings for the new capacity diagnostics. Target aggregation refers to the Bellman target, not candidate ranking.

Task	δ	Discount	Target aggregation
cube-double task1	0.001	0.99	mean
puzzle-3x3 task2	0.0005	0.99	mean
scene task2	0.001	0.99	mean
antmaze-large task2	0.01	0.99	min
humanoidmaze-medium task2	0.001	0.995	mean
antsoccer-arena task2	0.01	0.995	mean
cube-single task2	0.001	0.99	mean
puzzle-4x4 task1	0.0005	0.99	mean
pen-cloned	0.01	0.99	min

D Runtime and Compute Accounting

Table 4 reports the original NVIDIA RTX 3090 per-step timings. FQL and DeFlow take 2.2 ± 1.0 and 2.4 ± 1.0 ms per offline training step, respectively, and 4.2 ± 0.7 and 4.5 ± 0.5 ms per inference step. DeFlow's online training time is 5.3 ± 0.4 ms, versus 5.2 ± 0.5 ms when freezing the prior. These measurements characterize this implementation, not a controlled decomposition of residual-head overhead or an equal-compute comparison with SORL.

The additional capacity runs use ten Euler flow steps, batch size 256, and one million training updates for each OGBench run (500,000 for pen-cloned). Evaluation counts, seeds, and task-specific settings are listed in Section C.4 and Table 13. These update counts exclude evaluation and compilation time. The original timing record does not specify enough detail about warm-up, synchronization, timing repeats, or evaluation overhead to infer total experimental GPU-hours from per-step timings. We therefore do not report an extrapolated total-compute estimate.

The architecture avoids an actor gradient through the flow solver but still pays for its forward evaluations. A larger pretrained action prior could make this separation attractive, including VLA-style priors, but neither VLA adaptation nor large-model runtime scaling is tested here.

Table 14: **Task-specific hyperparameters for offline RL.** We take hyperparameters from FQL [Park et al., 2025b] for comparison.

