UNLEASHING FLOW POLICIES WITH DISTRIBUTIONAL CRITICS

Anonymous authors
Paper under double-blind review

ABSTRACT

Generative flow models have recently emerged as a powerful paradigm for policy learning in offline reinforcement learning, capable of capturing complex, multi-modal action distributions inherent in diverse datasets. However, the full potential of these expressive actors is often bottlenecked by their critics, which typically learn a single scalar estimate of the expected return. We argue that directly regressing this mean value is suboptimal, as it often fails to capture the underlying stochasticity of returns, leading to high-variance and inaccurate value estimates. To address this, we introduce the Distributional Flow Critic (DFC), a novel architecture that utilizes flow matching to learn the complete state-action return distribution. Crucially, we demonstrate that explicitly modeling this distribution acts as a robust representation regularizer: by capturing the comprehensive outcome distribution rather than compressing complex dynamics into a single scalar, DFC yields a much higher-fidelity value signal. We extensively validate our approach across 93 diverse tasks from the D4RL and OGBench benchmarks, spanning locomotion and manipulation. Results show that DFC achieves the highest score on 13 out of 17 domain groups, with an average domain-wise score ratio of 1.09 relative to the strongest baseline in each group. DFC further excels in offline-to-online adaptation, achieving the best or tied-best fine-tuned performance on 4 out of 5 domains.

1 INTRODUCTION

In modern reinforcement learning, particularly in offline and offline-to-online settings, a central challenge is learning effective policies from complex, pre-collected datasets (Fujimoto & Gu, 2021; Tarasov et al., 2023b; Park et al., 2025b). To this end, flow-based policies, trained with generative techniques like flow matching, represent a significant advance (Lipman et al., 2023; Zhang et al., 2025). Unlike traditional parameterizations, such as unimodal Gaussians, flow-based methods can capture the rich, often multimodal action distributions inherent in diverse expert data. The success of such agents is measured not only by their static offline performance but also by their capacity for safe and efficient online fine-tuning.

Flow-based policies are commonly paired with scalar critics that directly estimate the expected return, $Q(s, a)$. We instead train the critic to model the full return distribution, $Z(s, a)$, providing supervision beyond a single scalar target (Bellemare et al., 2017; Dabney et al., 2018b). Our design is motivated by representation regularization: learning the range of possible returns encourages the critic to capture structure that mean regression does not explicitly model. The return-distribution diagnostics and critic ablations support this design, linking improved distribution fitting to stronger policy performance.

To harness this advantage, we introduce the **Distributional Flow Critic (DFC)**, a novel architecture designed to learn the entire return distribution, $Z(s, a)$, using flow matching. However, a naive application of flow matching to established distributional losses faces a significant hurdle: estimating the loss requires sampling from the learned distribution $Z_\phi^{\text{flow}}(s, a)$, which necessitates backpropagating through the entire trajectory of an ODE solver. This process is computationally expensive and notoriously unstable (Park et al., 2025c). To circumvent this, we propose an elegant **distillation-based architecture** comprising a multi-step flow critic and a quantile-conditioned one-step critic; the flow critic uses a standard target copy for Bellman bootstrapping. Our ablation studies (Sec-

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

tion 5.3) establish the empirical contributions of the flow teacher and one-step distillation. The flow critic captures the complex return distribution, while the one-step critic efficiently distills its distributional knowledge via quantile regression. Proposition 1 states the contraction property of exact quantile-projected Bellman evaluation under a fixed policy. We integrate DFC into the Flow Q-learning framework (Park et al., 2025c), allowing our distributional critic to provide a superior value approximation to an already powerful flow-based actor. Although our experiments focus on this specific integration, DFC is designed as a modular component, serving as a potential drop-in replacement for scalar critics in various off-policy algorithms evidenced by the ablation study in Section 5.3.

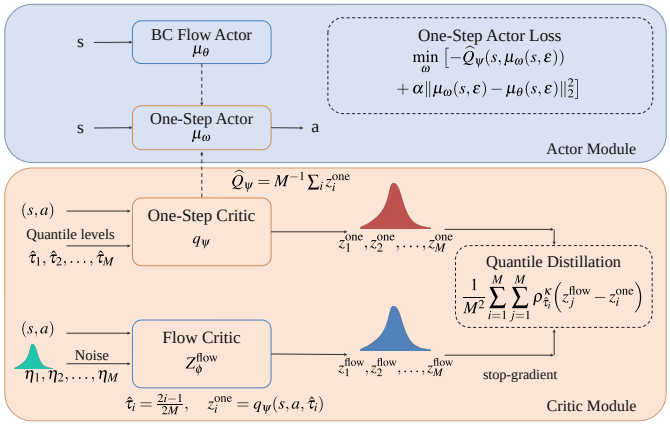
We evaluate DFC across offline and offline-to-online benchmarks and examine its critic through complementary diagnostics. Return-distribution comparisons show lower 1-Wasserstein error against Monte Carlo reference distributions, while the reported value-estimation analysis finds lower MAE than scalar FQL. Component ablations, scalar-ensemble comparisons, and critic replacements in ReBRAC and IQL further test the effectiveness of the design.

Our main contributions are:

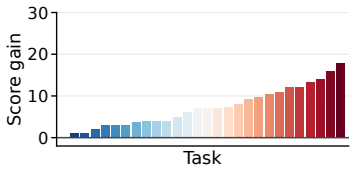
A Distributional Flow-Based Critic Architecture. We introduce a flow-matching model for the critic that moves beyond traditional scalar Q-value estimation. By learning to generate the entire distribution of future returns, our model provides a richer and more stable learning signal. We distill the flow teacher into a quantile critic to avoid differentiating through the ODE during policy optimization, and state a fixed-policy contraction result for exact quantile-projected Bellman evaluation.

Empirical Validation of Value Learning. Our return-distribution diagnostics show that DFC's 1-Wasserstein error is less than half that of the strongest baseline in each of the three displayed comparisons. The reported MAE analysis also finds lower value-estimation error than scalar FQL (Section 5.3). Critic ablations and ensemble comparisons provide complementary evidence for the effectiveness of the distributional flow critic.

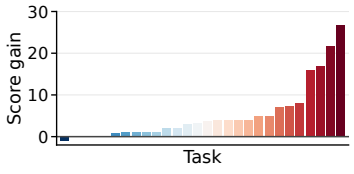
Strong Empirical Performance. Through extensive experiments on a wide range of benchmarks, we demonstrate that DFC achieves the highest score on 13 of 17 domain groups and attains an average domain-wise score ratio of 1.09 relative to the strongest baseline in each group. We further demonstrate its strong adaptability in tasks requiring multimodal action distributions and during offline-to-online fine-tuning.



(a) Distributional Flow Critic architecture



(b) Offline



(c) Offline-to-online

Figure 1: **Distributional Flow Critic: architecture and performance gains.** (a) A flow critic learns the return distribution and supervises a one-step critic through quantile regression. The one-step critic guides the actor without backpropagating through the ODE solver. (b–c) Performance comparison between DFC and FQL in the offline and offline-to-online settings. Bars show DFC's score improvements over FQL across the OGBench and D4RL benchmarks, sorted in ascending order. Results are averaged over 8 seeds.

2 PRELIMINARIES

2.1 DISTRIBUTIONAL REINFORCEMENT LEARNING

We consider a Markov Decision Process $(\mathcal{S}, \mathcal{A}, P, r, \gamma)$, where $\mathcal{S}$ and $\mathcal{A}$ are the state and action spaces, P is the transition kernel, r is the reward function, and $\gamma \in [0, 1)$ is the discount factor. A policy $\pi(a \mid s)$ (Sutton & Barto, 2018) maximizes the expected discounted return $J(\pi) = \mathbb{E}_\pi[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)]$. Standard RL estimates its expectation $Q^\pi(s, a)$, whereas Distributional RL (DRL) (Bellemare et al., 2017) models the return itself as a random variable:

$$Z^\pi(s, a) = \sum_{t=0}^{\infty} \gamma^t r(S_t, A_t), \text{ where } S_0 = s, A_0 = a.$$

The conventional action value is its expectation, $Q^\pi(s, a) = \mathbb{E}[Z^\pi(s, a)]$. The core of DRL is the *distributional Bellman equation*, which states that the distribution of the return at the current state-action pair is related to the distribution of the return at the next state: $\mathcal{T}^\pi Z(s, a) \overset{D}{:=} r(s, a) + \gamma Z^\pi(S', A')$, where $\mathcal{T}^\pi$ denotes the *distributional Bellman operator*, $\overset{D}{=}$ denotes equality in distribution, $S' \sim \mathcal{P}(\cdot|s, a)$, and $A' \sim \pi(\cdot|S')$.