Task	IQL α	ReBRAC (α_1, α_2)	IDQL N	SRPO β	CAC η	FAWAC α	FBRAC α	IFQL N	FQL α	DeFlow δ
antmaze-large-navigate-singletask-task1-v0	10	(0.003, 0.01)	32	0.3	1	3	3	32	10	0.01
antmaze-large-navigate-singletask-task2-v0	10	(0.003, 0.01)	32	0.3	1	3	3	32	10	0.01
antmaze-large-navigate-singletask-task3-v0	10	(0.003, 0.01)	32	0.3	1	3	3	32	10	0.01
antmaze-large-navigate-singletask-task4-v0	10	(0.003, 0.01)	32	0.3	1	3	3	32	10	0.01
antmaze-large-navigate-singletask-task5-v0	10	(0.003, 0.01)	32	0.3	1	3	3	32	10	0.01
antmaze-giant-navigate-singletask-task1-v0	10	(0.003, 0.01)	32	0.3	1	3	10	32	10	0.01
antmaze-giant-navigate-singletask-task2-v0	10	(0.003, 0.01)	32	0.3	1	3	10	32	10	0.01
antmaze-giant-navigate-singletask-task3-v0	10	(0.003, 0.01)	32	0.3	1	3	10	32	10	0.01
antmaze-giant-navigate-singletask-task4-v0	10	(0.003, 0.01)	32	0.3	1	3	10	32	10	0.01
antmaze-giant-navigate-singletask-task5-v0	10	(0.003, 0.01)	32	0.3	1	3	10	32	10	0.01
humanoidmaze-medium-navigate-singletask-task1-v0	10	(0.01, 0.01)	32	0.3	0.03	3	30	32	30	0.001
humanoidmaze-medium-navigate-singletask-task2-v0	10	(0.01, 0.01)	32	0.3	0.03	3	30	32	30	0.001
humanoidmaze-medium-navigate-singletask-task3-v0	10	(0.01, 0.01)	32	0.3	0.03	3	30	32	30	0.001
humanoidmaze-medium-navigate-singletask-task4-v0	10	(0.01, 0.01)	32	0.3	0.03	3	30	32	30	0.001
humanoidmaze-medium-navigate-singletask-task5-v0	10	(0.01, 0.01)	32	0.3	0.03	3	30	32	30	0.001
humanoidmaze-large-navigate-singletask-task1-v0	10	(0.01, 0.01)	32	0.3	1	3	30	32	30	0.001
humanoidmaze-large-navigate-singletask-task2-v0	10	(0.01, 0.01)	32	0.3	1	3	30	32	30	0.001
humanoidmaze-large-navigate-singletask-task3-v0	10	(0.01, 0.01)	32	0.3	1	3	30	32	30	0.001
humanoidmaze-large-navigate-singletask-task4-v0	10	(0.01, 0.01)	32	0.3	1	3	30	32	30	0.001
humanoidmaze-large-navigate-singletask-task5-v0	10	(0.01, 0.01)	32	0.3	1	3	30	32	30	0.001
antsoccer-arena-navigate-singletask-task1-v0	1	(0.01, 0.01)	32	0.03	1	10	30	64	10	0.01
antsoccer-arena-navigate-singletask-task2-v0	1	(0.01, 0.01)	32	0.03	1	10	30	64	10	0.01
antsoccer-arena-navigate-singletask-task3-v0	1	(0.01, 0.01)	32	0.03	1	10	30	64	10	0.01
antsoccer-arena-navigate-singletask-task4-v0	1	(0.01, 0.01)	32	0.03	1	10	30	64	10	0.01
antsoccer-arena-navigate-singletask-task5-v0	1	(0.01, 0.01)	32	0.03	1	10	30	64	10	0.01
cube-single-play-singletask-task1-v0	1	(1, 0)	32	0.03	0.003	1	100	32	300	0.001
cube-single-play-singletask-task2-v0	1	(1, 0)	32	0.03	0.003	1	100	32	300	0.001
cube-single-play-singletask-task3-v0	1	(1, 0)	32	0.03	0.003	1	100	32	300	0.001
cube-single-play-singletask-task4-v0	1	(1, 0)	32	0.03	0.003	1	100	32	300	0.001
cube-single-play-singletask-task5-v0	1	(1, 0)	32	0.03	0.003	1	100	32	300	0.001
cube-double-play-singletask-task1-v0	0.3	(0.1, 0)	32	0.1	0.3	0.3	100	32	300	0.001
cube-double-play-singletask-task2-v0	0.3	(0.1, 0)	32	0.1	0.3	0.3	100	32	300	0.001