DRL methods approximate Z^π using, for example, fixed categorical supports in C51 (Bellemare et al., 2017) or learned quantiles in QR-DQN (Dabney et al., 2018b). For one-dimensional distributions μ_X and μ_Y, their 1-Wasserstein distance is (Villani, 2003)

$$W_1(\mu_X, \mu_Y) = \int_0^1 \left| F_X^{-1}(\tau) - F_Y^{-1}(\tau) \right| d\tau.$$

Given i.i.d. samples $\{x_i\}_{i=1}^N$ and $\{y_i\}_{i=1}^N$, with sorted values $\tilde{x}_i$ and $\tilde{y}_i$, a natural empirical estimate is

$$\widehat{W}_1(\mu_X, \mu_Y) = \frac{1}{N} \sum_{i=1}^{N} |\tilde{x}_i - \tilde{y}_i|.$$

Directly optimizing this finite-sample estimate can introduce bias. Following QR-DQN (Dabney et al., 2018b), quantile regression (Koenker & Bassett, 1978) instead provides a stable sample-based objective that implicitly minimizes the Wasserstein discrepancy. We use this connection to distill flow-critic samples into the one-step critic.

2.2 FLOW MATCHING AND FLOW Q-LEARNING

Flow matching trains expressive generative models by learning a vector field v_θ that maps a simple noise distribution to a complex target distribution through an ordinary differential equation (ODE) (Lipman et al., 2023). The resulting *flow policies*, $\pi_\theta(a|s)$, can represent complex, multimodal action distributions that simpler parametric families such as Gaussians cannot capture. Direct optimization of these policies with reinforcement learning objectives, however, requires computationally expensive and unstable backpropagation through the ODE solver.

Flow Q-Learning (FQL) (Park et al., 2025c) addresses this problem by decoupling policy training from direct value maximization. It first trains an iterative flow policy with a behavioral cloning (BC) objective on the offline dataset. This BC flow policy, denoted by $\mu_\theta(s, \epsilon)$, maps an initial noise sample ϵ to an action and captures the dataset's action distribution. Although μ_θ is a deterministic function of (s, ϵ), sampling ϵ induces the stochastic policy $\pi_\theta(a \mid s)$. For simplicity, we use "policy" to refer to both μ_θ and π_θ.

FQL instead trains a separate, expressive one-step sampling map $\mu_\omega(s, \epsilon)$, inducing the policy $\pi_\omega(a \mid s)$. It maximizes Q-values while being regularized by distillation from the BC flow policy, which constrains deviation from the learned data distribution. The distillation loss matches the two sampling maps at the same state and noise:

$$\mathcal{L}_{\text{Distill}}(\omega) = \mathbb{E}_{s\sim\mathcal{D}, \epsilon\sim\mathcal{N}(0,I)} \left[\|\mu_\omega(s, \epsilon) - \mu_\theta(s, \epsilon)\|^2 \right]$$

The complete objective for the one-step policy π_ω combines the Q-maximization term with this distillation loss, which acts as a behavioral cloning regularizer:

$$\mathcal{L}_\pi(\omega) = \mathbb{E}_{s\sim\mathcal{D}, a^\pi\sim\pi_\omega(\cdot|s)}[-Q_\phi(s, a^\pi)] + \alpha \mathcal{L}_{\text{Distill}}(\omega) \tag{1}$$

This approach allows FQL to leverage the expressivity of the flow model while using a stable, one-step update for the RL objective, thereby avoiding unstable backpropagation through the ODE solver.

3 METHODOLOGY

Our core contribution is a novel distributional critic based on flow matching, designed to enhance the expressive power of any actor-critic method that employs a flow-based policy. In principle, our critic can be integrated into various such frameworks. To demonstrate its effectiveness, we build upon Flow Q-Learning (FQL), a strong flow-based offline RL method chosen for its stability and performance.

3.1 CRITIC AS A CONDITIONAL DISTRIBUTIONAL FLOW MODEL

A critical limitation in actor-critic methods arises when an expressive actor is paired with a simple critic. A standard critic that predicts only the mean of the return cannot capture complex value distributions, including multimodal or skewed returns, that an expressive actor might induce. This mismatch can lead to an impoverished learning signal and unstable training.

To address this, we propose a distributional critic that learns the entire distribution of returns. A naive approach would be to model the critic $Z_\phi^{\text{flow}}(s,a)$ as a single conditional flow model trained to minimize the Wasserstein distance to the target distribution, e.g., $W_1(\mathcal{T}^\pi Z_\phi^{\text{flow}}(s,a), Z_\phi^{\text{flow}}(s,a))$. However, this approach is infeasible for two key reasons:

Unstable Critic Training. Generating samples from Z_ϕ^{flow} to compute the loss would require solving an ODE. Training would thus necessitate backpropagation through the ODE solver's steps, which is notoriously unstable.

Unstable Actor Training. The actor update requires the gradient $\nabla_a Q(s,a)$. If the Q-value is the mean of the flow-generated distribution Z_ϕ^{flow}, computing this gradient would also require backpropagation through the ODE solver, which the very problem FQL was designed to avoid.

To circumvent this issue, we learn an **online distributional flow critic** Z_ϕ^{flow} using its standard target copy $Z_{\tilde{\phi}}^{\text{flow}}$, and distill the learned flow distribution into a separate **quantile-conditioned one-step critic** q_ψ. The online flow critic provides an expressive model of the return distribution, while the one-step critic predicts return quantiles in a single forward pass and thereby provides a direct, differentiable value estimate for policy optimization. The target copy is used only to construct stable bootstrapped Bellman targets for updating the online flow critic.

Learning the Flow Critic with Bootstrapped Targets. The online critic Z_ϕ^{flow} is a conditional flow model designed to approximate the return distribution. For a transition (s,a,r,s'), we construct M bootstrapped target return samples using the target copy:

$$\begin{aligned} a'_j &= \mu_\omega(s', \epsilon_j), \\ z'_j &= Z_{\tilde{\phi}}^{\text{flow}}(s', a'_j, \tilde{\xi}_j), \\ \tilde{z}_j &= r + \gamma\, \text{sg}[z'_j], \qquad j = 1, \dots, M, \end{aligned}$$

where $\tilde{\xi}_j, \epsilon_j \sim \mathcal{N}(0, I_d)$ and $\text{sg}[\cdot]$ denotes stop-gradient. The samples z'_j are generated by solving the ODE defined by the target copy, whose parameters are held fixed when constructing Bellman targets. With $t \sim \mathcal{U}(0,1)$, we define

$$z_j^t = (1-t)\tilde{\xi}_j + t\tilde{z}_j.$$

We then update the online parameters ϕ using the flow matching objective:

$$\mathcal{L}_{\text{Flow}}(\phi) = \mathbb{E}\left[\frac{1}{M}\sum_{j=1}^{M}\left\| v_\phi^{\text{flow}}(t, z_j^t \mid s,a) - (\tilde{z}_j - \tilde{\xi}_j)\right\|_2^2\right],$$

which trains the model to transport base Gaussian samples to the bootstrapped target distribution. The target parameters $\tilde{\phi}$ are updated from ϕ using the standard target-network rule.

Algorithm 1 Distributional Flow Critic (DFC)

Require: Dataset $\mathcal{D}$; parameters $\phi, \tilde{\phi}, \psi, \theta, \omega$; count M
Set $\hat{\tau}_i \leftarrow (2i-1)/(2M)$ for $i = 1, \ldots, M$
for each gradient step **do**
  Sample $\{(s, a, r, s')\} \sim \mathcal{D}$
  Flow critic: sample $\epsilon_j, \tilde{\xi}_j \sim \mathcal{N}(0, I_d)$
  $a'_j \leftarrow \mu_\omega(s', \epsilon_j), \quad \tilde{z}_j \leftarrow r + \gamma\, \mathrm{sg}[Z^{\mathrm{flow}}_{\tilde{\phi}}(s', a'_j, \tilde{\xi}_j)]$
  Sample $t \sim \mathcal{U}(0,1)$ and form $z^t_j \leftarrow (1-t)\tilde{\xi}_j + t\tilde{z}_j$
  Update ϕ with $\mathcal{L}_{\mathrm{Flow}}(\phi)$
  Update $\tilde{\phi}$ from ϕ using the target-network rule
  Distillation: sample $\eta_j \sim \mathcal{N}(0, I_d)$ and $z^{\mathrm{flow}}_j \leftarrow \mathrm{sg}[Z^{\mathrm{flow}}_{\phi}(s, a, \eta_j)]$
  Evaluate $z^{\mathrm{one}}_i \leftarrow q_\psi(s, a, \hat{\tau}_i)$ and update ψ with $\mathcal{L}_{\mathrm{one}}(\psi)$
  Policies: update behavior flow policy μ_θ by flow matching
  Estimate $\widehat{Q}_\psi(s, a) \leftarrow M^{-1} \sum_i q_\psi(s, a, \hat{\tau}_i)$
  Update ω by maximizing $\widehat{Q}_\psi$ with policy distillation regularization
end for
Ensure: One-step policy π_ω