cube-double-play-singletask-task3-v0	0.3	(0.1, 0)	32	0.1	0.3	0.3	100	32	300	0.001
cube-double-play-singletask-task4-v0	0.3	(0.1, 0)	32	0.1	0.3	0.3	100	32	300	0.001
cube-double-play-singletask-task5-v0	0.3	(0.1, 0)	32	0.1	0.3	0.3	100	32	300	0.001
scene-play-singletask-task1-v0	10	(0.1, 0.01)	32	0.1	0.3	0.3	100	32	300	0.001
scene-play-singletask-task2-v0	10	(0.1, 0.01)	32	0.1	0.3	0.3	100	32	300	0.001
scene-play-singletask-task3-v0	10	(0.1, 0.01)	32	0.1	0.3	0.3	100	32	300	0.001
scene-play-singletask-task4-v0	10	(0.1, 0.01)	32	0.1	0.3	0.3	100	32	300	0.001
scene-play-singletask-task5-v0	10	(0.1, 0.01)	32	0.1	0.3	0.3	100	32	300	0.001
puzzle-3x3-play-singletask-task1-v0	10	(0.3, 0.01)	32	0.1	0.01	0.3	100	32	1000	0.0005
puzzle-3x3-play-singletask-task2-v0	10	(0.3, 0.01)	32	0.1	0.01	0.3	100	32	1000	0.0005
puzzle-3x3-play-singletask-task3-v0	10	(0.3, 0.01)	32	0.1	0.01	0.3	100	32	1000	0.0005
puzzle-3x3-play-singletask-task4-v0	10	(0.3, 0.01)	32	0.1	0.01	0.3	100	32	1000	0.0005
puzzle-3x3-play-singletask-task5-v0	10	(0.3, 0.01)	32	0.1	0.01	0.3	100	32	1000	0.0005
puzzle-4x4-play-singletask-task1-v0	3	(0.3, 0.01)	32	0.1	0.01	0.3	300	32	1000	0.0005
puzzle-4x4-play-singletask-task2-v0	3	(0.3, 0.01)	32	0.1	0.01	0.3	300	32	1000	0.0005
puzzle-4x4-play-singletask-task3-v0	3	(0.3, 0.01)	32	0.1	0.01	0.3	300	32	1000	0.0005
puzzle-4x4-play-singletask-task4-v0	3	(0.3, 0.01)	32	0.1	0.01	0.3	300	32	1000	0.0005
puzzle-4x4-play-singletask-task5-v0	3	(0.3, 0.01)	32	0.1	0.01	0.3	300	32	1000	0.0005
antmaze-umaze-v2	-	-	-	-	0.01	3	10	32	10	0.015
antmaze-umaze-diverse-v2	-	-	-	-	0.01	3	10	32	10	0.015
antmaze-medium-play-v2	-	-	-	-	0.01	3	10	32	10	0.015
antmaze-medium-diverse-v2	-	-	-	-	0.01	3	10	32	10	0.015
antmaze-large-play-v2	-	-	-	-	4.5	3	1	32	3	0.015
antmaze-large-diverse-v2	-	-	-	-	3.5	3	1	32	3	0.015
pen-human-v1	-	-	32	0.03	0.003	0.03	30000	32	10000	0.01
pen-cloned-v1	-	-	32	0.1	0.003	0.3	10000	32	10000	0.01
pen-expert-v1	-	-	32	0.1	0.03	0.1	30000	32	3000	0.01
door-human-v1	-	-	32	0.01	0.03	1	30000	32	30000	0.001
door-cloned-v1	-	-	32	0.03	0.03	1	10000	128	30000	0.001
door-expert-v1	-	-	32	0.01	0.03	3	30000	32	30000	0.001
hammer-human-v1	-	-	128	0.1	0.03	3	30000	32	30000	0.001
hammer-cloned-v1	-	-	32	0.1	0.003	0.03	10000	32	10000	0.001
hammer-expert-v1	-	-	32	0.03	0.03	3	30000	32	30000	0.001
relocate-human-v1	-	-	32	0.03	0.01	0.3	30000	128	10000	0.001
relocate-cloned-v1	-	-	64	0.03	0.01	0.1	3000	32	30000	0.001
relocate-expert-v1	-	-	32	0.01	0.003	1	30000	32	30000	0.001
visual-cube-single-play-singletask-task1-v0	1	(1, 0)	-	-	-	-	100	32	300	0.001
visual-cube-double-play-singletask-task1-v0	0.3	(0.1, 0)	-	-	-	-	100	32	100	0.001
visual-scene-play-singletask-task1-v0	10	(0.1, 0.01)	-	-	-	-	100	32	100	0.001
visual-puzzle-3x3-play-singletask-task1-v0	10	(0.3, 0.01)	-	-	-	-	100	32	300	0.0005
visual-puzzle-4x4-play-singletask-task1-v0	3	(0.3, 0.01)	-	-	-	-	300	32	300	0.0005