Distilling the Flow Distribution via Quantile Regression. After updating the online flow critic, we draw M independent noises $\eta_j \sim \mathcal{N}(0, I_d)$ and sample its learned return distribution:

$$z^{\mathrm{flow}}_j = \mathrm{sg}\big[Z^{\mathrm{flow}}_{\phi}(s, a, \eta_j)\big], \qquad j = 1, \ldots, M.$$

Although obtaining these teacher samples requires an ODE solve, stop-gradient prevents backpropagation through the solver. We use a scalar-output, quantile-conditioned one-step critic $q_\psi(s, a, \tau)$, following the quantile-function parameterization of IQN (Dabney et al., 2018a). At the fixed midpoint levels,

$$\hat{\tau}_i = \frac{2i-1}{2M}, \qquad z^{\mathrm{one}}_i = q_\psi(s, a, \hat{\tau}_i), \qquad i = 1, \ldots, M.$$

Each prediction is paired with its conditioning quantile level in the quantile Huber objective (Dabney et al., 2018b):

$$\mathcal{L}_{\mathrm{one}}(\psi) = \mathbb{E}_{(s,a)\sim\mathcal{D},\, \eta_{1:M}} \left[\frac{1}{M^2} \sum_{i=1}^{M} \sum_{j=1}^{M} \rho^{\kappa}_{\hat{\tau}_i} \left(z^{\mathrm{flow}}_j - q_\psi(s, a, \hat{\tau}_i) \right) \right],$$

where $\rho^{\kappa}_{\hat{\tau}}$ is the quantile Huber loss function defined as:

$$\rho^{\kappa}_{\hat{\tau}}(u) = |\hat{\tau} - \mathbb{I}(u < 0)| \mathcal{L}_\kappa(u),$$

$$\text{with} \quad \mathcal{L}_\kappa(u) = \begin{cases} 0.5u^2 & \text{if } |u| \le \kappa \\ \kappa(|u| - 0.5\kappa) & \text{otherwise.} \end{cases}$$

The teacher samples provide return targets without requiring sorting or a noise-wise correspondence with the student. The quantile Huber objective provides a smooth surrogate for quantile regression, distilling the flow distribution into a feed-forward critic. The actor then uses the mean of the predicted quantiles without differentiating through the flow ODE.

3.2 Theoretical Analysis: Exact Quantile Projection

Fix a policy π, assume $|r| \le R_{\max}$ and $0 \le \gamma < 1$, and set $B = R_{\max}/(1-\gamma)$. Let $\mathcal{Z}$ be the space of measurable value-distribution functions supported on $[-B, B]$, and let $\mathcal{Z}_M \subset \mathcal{Z}$ contain distributions with M uniformly weighted atoms, allowing repeated atoms. We use the *maximal* ∞*-Wasserstein metric*:

$$\bar{d}_\infty(Z_1, Z_2) := \sup_{s,a} W_\infty(Z_1(s, a), Z_2(s, a)).$$

For $F_\mu^{-1}(u) := \inf\{x : F_\mu(x) \geq u\}$, define the canonical midpoint quantile projection, applied separately at each state-action pair, by

$$\Pi_{W_1}^M \mu := \frac{1}{M} \sum_{i=1}^{M} \delta_{F_\mu^{-1}(\hat{\tau}_i)}, \qquad \hat{\tau}_i = \frac{2i-1}{2M},$$

where δ_x is a point mass at x. This is an exact W_1-minimizing projection onto uniformly weighted M-atom distributions (Dabney et al., 2018b).

Proposition 1. *Under the conditions above, the exact projected Bellman operator $\hat{\mathcal{T}}_M := \Pi_{W_1}^M \mathcal{T}^\pi$ is a γ-contraction on $(\mathcal{Z}_M, \bar{d}_\infty)$ (Dabney et al., 2018b). For any $Z_1, Z_2 \in \mathcal{Z}_M$,*

$$\bar{d}_\infty(\hat{\mathcal{T}}_M Z_1, \hat{\mathcal{T}}_M Z_2) \leq \gamma \bar{d}_\infty(Z_1, Z_2).$$

The operator has a unique fixed point in $\mathcal{Z}_M$, and its exact iterates converge to that fixed point.

The proof is provided in Appendix A.

3.3 Policy Optimization

With the distilled one-step critic q_ψ in hand, we optimize the policy using the dual-policy architecture from FQL: a one-step policy π_ω maximizes Q-values while being regularized via distillation from an expressive, behaviorally-cloned flow policy μ_θ (Equation 1). Using the same midpoint quantile grid, we approximate the mean return by

$$\widehat{Q}_\psi(s,a) := \frac{1}{M} \sum_{i=1}^{M} q_\psi(s,a,\hat{\tau}_i).$$

The final objective combines Q-maximization with the policy-distillation regularizer:

$$\mathbb{E}\left[-\frac{1}{M} \sum_{i=1}^{M} q_\psi(s, \mu_\omega(s,\epsilon), \hat{\tau}_i) + \alpha \|\mu_\omega(s,\epsilon) - \mu_\theta(s,\epsilon)\|_2^2 \right].$$

Because q_ψ is a simple feed-forward network, the actor receives action gradients directly through its quantile predictions while ψ is held fixed, without backpropagating through an ODE solver.

The complete procedure is summarized in Algorithm 1. By capturing the full return distribution, our critic provides a richer learning signal than scalar value estimation, with the key practical benefit being an efficient mean-return estimate for policy optimization, as we empirically validate in Section 5.3.

4 Related Works

Additional related work, including concurrent studies on flow-based value learning, is discussed in Appendix D.

4.1 Diffusion and Flow Policy RL

Recent diffusion and flow models have been applied to reinforcement learning because they can represent complex, multimodal distributions (Sohl-Dickstein et al., 2015; Ho et al., 2020; Lipman et al., 2023; Esser et al., 2024). Applications include planning and hierarchical decision making (Janner et al., 2022; Ajay et al., 2023; Zheng et al., 2023; Liang et al., 2023; Suh et al., 2023; Venkatraman et al., 2024; Chen et al., 2024a), data generation and world modeling (Lu et al., 2023b; Ding et al., 2024; Jackson et al., 2024; Alonso et al., 2024), and expressive policy modeling (Mazoure et al., 2020; Ren et al., 2025; Park et al., 2025c). We focus on the last setting: training expressive generative policies for offline reinforcement learning.

Several strategies have been proposed to train flow and diffusion policies for offline reinforcement learning. These can be broadly categorized into three main paradigms: 1) **Value-weighted regression**, which prioritizes high-advantage actions but can be limited by the expressivity of

Task Category	IQL	ReBRAC	IDQL	CAC	IFQL	FQL	C51	IQN	EWFM	DFC
OGBench visual-antmaze-medium-navigate (5 tasks)	84.2 ±5	87.4 ±4	-	-	86.6 ±2	38.2 ±4	-	73.8 ±4	72.4 ±9	**88.2** ±6
OGBench visual-antmaze-teleport-navigate (5 tasks)	5.6 ±4	4.4 ±1	-	-	10.0 ±4	5.4 ±2	-	4.2 ±2	13.6 ±4	**14.6** ±4
OGBench visual-cube-double-play (5 tasks)	11.4 ±6	1.2 ±1	-	-	2.2 ±2	6.2 ±1	-	1.0 ±0	0.2 ±0	**13.2** ±3
OGBench visual-puzzle-3x3-play (5 tasks)	1.6 ±3	20.2 ±1	-	-	20.6 ±0	20.0 ±1	-	19.0 ±1	2.2 ±3	**33.0** ±3
OGBench visual-scene-play (5 tasks)	26.2 ±5	27.6 ±5	-	-	21.0 ±6	41.4 ±4	-	42.0 ±6	11.0 ±5	**54.2** ±3
OGBench antmaze-large-navigate (5 tasks)	53.4 ±3	80.8 ±5	20.8 ±5	32.8 ±4	28.0 ±5	78.6 ±3	0.0 ±0	40.6 ±3	40.0 ±6	**88.4** ±2
OGBench antmaze-giant-navigate (5 tasks)	4.0 ±1	**26.2** ±8	0.0 ±0	0.0 ±0	2.6 ±2	8.6 ±6	0.0 ±0	0.4 ±0	10.0 ±3	19.4 ±14
OGBench humanoidmaze-medium-navigate (5 tasks)	32.8 ±2	21.8 ±8	0.8 ±0	52.8 ±8	60.4 ±14	57.4 ±5	0.0 ±0	46.8 ±6	0.0 ±0	**66.6** ±14
OGBench humanoidmaze-large-navigate (5 tasks)	2.4 ±1	2.6 ±1	0.6 ±0	0.6 ±0	11.0 ±2	4.2 ±2	0.4 ±0	**15.6** ±2	0.0 ±0	8.0 ±3
OGBench antsoccer-arena-navigate (5 tasks)	8.4 ±2	0.0 ±0	11.8 ±4	1.8 ±4	33.2 ±6	60.2 ±2	2.4 ±0	33.6 ±5	4.8 ±2	**73.4** ±4
OGBench cube-single-play (5 tasks)	83.0 ±3	90.6 ±2	94.6 ±2	85.2 ±9	79.2 ±2	95.8 ±1	7.0 ±0	98.4 ±1	24.6 ±5	**99.8** ±0
OGBench cube-double-play (5 tasks)	6.4 ±1	12.2 ±1	14.6 ±6	5.8 ±2	14.0 ±3	28.6 ±2	1.8 ±0	**42.0** ±8	0.8 ±1	40.6 ±4
OGBench scene-play (5 tasks)	27.6 ±1	40.6 ±3	46.2 ±3	39.8 ±7	30.4 ±3	55.8 ±2	4.2 ±1	39.6 ±1	8.0 ±3	**63.2** ±2
OGBench puzzle-3x3-play (5 tasks)	9.0 ±1	21.6 ±1	10.4 ±2	19.4 ±0	19.0 ±1	29.6 ±1	1.0 ±0	14.6 ±1	0.2 ±0	**40.0** ±2
OGBench puzzle-4x4-play (5 tasks)	7.4 ±1	14.0 ±1	29.2 ±3	14.8 ±3	25.2 ±5	17.2 ±2	0.0 ±0	27.4 ±4	0.2 ±0	**35.0** ±4
D4RL antmaze (6 tasks)	57.2	78.7	79.2	30.2 ±3	64.8 ±7	83.5 ±3	20.0 ±0	64.8 ±3	75.2 ±3	**92.0** ±2
D4RL adroit (12 tasks)	53.4	**59.4**	52.2 ±1	42.9 ±2	52.4 ±1	51.6 ±1	48.2 ±2	49.8 ±3	13.2 ±3	55.9 ±3

Table 1: **Offline RL results.** DFC achieves the best or near-best performance on most of the 93 benchmark tasks, including 25 image-based tasks. Group means are computed from the per-task appendix results and shown to one decimal place; ± denotes the group SD across 8 seeds (4 for image-based tasks). The best scores are highlighted in **bold** and the second-best scores are underlined. Given the substantial computational overhead associated with image-based evaluation, we strategically restrict our comparison in the visual domains to a curated subset of 5 top-performing baselines from the state-based OGBench tasks.

Task	IQL	IFQL	FQL	IQN	DFC
antmaze-large-navigate	$41 \pm 5 \rightarrow 45 \pm 5$	$11 \pm 3 \rightarrow 3 \pm 2$	$\mathbf{80 \pm 8 \rightarrow 100 \pm 0}$	$55 \pm 5 \rightarrow 91 \pm 1$	$\mathbf{90 \pm 1 \rightarrow 100 \pm 0}$
humanoidmaze-medium-navigate	$21 \pm 5 \rightarrow 16 \pm 3$	$\mathbf{56 \pm 12 \rightarrow 82 \pm 7}$	$19 \pm 12 \rightarrow 22 \pm 13$	$23 \pm 3 \rightarrow 14 \pm 2$	$35 \pm 9 \rightarrow 59 \pm 32$
cube-double-play	$0 \pm 0 \rightarrow 0 \pm 0$	$12 \pm 3 \rightarrow 41 \pm 2$	$36 \pm 6 \rightarrow 95 \pm 2$	$29 \pm 4 \rightarrow 42 \pm 7$	$\mathbf{49 \pm 2 \rightarrow 98 \pm 2}$
puzzle-4x4-play	$5 \pm 1 \rightarrow 0 \pm 0$	$23 \pm 2 \rightarrow 19 \pm 12$	$11 \pm 3 \rightarrow 53 \pm 50$	$22 \pm 2 \rightarrow 6 \pm 1$	$\mathbf{20 \pm 3 \rightarrow 100 \pm 0}$
scene-play	$15 \pm 4 \rightarrow 10 \pm 3$	$0 \pm 0 \rightarrow 60 \pm 14$	$\mathbf{76 \pm 9 \rightarrow 100 \pm 0}$	$0 \pm 0 \rightarrow 0 \pm 0$	$\mathbf{96 \pm 3 \rightarrow 100 \pm 0}$

Table 2: **Offline-to-online evaluations on OGBench tasks.** We present full results on 5 OGBench tasks in the offline-to-online setting. DFC achieves strong fine-tuning performance compared to baselines. The $\rightarrow$ symbol separates offline and fine-tuned scores (offline result $\rightarrow$ result after one million online fine-tuning steps). Scores are reported as mean ± SD over 8 seeds, and we **bold** values of the best performance for each task.

policies trained on static datasets (Lu et al., 2023a; Kang et al., 2023; Zhang et al., 2025); 2) **Reparameterization-based policy gradients**, which optimize the value function by backpropagating through the policy but often suffer from instability and high variance (Wang et al., 2023; He et al., 2023; Ding & Jin, 2024); and 3) **Rejection sampling**, which enhances stability by filtering actions but incurs substantial computational overhead (Chen et al., 2023; Hansen-Estruch et al., 2023; He et al., 2024). Beyond these, other techniques include direct action gradients (Yang et al., 2023; Psenka et al., 2024; Li et al., 2024), bi-level MDP formulations (Ren et al., 2025), and integrations with implicit Q-learning (Chen et al., 2024b;c). Among value-weighted approaches, EWFM (Zhang et al., 2025) uses energy-based importance weights derived from Q-values to bias flow matching toward high-return actions. While effective on locomotion tasks, its reliance on a scalar critic limits its ability to capture distributional structure, leading to degraded performance on complex manipulation and high-dimensional navigation tasks (see Table 1).

In this work, we employ a reparameterized policy gradient approach. Distinct from prior methods such as Diffusion-QL (Wang et al., 2023), DiffCPS (He et al., 2023), Consistency-AC (Ding & Jin, 2024), SRDP (Ada et al., 2024), and EQL (Zhang et al., 2024), our method circumvents backpropagation through the generative model by instead training a one-step policy alongside a distributional critic. This strategy notably improves training stability and computational efficiency. Empirically, we demonstrate that our approach achieves a significant performance leap over existing baselines.

4.2 Distributional RL

Distributional Reinforcement Learning (DRL) marked a paradigm shift from modeling expected returns to capturing the full distribution of stochastic outcomes (Bellemare et al., 2017; Dabney et al., 2018b; Qu et al., 2019). Early value-based methods for discrete action spaces, such as C51 (Bellemare et al., 2017), parameterized the return distribution with discrete atoms. This was advanced by QR-DQN (Dabney et al., 2018b) and IQN (Dabney et al., 2018a), which respectively used discrete

quantiles and learned a full continuous quantile function for enhanced flexibility. The extension of DRL to continuous control yielded policy-gradient methods like D4PG (Barth-Maron et al., 2018), which demonstrated strong performance but inherited the representational limitations of a discrete categorical critic. To overcome this constraint, which makes it difficult to capture complex return distributions, SDPG (Singh et al., 2022) introduced a sample-based distributional policy gradient method that models the return distribution via reparameterization, thus avoiding the constraints of discrete representations.

DFC learns a return-distribution teacher through flow matching and transfers its predictions to a quantile critic for policy optimization. This separates distribution learning from the feed-forward value estimation used by the actor.

5 Experiment

5.1 Experimental Setup

Our evaluation spans 93 tasks from two benchmarks under both offline and offline-to-online settings. From OGBench (Park et al., 2025a), we use 50 state-based tasks across 10 domains and 25 image-based tasks across 5 domains, all measured by success rate. From D4RL (Fu et al., 2020), we include 6 `antmaze` tasks (success rate) and 12 `adroit` tasks (normalized return). We report mean $\pm$ standard deviation (SD) across 8 seeds for state-based tasks and 4 seeds for image-based tasks. Due to computational overhead, C51, IDQL, and CAC are excluded from image-based comparisons. We compare against Gaussian policies (IQL (Kostrikov et al., 2022), ReBRAC (Tarasov et al., 2023a)), diffusion policies (IDQL (Hansen-Estruch et al., 2023), CAC (Ding & Jin, 2024)), flow policies (IFQL, FQL (Park et al., 2025c), EWFM (Zhang et al., 2025)), and classical distributional methods (C51 (Bellemare et al., 2017), IQN (Dabney et al., 2018a)) paired with the same flow-actor backbone as FQL. Appendix B provides the complete implementation details: hyperparameters and architectures in Appendix B.1, and the dataset composition, environment list, and evaluation metrics in Appendix B.2.