Table 15: **Full offline RL results.** We present the full results on the 73 OGBench and D4RL tasks. (*) indicates the default task in each environment. The results are averaged over 8 seeds (4 seeds for pixel-based tasks) unless otherwise mentioned.

	Gaussian Policies			Diffusion Policies			Flow Policies				
Task	BC	IQL	ReBRAC	IDQL	SRPO	CAC	FAWAC	FBRAC	IFQL	FQL	Deflow
antmaze-large-navigate-singletask-task1-v0 (*)	0 ±0	48 ±9	**91** ±10	0 ±0	0 ±0	42 ±7	1 ±1	70 ±20	24 ±17	80 ±8	**88** ±6
antmaze-large-navigate-singletask-task2-v0	6 ±3	42 ±6	**88** ±4	14 ±8	4 ±4	1 ±1	0 ±1	35 ±12	8 ±3	57 ±10	71 ±10
antmaze-large-navigate-singletask-task3-v0	29 ±5	72 ±7	51 ±18	26 ±8	3 ±2	49 ±10	12 ±4	83 ±15	52 ±17	**93** ±3	**88** ±5
antmaze-large-navigate-singletask-task4-v0	8 ±3	51 ±9	**84** ±7	62 ±25	45 ±19	17 ±6	10 ±3	37 ±18	18 ±8	**80** ±4	77 ±5
antmaze-large-navigate-singletask-task5-v0	10 ±3	54 ±22	**90** ±2	2 ±2	1 ±1	55 ±6	9 ±5	76 ±8	38 ±18	83 ±4	84 ±3
antmaze-giant-navigate-singletask-task1-v0 (*)	0 ±0	0 ±0	**27** ±22	0 ±0	0 ±0	0 ±0	0 ±0	0 ±1	0 ±0	4 ±5	4 ±4
antmaze-giant-navigate-singletask-task2-v0	0 ±0	1 ±1	16 ±17	0 ±0	0 ±0	0 ±0	0 ±0	4 ±7	0 ±0	9 ±7	**36** ±13
antmaze-giant-navigate-singletask-task3-v0	0 ±0	0 ±0	**34** ±22	0 ±0	0 ±0	0 ±0	0 ±0	0 ±0	0 ±0	0 ±1	2 ±3
antmaze-giant-navigate-singletask-task4-v0	0 ±0	0 ±0	5 ±12	0 ±0	0 ±0	0 ±0	0 ±0	9 ±4	0 ±0	**14** ±23	5 ±8
antmaze-giant-navigate-singletask-task5-v0	1 ±1	19 ±7	**49** ±22	0 ±1	0 ±0	0 ±0	0 ±0	6 ±10	13 ±9	16 ±28	16 ±19
humanoidmaze-medium-navigate-singletask-task1-v0 (*)	1 ±0	32 ±7	16 ±9	1 ±1	0 ±0	38 ±19	6 ±2	25 ±8	**69** ±19	19 ±12	21 ±11
humanoidmaze-medium-navigate-singletask-task2-v0	1 ±0	41 ±9	18 ±16	1 ±1	1 ±1	47 ±35	40 ±2	76 ±10	85 ±11	**94** ±3	69 ±5
humanoidmaze-medium-navigate-singletask-task3-v0	6 ±2	25 ±5	36 ±13	0 ±1	2 ±1	**83** ±18	19 ±2	27 ±11	49 ±49	74 ±18	63 ±14
humanoidmaze-medium-navigate-singletask-task4-v0	0 ±0	0 ±1	**15** ±16	1 ±1	1 ±1	5 ±4	1 ±1	1 ±2	1 ±1	3 ±4	6 ±7
humanoidmaze-medium-navigate-singletask-task5-v0	2 ±1	66 ±4	24 ±20	1 ±1	3 ±3	91 ±5	31 ±7	63 ±9	**98** ±2	**97** ±2	81 ±7
humanoidmaze-large-navigate-singletask-task1-v0 (*)	0 ±0	3 ±1	2 ±1	0 ±0	0 ±0	1 ±1	0 ±0	0 ±1	6 ±2	**7** ±6	3 ±4
humanoidmaze-large-navigate-singletask-task2-v0	**0** ±0	**0** ±0	**0** ±0	**0** ±0	**0** ±0	**0** ±0	**0** ±0	**0** ±0	**0** ±0	**0** ±0	**0** ±4
humanoidmaze-large-navigate-singletask-task3-v0	1 ±1	7 ±3	8 ±4	3 ±1	1 ±1	2 ±3	1 ±1	10 ±2	**48** ±10	11 ±7	10 ±6
humanoidmaze-large-navigate-singletask-task4-v0	1 ±0	1 ±0	1 ±1	0 ±0	0 ±0	0 ±1	0 ±0	0 ±0	1 ±1	2 ±3	**8** ±3
humanoidmaze-large-navigate-singletask-task5-v0	0 ±1	1 ±1	**2** ±2	0 ±0	0 ±0	0 ±0	0 ±0	1 ±1	0 ±0	1 ±3	1 ±1