5.2 Results

Performance across policy families. Our baseline suite evaluates DFC against several policy classes: it includes Gaussian policies (IQL and ReBRAC), diffusion policies (IDQL and CAC), flow policies (IFQL, FQL, and EWFM), and classical distributional critics (C51 and IQN) paired with the FQL actor. Across these alternatives, DFC ranks first on 8 of 12 state-based domain groups and is outperformed on `antmaze-giant-navigate`, `humanoidmaze-large-navigate`, `cube-double-play`, and D4RL Adroit (Table 1). Across all 17 domain groups, the unweighted mean of the DFC-to-best-baseline score ratios is 1.09.

Complementary actor and critic expressivity. The component comparisons show the benefit of combining a flow teacher with quantile distillation. Gaussian and diffusion baselines can be competitive on individual domains, while flow baselines benefit from a more expressive actor but still use scalar value information. Conversely, C51 and IQN add distributional value modeling to the same flow-actor backbone but remain inconsistent across domains. DFC combines both forms of expressivity and is particularly strong on difficult navigation and manipulation: it scores 73.4 on `antsoccer-arena-navigate` versus the best baseline's 60.2, and 40.6 on `cube-double-play` versus 28.6 for FQL and 14.8 for the strongest diffusion baseline. IQN remains competitive on the latter domain at 42.0. The ablations in Section 5.3 isolate the contribution of the critic design.

Generalization across task complexity and observations. DFC's gains extend from state-based locomotion to long-horizon navigation and multimodal manipulation, rather than being concentrated in one task type. They also persist under high-dimensional visual observations: DFC ranks first on all five image-based OGBench domains, including 54.2 on `visual-scene-play` compared with 42.0 for the strongest baseline. This pattern supports our hypothesis that modeling full return distributions is especially useful when heterogeneous outcomes are difficult to summarize with a scalar critic.

Offline-to-online adaptation. The comparison in Table 2 spans Gaussian (IQL), flow (IFQL and FQL), and distributional flow-actor (IQN) alternatives. DFC achieves the best or tied-best final performance on four of five tasks and reaches 100 on three, including an increase from 20 to 100 on `puzzle-4x4-play`. The learned representation therefore provides a strong initialization for online fine-tuning, not only a better offline policy.

Task-level consistency. Beyond domain averages, Figure 1 summarizes 28 comparisons (10 OGBench task groups and 18 D4RL tasks). Using the reported table means, DFC improves over FQL on 27 comparisons offline (one tie) and on 24 after fine-tuning (three ties and one decrease). The aggregate gains are therefore not driven by a few outlier tasks. Appendix C reports complete state-based and image-based per-task results in Tables 6 and 5, respectively, as well as offline-to-online learning curves in Figure 2.

5.3 Analysis

Value estimation accuracy. We validate our core motivation by comparing predicted Q-values with ground-truth values estimated from $5{,}000$ Monte Carlo rollouts on OGBench tasks. DFC achieves a mean absolute error (MAE) of 0.0432, which is less than half the 0.1085 error of FQL with direct mean regression. This result shows lower value-estimation error for DFC than for direct mean regression in the reported evaluation.

Contribution of the two-stage architecture. Table 7 compares DFC with a Direct Flow Critic (FC), which updates the actor by differentiating through the flow ODE, and a Quantile-Only Critic (DC), which removes the flow teacher. Removing distillation reduces offline performance on 57 of 68 tasks, with 8 ties; removing the flow teacher reduces performance on 53 tasks, with ties on the remaining 15. These ablations demonstrate the empirical contributions of both components. The full architecture combines flow-based distribution learning with direct policy gradients through the feed-forward quantile critic.

Transfer across actor-critic algorithms. Replacing the scalar critic with DFC improves ReBRAC on 8 of 10 tasks and IQL on 9 of 10, with one tie for each algorithm (Table 9). These results demonstrate that the benefits of DFC extend beyond its primary integration with FQL.

Comparison with scalar ensembles. DFC achieves the best or tied-best score on all 22 tasks compared with the two- and four-critic FQL ensembles (Table 8). Increasing the scalar ensemble from two to four critics does not close the performance gap, demonstrating the practical advantage of the proposed distributional critic over these ensemble alternatives.

Robustness to hyperparameters. We vary the common critic sample count $M \in \{51, 101\}$ following Bellemare et al. (2017); we use M teacher return samples and M student quantile levels. Appendix C, Table 10, shows that performance differences are minor and environment-dependent, with neither setting dominating across all tasks. This indicates that DFC is robust to the sample count.

Additional analyses of return-distribution fitting and computational overhead are provided in Appendix C (Figures 4 and 3, respectively).

6 Conclusion

In this work, we introduce the Distributional Flow Critic (DFC), which combines flow-based return-distribution learning with a distilled quantile critic for policy optimization. Return-distribution diagnostics, component ablations, and scalar-ensemble comparisons demonstrate the empirical benefits of this design. Our distillation-based design avoids backpropagation through the flow ODE during actor updates, while our theoretical analysis considers contraction of exact quantile-projected Bellman evaluation under a fixed policy. Across 93 locomotion and manipulation tasks, DFC achieves the highest score on 13 of 17 domain groups and yields consistent gains in both offline and offline-to-online settings. Moreover, the architecture is modular and can serve as a drop-in critic upgrade for diverse actor-critic frameworks, while introducing zero inference overhead. Future work will explore extending DFC to online RL and real-world robotic control.

AI USE STATEMENT

We used generative AI tools to aid and polish the writing, improving language clarity and readability. We also used these tools to search for relevant literature and check bibliographic information. The authors take responsibility for the final content of the paper, including its claims and references.

REPRODUCIBILITY STATEMENT

Algorithm 1 specifies the training procedure. Appendix B.1 reports network architectures and hyperparameters, while Appendix B.2 describes the benchmarks, tasks, and evaluation metrics. The experimental setup specifies the number of random seeds and the reported variability; Appendix C provides per-task results, ablations, learning curves, and runtime comparisons. The assumptions underlying Proposition 1 are stated in Section 3.2, and its proof is provided in Appendix A. We will release the source code publicly upon acceptance.

REFERENCES

Suzan Ece Ada, Erhan Oztop, and Emre Ugur. Diffusion policies for out-of-distribution generalization in offline reinforcement learning. *IEEE Robotics and Automation Letters*, 9(4):3116–3123, 2024.

Kingma DP Ba J Adam et al. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

Bhavya Agrawalla, Michal Nauman, Khush Agrawal, and Aviral Kumar. floq: Training critics via flow-matching for scaling compute in value-based RL. In *International Conference on Learning Representations*, 2026.

Anurag Ajay, Yilun Du, Abhi Gupta, Joshua B. Tenenbaum, Tommi S. Jaakkola, and Pulkit Agrawal. Is conditional generative modeling all you need for decision making? In *The Eleventh International Conference on Learning Representations*, 2023.

Eloi Alonso, Adam Jelley, Vincent Micheli, Anssi Kanervisto, Amos J Storkey, Tim Pearce, and François Fleuret. Diffusion for world modeling: Visual details matter in atari. *Advances in Neural Information Processing Systems*, 37:58757–58791, 2024.

Gabriel Barth-Maron, Matthew W Hoffman, David Budden, Will Dabney, Dan Horgan, Dhruva TB, Alistair Muldal, Nicolas Heess, and Timothy Lillicrap. Distributed distributional deterministic policy gradients. In *International Conference on Learning Representations*, 2018.

Marc G. Bellemare, Will Dabney, and Rémi Munos. A distributional perspective on reinforcement learning. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, pp. 449–458, 2017.

Chang Chen, Fei Deng, Kenji Kawaguchi, Caglar Gulcehre, and Sungjin Ahn. Simple hierarchical planning with diffusion. In *The Twelfth International Conference on Learning Representations*, 2024a.

Huayu Chen, Cheng Lu, Chengyang Ying, Hang Su, and Jun Zhu. Offline reinforcement learning via high-fidelity generative behavior modeling. In *The Eleventh International Conference on Learning Representations*, 2023.

Huayu Chen, Cheng Lu, Zhengyi Wang, Hang Su, and Jun Zhu. Score regularized policy optimization through diffusion behavior. In *The Twelfth International Conference on Learning Representations*, 2024b.

Tianyu Chen, Zhendong Wang, and Mingyuan Zhou. Diffusion policies creating a trust region for offline reinforcement learning. *Advances in Neural Information Processing Systems*, 37:50098–50125, 2024c.

Will Dabney, Georg Ostrovski, David Silver, and Rémi Munos. Implicit quantile networks for distributional reinforcement learning. In *International conference on machine learning*, pp. 1096–1105. PMLR, 2018a.

Will Dabney, Mark Rowland, Marc G. Bellemare, and Rémi Munos. Distributional reinforcement learning with quantile regression. In *AAAI Conference on Artificial Intelligence*, volume 32, 2018b.

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

Zihan Ding, Amy Zhang, Yuandong Tian, and Qinqing Zheng. Diffusion world model: Future modeling beyond step-by-step rollout for offline reinforcement learning. *CoRR*, 2024.