antsoccer-arena-navigate-singletask-task1-v0	2 ±1	14 ±5	0 ±0	44 ±12	2 ±1	1 ±3	22 ±2	17 ±3	61 ±25	77 ±4	**84** ±7
antsoccer-arena-navigate-singletask-task2-v0	2 ±2	17 ±7	0 ±1	15 ±12	3 ±1	0 ±0	8 ±1	8 ±2	75 ±3	**88** ±3	**89** ±4
antsoccer-arena-navigate-singletask-task3-v0	0 ±0	6 ±4	0 ±0	0 ±0	0 ±0	8 ±19	11 ±5	16 ±3	14 ±22	**61** ±6	56 ±7
antsoccer-arena-navigate-singletask-task4-v0 (*)	1 ±0	3 ±2	0 ±0	0 ±1	0 ±0	0 ±0	12 ±3	24 ±4	16 ±9	39 ±6	**43** ±5
antsoccer-arena-navigate-singletask-task5-v0	0 ±0	2 ±2	0 ±0	0 ±0	0 ±0	0 ±0	9 ±2	15 ±4	0 ±1	36 ±9	**39** ±9
cube-single-play-singletask-task1-v0	10 ±5	88 ±3	89 ±5	**95** ±2	89 ±7	77 ±28	81 ±9	73 ±33	79 ±4	**97** ±2	**96** ±3
cube-single-play-singletask-task2-v0 (*)	3 ±1	85 ±8	92 ±4	**96** ±2	82 ±16	80 ±30	81 ±9	83 ±13	73 ±3	**97** ±2	**99** ±1
cube-single-play-singletask-task3-v0	9 ±3	91 ±5	93 ±3	**99** ±1	**96** ±2	**98** ±1	87 ±4	82 ±12	88 ±4	**98** ±2	**100** ±1
cube-single-play-singletask-task4-v0	2 ±1	73 ±6	**92** ±3	**93** ±4	70 ±18	**91** ±2	79 ±6	79 ±20	79 ±6	**94** ±3	**95** ±4
cube-single-play-singletask-task5-v0	3 ±3	78 ±9	87 ±8	**90** ±6	61 ±12	80 ±20	78 ±10	76 ±33	77 ±7	**93** ±3	**92** ±5
cube-double-play-singletask-task1-v0	8 ±3	27 ±5	45 ±6	39 ±19	7 ±6	21 ±8	21 ±7	47 ±11	35 ±9	**61** ±9	**65** ±6
cube-double-play-singletask-task2-v0 (*)	0 ±0	1 ±1	7 ±3	16 ±10	0 ±0	2 ±2	2 ±1	22 ±12	9 ±5	36 ±6	**49** ±13
cube-double-play-singletask-task3-v0	0 ±0	0 ±0	4 ±1	17 ±8	0 ±1	3 ±1	1 ±1	4 ±2	8 ±5	22 ±5	**34** ±15
cube-double-play-singletask-task4-v0	0 ±0	0 ±0	1 ±1	0 ±1	0 ±0	0 ±1	0 ±0	0 ±1	1 ±1	5 ±2	**7** ±4
cube-double-play-singletask-task5-v0	0 ±0	4 ±3	4 ±2	1 ±1	0 ±0	3 ±2	2 ±1	2 ±2	17 ±6	19 ±10	**46** ±11
scene-play-singletask-task1-v0	19 ±6	94 ±3	**95** ±2	**100** ±0	94 ±4	**100** ±1	87 ±8	**96** ±8	**98** ±3	**100** ±0	**98** ±2
scene-play-singletask-task2-v0 (*)	1 ±1	12 ±3	50 ±13	33 ±14	2 ±2	50 ±40	18 ±8	46 ±10	0 ±0	**76** ±9	59 ±17
scene-play-singletask-task3-v0	1 ±1	32 ±7	55 ±16	**94** ±4	4 ±4	49 ±16	38 ±9	78 ±14	54 ±19	**98** ±1	85 ±5
scene-play-singletask-task4-v0	2 ±2	0 ±1	3 ±3	4 ±3	0 ±0	0 ±0	6 ±1	4 ±4	0 ±0	5 ±1	**13** ±8
scene-play-singletask-task5-v0	**0** ±0	**0** ±0	**0** ±0	**0** ±0	**0** ±0	**0** ±0	**0** ±0	**0** ±0	**0** ±0	**0** ±0	**0** ±0
puzzle-3x3-play-singletask-task1-v0	5 ±2	33 ±6	**97** ±4	52 ±12	89 ±5	**97** ±2	25 ±9	63 ±19	**94** ±3	90 ±4	89 ±7
puzzle-3x3-play-singletask-task2-v0	1 ±1	4 ±3	1 ±1	0 ±1	0 ±1	0 ±0	4 ±2	2 ±2	1 ±2	16 ±5	**38** ±11
puzzle-3x3-play-singletask-task3-v0	1 ±1	3 ±2	3 ±1	0 ±0	0 ±0	0 ±0	1 ±0	1 ±1	0 ±0	10 ±3	**35** ±11
puzzle-3x3-play-singletask-task4-v0 (*)	1 ±1	2 ±1	2 ±1	0 ±0	0 ±0	0 ±0	1 ±1	2 ±2	0 ±0	16 ±5	**23** ±8