Perry Dong, Chongyi Zheng, Chelsea Finn, Dorsa Sadigh, and Benjamin Eysenbach. Value flows. In *International Conference on Learning Representations*, 2026.

Lasse Espeholt, Hubert Soyer, Remi Munos, Karen Simonyan, Vlad Mnih, Tom Ward, Yotam Doron, Vlad Firoiu, Tim Harley, Iain Dunning, et al. Impala: Scalable distributed deep-rl with importance weighted actor-learner architectures. In *International conference on machine learning*, pp. 1407–1416. PMLR, 2018.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*, 2024.

Justin Fu, Aviral Kumar, Ofir Nachum, George Tucker, and Sergey Levine. D4rl: Datasets for deep data-driven reinforcement learning. *arXiv preprint arXiv:2004.07219*, 2020.

Scott Fujimoto and Shixiang Shane Gu. A minimalist approach to offline reinforcement learning. *Advances in neural information processing systems*, 34:20132–20145, 2021.

Philippe Hansen-Estruch, Ilya Kostrikov, Michael Janner, Jakub Grudzien Kuba, and Sergey Levine. Idql: Implicit q-learning as an actor-critic method with diffusion policies. *arXiv preprint arXiv:2304.10573*, 2023.

Longxiang He, Li Shen, Linrui Zhang, Junbo Tan, and Xueqian Wang. Diffcps: Diffusion model based constrained policy search for offline reinforcement learning. *arXiv preprint arXiv:2310.05333*, 2023.

Longxiang He, Li Shen, and Xueqian Wang. AlignIQL: Policy alignment in implicit q-learning through constrained optimization. *arXiv preprint arXiv:2405.18187*, 2024.

Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016.

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

Matthew Thomas Jackson, Michael Matthews, Cong Lu, Benjamin Ellis, Shimon Whiteson, and Jakob Nicolaus Foerster. Policy-guided diffusion. In *Reinforcement Learning Conference*, 2024.

Michael Janner, Yilun Du, Joshua Tenenbaum, and Sergey Levine. Planning with diffusion for flexible behavior synthesis. In *International Conference on Machine Learning*, pp. 9902–9915. PMLR, 2022.

Bingyi Kang, Xiao Ma, Chao Du, Tianyu Pang, and Shuicheng Yan. Efficient diffusion policies for offline reinforcement learning. *Advances in Neural Information Processing Systems*, 36:67195–67212, 2023.

Roger Koenker and Jr. Bassett, Gilbert. Regression quantiles. *Econometrica*, 46(1):33–50, 1978.

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

Steven Li, Rickmer Krohn, Tao Chen, Anurag Ajay, Pulkit Agrawal, and Georgia Chalvatzaki. Learning multimodal behaviors from scratch with diffusion policy gradient. *Advances in Neural Information Processing Systems*, 37:38456–38479, 2024.

Zhixuan Liang, Yao Mu, Mingyu Ding, Fei Ni, Masayoshi Tomizuka, and Ping Luo. Adaptdiffuser: diffusion models as adaptive self-evolving planners. In *Proceedings of the 40th International Conference on Machine Learning*, pp. 20725–20745, 2023.

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

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*, pp. 22825–22855. PMLR, 2023a.

Cong Lu, Philip Ball, Yee Whye Teh, and Jack Parker-Holder. Synthetic experience replay. *Advances in Neural Information Processing Systems*, 36:46323–46344, 2023b.

Bogdan Mazoure, Thang Doan, Audrey Durand, Joelle Pineau, and R Devon Hjelm. Leveraging exploration in off-policy algorithms via normalizing flows. In *Conference on Robot Learning*, pp. 430–444. PMLR, 2020.

Seohong Park, Kevin Frans, Benjamin Eysenbach, and Sergey Levine. OGBench: Benchmarking offline goal-conditioned RL. In *The Thirteenth International Conference on Learning Representations*, 2025a.

Seohong Park, Kevin Frans, Benjamin Eysenbach, and Sergey Levine. Ogbench: Benchmarking offline goal-conditioned rl. In *The Thirteenth International Conference on Learning Representations*, 2025b.

Seohong Park, Qiyang Li, and Sergey Levine. Flow q-learning. In *Forty-second International Conference on Machine Learning*, 2025c.

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*, pp. 41163–41182, 2024.

Chao Qu, Shie Mannor, and Huan Xu. Nonlinear distributional gradient temporal-difference learning. In *International Conference on Machine Learning*, pp. 5251–5260. PMLR, 2019.

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

Rahul Singh, Keuntaek Lee, and Yongxin Chen. Sample-based distributional policy gradient. In *Learning for Dynamics and Control Conference*, pp. 676–688. PMLR, 2022.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pp. 2256–2265. pmlr, 2015.

HJ Terry Suh, Glen Chou, Hongkai Dai, Lujie Yang, Abhishek Gupta, and Russ Tedrake. Fighting uncertainty with gradients: Offline reinforcement learning via diffusion score matching. In *Conference on Robot Learning*, pp. 2878–2904. PMLR, 2023.

Richard S. Sutton and Andrew G. Barto. *Finite Markov Decision Processes*, pp. 67–69. MIT Press, 2nd edition, 2018.

Denis Tarasov, Vladislav Kurenkov, Alexander Nikulin, and Sergey Kolesnikov. Revisiting the minimalist approach to offline reinforcement learning. *Advances in Neural Information Processing Systems*, 36:11592–11620, 2023a.

Denis Tarasov, Alexander Nikulin, Dmitry Akimov, Vladislav Kurenkov, and Sergey Kolesnikov. Corl: Research-oriented deep offline reinforcement learning library. *Advances in Neural Information Processing Systems*, 36:30997–31020, 2023b.

Siddarth Venkatraman, Shivesh Khaitan, Ravi Tej Akella, John Dolan, Jeff Schneider, and Glen Berseth. Reasoning with latent diffusion in offline reinforcement learning. In *The Twelfth International Conference on Learning Representations*, 2024.

Cédric Villani. *Topics in Optimal Transportation*, volume 58. American Mathematical Society, 2003.

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

Long Yang, Zhixiong Huang, Fenghao Lei, Yucun Zhong, Yiming Yang, Cong Fang, Shiting Wen, Binbin Zhou, and Zhouchen Lin. Policy representation via diffusion probability model for reinforcement learning. *arXiv preprint arXiv:2305.13122*, 2023.

Ruoqi Zhang, Ziwei Luo, Jens Sjölund, Thomas Schön, and Per Mattsson. Entropy-regularized diffusion policy with q-ensembles for offline reinforcement learning. *Advances in Neural Information Processing Systems*, 37:98871–98897, 2024.

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

Qinqing Zheng, Matt Le, Neta Shaul, Yaron Lipman, Aditya Grover, and Ricky T. Q. Chen. Guided flows for generative modeling and decision making. *CoRR*, abs/2311.13443, 2023.

A Proof of Proposition 1

Proof. For probability measures μ, ν supported on $[-B, B]$, the ∞-Wasserstein distance is

$$W_\infty(\mu, \nu) = \operatorname*{ess\,sup}_{u \in (0,1)} |F_\mu^{-1}(u) - F_\nu^{-1}(u)|. \tag{2}$$

We use $\bar{d}_\infty(Z_1, Z_2) = \sup_{s,a} W_\infty(Z_1(s,a), Z_2(s,a))$ throughout. The projection $\Pi_{W_1}^M$ is the canonical midpoint projection defined in Section 3.2.

Contraction of the Bellman operator. For the fixed policy π, the distributional Bellman operator is a γ-contraction in the maximal ∞-Wasserstein metric (Bellemare et al., 2017):

$$\bar{d}_\infty(\mathcal{T}^\pi Z_1, \mathcal{T}^\pi Z_2) \le \gamma \bar{d}_\infty(Z_1, Z_2). \tag{3}$$

Non-expansiveness of the exact quantile projection. The atoms of $\Pi_{W_1}^M \mu$ are already ordered by their quantile levels. Since both projected distributions have uniform weights,

$$\begin{aligned} W_\infty(\Pi_{W_1}^M \mu, \Pi_{W_1}^M \nu) &= \max_{1 \le i \le M} |F_\mu^{-1}(\hat{\tau}_i) - F_\nu^{-1}(\hat{\tau}_i)| \\ &\le W_\infty(\mu, \nu). \end{aligned} \tag{4}$$

The final inequality holds at every midpoint because the canonical generalized inverse is left-continuous on $(0, 1)$, so these pointwise differences are bounded by their essential supremum. Taking the supremum over state-action pairs and combining the two inequalities gives

$$\begin{aligned} &\bar{d}_\infty\big(\Pi_{W_1}^M \mathcal{T}^\pi Z_1, \Pi_{W_1}^M \mathcal{T}^\pi Z_2\big) \\ &\quad \le \bar{d}_\infty(\mathcal{T}^\pi Z_1, \mathcal{T}^\pi Z_2) \le \gamma \bar{d}_\infty(Z_1, Z_2). \end{aligned} \tag{5}$$

Invariant space and fixed point. The reward bound implies $|r + \gamma z| \le R_{\max} + \gamma B = B$ for $z \in [-B, B]$, so the Bellman operator preserves this support interval. The projection also preserves the interval and produces M uniformly weighted atoms. Thus $\hat{\mathcal{T}}_M$ maps $\mathcal{Z}_M$ into itself. Representing each member of $\mathcal{Z}_M$ by its ordered atom locations identifies $\bar{d}_\infty$ with the uniform norm on bounded measurable M-tuples. The ordering and support constraints form a closed subset of this complete function space. Therefore $(\mathcal{Z}_M, \bar{d}_\infty)$ is complete, and Banach's fixed-point theorem gives a unique fixed point of $\hat{\mathcal{T}}_M$ and convergence of its exact iterates. □

B Implementation Details

B.1 Hyperparameters and Architecture

Across all experiments, we set the learning rates for the online flow critic Z_ϕ^{flow} and the one-step critic q_ψ to 1×10^{-4} and 3×10^{-4}, respectively, and update the target flow critic using the standard target-network rule. By default, we use a common critic sample count of $M = 51$ for the teacher return samples and student quantile levels, following the classical quantile setting of Dabney et al. (2018b). We set the exploration constant to $\delta = 0.3$, the Huber loss parameter to $\kappa = 1$, and use 10 integration steps for the flow models. For all neural network components, which are realized as $[512, 512, 512, 512]$-sized multi-layer perceptrons (MLPs) . To ensure a fair and direct comparison, we also adopt the identical hyperparameters for the actor, including the behavioral cloning coefficient α_{BC}, as specified in FQL. We provide the complete list of hyperparameters in Table 3.

Due to computational resource constraints, our hyperparameter selection largely adheres to the configurations established in FQL. A comprehensive list of general hyperparameters is provided in Table 3, while task-specific settings are detailed in Table 4. For key baseline methods, we made the following specific adjustments:

- **For IQL and IFQL**, we set exploring expectile value to be 0.9, while the AWR inverse temperature, α, is tuned on a per-environment basis (see Table 4).
- **For ReBRAC**, the actor BC coefficient, α_1, and the critic BC coefficient, α_2, are set on a per-environment basis, as specified in Table 4.

- **For IDQL**, the expectile value, τ, is configured to 0.7 for OGBench locomotion and Adroit tasks, and to 0.9 for OGBench manipulation and AntMaze tasks. The number of action samples, N, is also tuned individually per task (see Table 4). Consistent with its original protocol, the agent is trained for 3 million steps (1.5 million for the value function).
- **For CAC**, the Q-loss coefficient, η, is determined based on the environment-specific values listed in Table 4.
- **For C51**, the atom count $N \in \{51, 101\}$ varies by tasks, with the exact assignments detailed in Table 4.
- **For IQN**, the threshold parameter $\kappa \in \{0.7, 0.8, 0.9, 0.95\}$ is customized for each dataset, relying on the configurations reported in Table 4.
- **For EWFM**, we use `alpha=3` throughout. The Q-weight temperature `q_alpha` is set to 20 for navigation tasks and 1 for manipulation tasks. For D4RL Adroit, we enable `normalize_energy=True` and set `bc_lambda=5`, which we found to be critical for tasks with negative Q-values. The per-domain settings are listed in Table 4.

Hyperparameter	Value
Online flow critic learning rate	0.0001
One-step critic learning rate	0.0003
Critic sample count M	51
Exploration constant δ	0.3
Huber loss parameter κ	1
Actor learning rate	0.0003
Optimizer	Adam (Adam et al., 2015)
Gradient steps	1000000 (default), 500000 (Offline D4RL, pixel-based OGBench)
Minibatch size	256
MLP dimensions	$[512, 512, 512, 512]$
Nonlinearity	GELU (Hendrycks & Gimpel, 2016)
Target network smoothing coefficient	0.005
Discount factor γ	0.99 (default), 0.995 (`antmaze-giant`, `humanoidmaze`, `antsoccer`)
Image augmentation probability	0.5
Flow steps	10
Flow time sampling distribution	$\mathcal{U}([0, 1])$
BC coefficient α	See Table 4.

Table 3: **Hyperparameters for DFC.**

For the state-based OGBench tasks, DFC is trained for one million gradient steps, whereas for the D4RL and pixel-based OGBench tasks, training is conducted for 500K steps, in alignment with the FQL protocol. The agent's performance is evaluated every 100K steps over 50 episodes. For the offline-to-online experiments, we report the performance metrics at both one million and two million total steps. For reporting final scores on OGBench, we follow the protocol of FQL and average the success rates over the last three evaluation epochs (i.e., 800K, 900K, and 1M steps for state-based tasks; 300K, 400K, and 500K for pixel-based tasks). For D4RL, consistent with Tarasov et al. (2023b), we report the performance at the final training epoch.

In accordance with the official implementation of OGBench, for all pixel-based experiments, we use a compact version of the IMPALA encoder (Espeholt et al., 2018) with random-shift augmentation.

B.2 Datasets and Environments

Our empirical evaluation is conducted on two prominent offline reinforcement learning benchmarks: OGBench and D4RL. These benchmarks provide a diverse set of environments and tasks designed to test the limits of offline RL algorithms, particularly their ability to stitch together suboptimal trajectories into effective policies. Detailed results are provided in Table 6.

OGBench (Park et al., 2025a) features a semi-sparse reward structure where the reward is defined as the negative count of remaining subtasks required to achieve a fixed goal. This structure presents distinct challenges: locomotion tasks typically involve a single subtask (reaching a goal) with binary rewards (0 or -1), while manipulation tasks can involve up to 16 sequential subtasks. The datasets are collected from policies executing random tasks, resulting in highly suboptimal data

Task	IQL α	ReBRAC (α_1, α_2)	IDQL N	CAC η	IFQL N	FQL α	C51 N	IQN κ	EWFM $(q_\alpha, \lambda_{\text{bc}})$	DFC α	DFC M
antmaze-large-navigate-v0	10	$(0.003, 0.01)$	32	1	32	10	101	0.7	$(20, 1)$	10	101
antmaze-giant-navigate-v0	10	$(0.003, 0.01)$	32	1	32	10	101	0.8	$(20, 1)$	10	51
humanoidmaze-medium-navigate-v0	10	$(0.01, 0.01)$	32	0.03	32	30	51	0.7	$(20, 1)$	30	51
humanoidmaze-large-navigate-v0	10	$(0.01, 0.01)$	32	1	32	30	101	0.8	$(20, 1)$	30	51
antsoccer-arena-navigate-v0	1	$(0.01, 0.01)$	32	1	64	10	51	0.8	$(20, 1)$	10	51
cube-single-play-v0	1	$(1, 0)$	32	0.003	32	300	51	0.8	$(1, 1)$	300	51
cube-double-play-v0	0.3	$(0.1, 0)$	32	0.3	32	300	101	0.9	$(1, 1)$	300	101
scene-play-v0	10	$(0.1, 0.01)$	32	0.3	32	300	51	0.95	$(1, 1)$	300	101
puzzle-3x3-play-v0	10	$(0.3, 0.01)$	32	0.01	32	1000	101	0.8	$(1, 1)$	300	101
puzzle-4x4-play-v0	3	$(0.3, 0.01)$	32	0.01	32	1000	101	0.95	$(1, 1)$	300	51
antmaze-umaze-v2	10	$(0.003, 0.002)$	32	0.01	32	10	51	0.9	$(20, 1)$	10	51
antmaze-umaze-diverse-v2	10	$(0.003, 0.001)$	32	0.01	32	10	51	0.9	$(20, 1)$	10	51
antmaze-medium-play-v2	10	$(0.001, 0.0005)$	32	0.01	32	10	51	0.8	$(20, 1)$	10	51
antmaze-medium-diverse-v2	10	$(0.001, 0.0)$	32	0.01	32	10	51	0.8	$(20, 1)$	10	51
antmaze-large-play-v2	10	$(0.002, 0.001)$	32	4.5	32	3	101	0.7	$(20, 1)$	3	51
antmaze-large-diverse-v2	10	$(0.002, 0.002)$	32	3.5	32	3	101	0.7	$(20, 1)$	3	101
pen-human-v1	3	$(0.1, 0.5)$	32	0.003	32	10000	51	0.8	$(1, 5)$	10000	101
pen-cloned-v1	3	$(0.05, 0.5)$	32	0.003	32	10000	51	0.8	$(1, 5)$	10000	51
pen-expert-v1	3	$(0.01, 0.01)$	32	0.03	32	3000	51	0.8	$(1, 5)$	3000	51
door-human-v1	3	$(0.1, 0.1)$	32	0.03	32	30000	101	0.9	$(1, 5)$	30000	101
door-cloned-v1	3	$(0.01, 0.1)$	32	0.03	128	30000	51	0.9	$(1, 5)$	30000	51
door-expert-v1	3	$(0.05, 0.01)$	32	0.03	32	30000	51	0.9	$(1, 5)$	30000	101
hammer-human-v1	3	$(0.01, 0.5)$	128	0.03	32	30000	51	0.7	$(1, 5)$	30000	51
hammer-cloned-v1	3	$(0.1, 0.5)$	32	0.003	32	10000	51	0.7	$(1, 5)$	10000	51
hammer-expert-v1	3	$(0.01, 0.01)$	32	0.03	32	30000	51	0.9	$(1, 5)$	30000	51
relocate-human-v1	3	$(0.1, 0.01)$	32	0.01	128	10000	101	0.9	$(1, 5)$	10000	51
relocate-cloned-v1	3	$(0.1, 0.01)$	64	0.01	32	30000	51	0.9	$(1, 5)$	30000	51
relocate-expert-v1	3	$(0.05, 0.01)$	32	0.003	32	30000	101	0.9	$(1, 5)$	30000	51
visual-antmaze-medium-navigate-v0	1	$(0.01, 0.003)$	-	-	32	100	-	0.9	$(20, 1)$	100	51
visual-antmaze-teleport-navigate-v0	1	$(0.01, 0.003)$	-	-	32	100	-	0.8	$(20, 1)$	100	51
visual-cube-double-play-v0	0.3	$(0.1, 0)$	-	-	32	100	-	0.9	$(1, 1)$	100	51
visual-scene-play-v0	10	$(0.1, 0.003)$	-	-	32	100	-	0.95	$(1, 1)$	100	51
visual-puzzle-3x3-play-v0	10	$(0.3, 0.01)$	-	-	32	300	-	0.8	$(1, 1)$	300	51