puzzle-3x3-play-singletask-task5-v0	1 ±0	3 ±2	5 ±3	0 ±0	0 ±0	0 ±0	1 ±1	2 ±2	0 ±0	16 ±3	31 ±8
puzzle-4x4-play-singletask-task1-v0	1 ±1	12 ±2	26 ±4	**48** ±5	24 ±9	44 ±10	1 ±2	32 ±9	**49** ±9	34 ±8	23 ±3
puzzle-4x4-play-singletask-task2-v0	0 ±0	7 ±4	12 ±4	14 ±5	0 ±1	0 ±0	0 ±1	5 ±3	4 ±4	**16** ±5	9 ±2
puzzle-4x4-play-singletask-task3-v0	0 ±0	9 ±3	15 ±3	34 ±5	21 ±10	29 ±12	1 ±1	20 ±10	**50** ±14	18 ±5	9 ±4
puzzle-4x4-play-singletask-task4-v0 (*)	0 ±0	5 ±2	10 ±3	**26** ±6	7 ±4	1 ±1	0 ±0	5 ±1	21 ±11	11 ±3	7 ±4
puzzle-4x4-play-singletask-task5-v0	0 ±0	4 ±1	7 ±3	**24** ±11	1 ±1	0 ±0	0 ±1	4 ±3	2 ±2	7 ±3	7 ±4
antmaze-umaze-v2	55	77	**98**	**94**	**97**	66 ±5	90 ±6	**94** ±3	92 ±6	**96** ±2	**96** ±3
antmaze-umaze-diverse-v2	47	54	84	80	82	66 ±11	55 ±7	82 ±9	62 ±12	**89** ±5	**87** ±8
antmaze-medium-play-v2	0	66	**90**	84	81	49 ±24	52 ±12	77 ±7	56 ±15	78 ±7	80 ±7
antmaze-medium-diverse-v2	1	74	**84**	**85**	75	0 ±1	44 ±15	77 ±6	60 ±25	71 ±13	76 ±19
antmaze-large-play-v2	0	42	52	64	54	0 ±0	10 ±6	32 ±21	55 ±9	**84** ±7	**78** ±6
antmaze-large-diverse-v2	0	30	64	68	54	0 ±0	16 ±10	20 ±17	64 ±8	**83** ±4	78 ±6
pen-human-v1	71	78	**103**	76 ±10	69 ±7	64 ±8	67 ±5	77 ±7	71 ±12	53 ±6	61 ±9
pen-cloned-v1	52	83	**103**	64 ±7	61 ±7	56 ±10	62 ±10	67 ±9	80 ±11	74 ±11	75 ±4
pen-expert-v1	110	128	**152**	140 ±6	134 ±4	103 ±9	118 ±6	119 ±7	139 ±5	142 ±6	138 ±5
door-human-v1	2	3	−0	6 ±2	3 ±3	5 ±2	2 ±1	4 ±2	**7** ±2	0 ±0	1 ±0
door-cloned-v1	−0	**3**	0	0 ±0	0 ±0	1 ±0	0 ±1	0 ±0	2 ±2	2 ±1	2 ±1
door-expert-v1	**105**	**107**	**106**	**105** ±1	**105** ±0	98 ±3	**103** ±1	**104** ±1	**104** ±2	**104** ±1	**104** ±1
hammer-human-v1	**3**	2	0	2 ±1	1 ±1	2 ±0	2 ±1	2 ±1	**3** ±1	1 ±1	1 ±0
hammer-cloned-v1	1	2	5	2 ±1	2 ±1	1 ±1	1 ±0	2 ±1	2 ±1	**11** ±9	7 ±4
hammer-expert-v1	127	**129**	**134**	125 ±4	127 ±0	92 ±11	118 ±3	119 ±9	117 ±9	125 ±3	127 ±1
relocate-human-v1	**0**	**0**	**0**	**0** ±0	**0** ±0	**0** ±0	**0** ±0	**0** ±0	**0** ±0	**0** ±0	**0** ±0
relocate-cloned-v1	−0	0	**2**	−0 ±0	−0 ±0	−0 ±0	−0 ±0	1 ±1	−0 ±0	−0 ±0	1 ±8
relocate-expert-v1	**108**	**106**	**108**	**107** ±1	**106** ±2	93 ±6	**105** ±3	**105** ±2	**104** ±3	**107** ±1	**109** ±2
visual-cube-single-play-singletask-task1-v0[1]	-	70 ±12	**83** ±6	-	-	-	-	55 ±8	49 ±7	81 ±12	**86** ±9
visual-cube-double-play-singletask-task1-v0[1]	-	**34** ±23	4 ±4	-	-	-	-	6 ±2	8 ±6	21 ±11	**53** ±7
visual-scene-play-singletask-task1-v0[1]	-	**97** ±2	**98** ±4	-	-	-	-	46 ±4	86 ±10	**98** ±3	**97** ±2
visual-puzzle-3x3-play-singletask-task1-v0[1]	-	7 ±15	88 ±4	-	-	-	-	7 ±2	**100** ±0	94 ±1	**97** ±2
visual-puzzle-4x4-play-singletask-task1-v0[1]	-	0 ±0	26 ±6	-	-	-	-	0 ±0	8 ±15	**33** ±6	14 ±2

[1] Due to the high computational cost of pixel-based tasks, we selectively benchmark 5 methods that achieve strong performance on state-based OGBench tasks.

NeurIPS Paper Checklist

- **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 offline RL policy-extraction problem, the decoupled flow prior and refinement module, the adaptive trust-region constraint, and the benchmark scope, and these claims are developed in Sections 1, 3, and 4.

  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.

- **Limitations**

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

  Answer: [Yes]

  Justification: Section 6 discusses prior coverage, critic error, budget and capacity sensitivity, compute cost, and the limits of the diagnostic explanations.

  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.

- **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: Appendix A states the residual trust-region proposition, its assumptions, and a complete proof. The main text uses this result only as an interpretation of proximity to the learned flow prior, not as a guarantee of exact data-support preservation.

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.

- **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: Section 4 and the appendix describe the benchmark protocol, baselines, evaluation seeds, architecture and optimization settings, target-divergence choices, and full task-level results needed to reproduce the main experiments.

  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
    - If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
    - If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
    - 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).
    - 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.

- **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: [No]

  Justification: OGBench and D4RL are public, and a code supplement was prepared for review. This draft does not yet provide a verified public code-release link covering the final configuration and additional diagnostics.

  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.

- **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: Section 4 and Appendix B report the benchmark suite, baseline set, evaluation protocol, shared architecture and optimizer settings, and benchmark-level target-divergence choices used for the reported experiments.

  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.

- **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 main result tables report mean $\pm$ standard deviation over 8 state-based or 4 pixel-based seeds. Section C.4 separately specifies the four-seed final-evaluation protocol for additional capacity diagnostics and identifies original benchmark references.

  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.

- **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: Section D and Table 4 report NVIDIA RTX 3090 per-step timings and diagnostic update budgets, but the original timing protocol and total experimental GPU-hours are not fully recorded.

  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).

- **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: To the best of our knowledge, the work conforms to the NeurIPS Code of Ethics: it studies standard offline RL algorithms on existing benchmark datasets and does not involve deception, human subjects, or deployment claims.

  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).

- **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 paper is a methodological study of offline RL policy extraction and does not introduce a deployed system, a human-facing dataset, or a high-risk released asset; we therefore judge a separate broader-impact discussion to be not applicable here.

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).

- **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 submission does not release a high-risk generative model, scraped dataset, or other asset that would require special misuse-mitigation safeguards.

  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.

- **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 cites the original OGBench, D4RL, FQL, and floq sources, and Appendix B identifies the public benchmarks and implementation bases used by DeFlow.

  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.

- **New assets**

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

  Answer: [No]

  Justification: The review code supplement includes a README and training scripts. A verified final public release, including the additional diagnostic configurations and licensing details, remains to be completed; no new dataset is introduced.

  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.

- **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 uses benchmark datasets and simulated environments only; it does not involve crowdsourcing or experiments with human participants.

  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.

- **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 subjects or crowdsourcing, so IRB approval or an 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.

- **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 an important, original, or non-standard component of the proposed method, theory, or experimental pipeline described in the paper.

  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.