Table 4: **Task-specific hyperparameters.** For a detailed description of each hyperparameter, we refer the reader to Appendix B.1. To ensure experimental consistency, a single hyperparameter configuration is applied uniformly to all five tasks derived from each OGBench environment. An em-dash "-" indicates that a corresponding result is not available.

that necessitates strong stitching capabilities from the learning algorithm. OGBench also provides image-based variants of its environments, where policies operate directly from $64 \times 64 \times 3$ pixel observations. These tasks introduce additional challenges of partial observability and high-dimensional input spaces.

Our experiments leverage the following OGBench environments, categorized by input type and task domain:

- **State-Based Tasks (50 tasks from 10 environments):** These tasks require control based on proprioceptive states.
  - *Locomotion*: The agent, either a quadruped (`ant`) or a humanoid, must navigate through various maze layouts or dribble a ball to a target location.
    * `antmaze-large-navigate-v0`
    * `antmaze-giant-navigate-v0`
    * `humanoidmaze-medium-navigate-v0`
    * `humanoidmaze-large-navigate-v0`
    * `antsoccer-arena-navigate-v0`
  - *Manipulation*: A robotic arm must manipulate diverse objects. These tasks range from simple object interaction (`cube`) to long-horizon, multi-object control (`scene`) and challenges requiring combinatorial generalization (`puzzle`).
    * `cube-single-play-v0`
    * `cube-double-play-v0`
    * `scene-play-v0`
    * `puzzle-3x3-play-v0`
    * `puzzle-4x4-play-v0`
- **Image-Based Tasks (25 tasks from 5 environments):** These are the visual counterparts spanning both navigation and manipulation, requiring control solely from $64 \times 64 \times 3$ image observations. Each environment contributes 5 tasks, forming a comprehensive visual benchmark of 25 tasks in total.
  - `visual-antmaze-medium-navigate-v0`
  - `visual-antmaze-teleport-navigate-v0`
  - `visual-cube-double-play-v0`
  - `visual-scene-play-v0`
  - `visual-puzzle-3x3-play-v0`

D4RL (Fu et al., 2020) is also evaluated on 18 challenging tasks to facilitate direct comparison with a broad range of prior work. These tasks are selected to cover complex locomotion and dexterous manipulation challenges.

- **AntMaze Tasks (6 tasks):** These locomotion tasks share a similar high-level objective with their OGBench counterparts but feature different maze layouts and dataset characteristics.
  - `antmaze-umaze-v2`
  - `antmaze-umaze-diverse-v2`
  - `antmaze-medium-play-v2`
  - `antmaze-medium-diverse-v2`
  - `antmaze-large-play-v2`
  - `antmaze-large-diverse-v2`
- **Adroit Tasks (12 tasks):** These tasks demand dexterous manipulation using a high-dimensional (24-DoF) robotic hand, with objectives such as spinning a pen, opening a door, hammering a nail, and relocating an object.
  - `pen-human-v1`, `-cloned-v1`, `-expert-v1`
  - `door-human-v1`, `-cloned-v1`, `-expert-v1`
  - `hammer-human-v1`, `-cloned-v1`, `-expert-v1`
  - `relocate-human-v1`, `-cloned-v1`, `-expert-v1`

Evaluation metrics adhere to the standard protocols for each benchmark: success rates for OGBench and D4RL AntMaze tasks, and normalized returns for Adroit tasks.

C Detailed Results

In this section, we present detailed results for DFC, including comprehensive empirical analyses of the critic and actor architectures, training time, and sample efficiency. Tables 6 and 5 present the complete per-task results on state-based and image-based benchmarks, respectively.

State-based tasks. Across the 50 state-based OGBench tasks, DFC achieves the best score on the majority of tasks, particularly excelling in the manipulation domains. On `cube-single-play`, DFC attains near-perfect scores (99–100) on all 5 tasks, matched only by IQN which also reaches 97–100. On `cube-double-play`, `scene-play`, and `puzzle-3x3-play`, DFC ranks first on all or nearly all tasks. IQN emerges as a notable competitor on specific manipulation tasks (e.g., 81 on `cube-double-play-task5`, 100 on `scene-play-task1`), but lacks consistency across domains. In the navigation domains, DFC is the top performer on `antsoccer-arena-navigate` (4 out of 5 best scores) and `antmaze-large-navigate` (3 out of 5 best). On `humanoidmaze-large-navigate`, IQN achieves the best score on 3 out of 5 tasks, surpassing DFC on this domain. IQN also has a higher group mean on `cube-double-play` (42.0 vs. 40.6). On the 6 D4RL `antmaze` tasks, DFC achieves the best or tied-best score on 5 out of 6, reaching 100 on `antmaze-umaze` and 95 on `antmaze-large-diverse`. IQN performs moderately on this suite (e.g., 92 on `antmaze-umaze`, 65 on `antmaze-large-play`), but falls behind DFC on the harder tasks. On the 12 D4RL `adroit` tasks, DFC obtains the best or second-best score on 8 tasks, with notable strengths on `pen-cloned` (90), `pen-expert` (147), `door-cloned` (6), and `relocate-expert` (109). ReBRAC remains a strong competitor on `pen-human` and `hammer-expert` in this domain. EWFM (Zhang et al., 2025), a value-weighted flow matching baseline, achieves moderate results on D4RL `antmaze` (e.g., 97 on `antmaze-umaze`, 75 on `antmaze-large-play`) but largely fails outside this domain. On OGBench navigation, it scores 0 on all humanoidmaze tasks and near-zero on `antsoccer-arena-navigate`. On manipulation, it remains below 25 even on the easiest domain (`cube-single-play`) and collapses to 0 on `puzzle-3x3-play` and `puzzle-4x4-play`. Even with tuned hyperparameters (`normalize_energy=True, bc_lambda=5`), its D4RL Adroit average reaches 13.2, below DFC's 55.9. These results favor DFC over the evaluated energy-weighted baseline on navigation and manipulation tasks. Overall, DFC achieves the best domain-level average on 8 out of 12 state-based domain groups across OGBench and D4RL, demonstrating strong performance across navigation and manipulation tasks.

Image-based tasks. Table 5 reports results on 25 image-based OGBench tasks spanning 5 visual domains. DFC achieves the best score on the majority of visual tasks, leading `visual-cube-double-play` (best on all 5 tasks) and `visual-antmaze-teleport-navigate` (best on 4 out of 5). On `visual-scene-play`, DFC leads on 4 out of 5 tasks including a perfect 100 on task1, with FQL remaining competitive on task2 (86). On `visual-puzzle-3x3-play`, DFC achieves the best score on 4 out of 5 tasks, sharing the lead with IFQL on task1 (100). On `visual-antmaze-medium-navigate`, ReBRAC achieves the highest per-task scores on 4 out of 5 tasks, but DFC matches or exceeds it on task1 (87) and task5 (97), resulting in the highest domain average (88.2 vs. 87.4). EWFM shows partial competitiveness on `visual-antmaze-medium-navigate` (72) but scores 0 on `visual-cube-double-play` and near-zero on `visual-puzzle-3x3-play`, mirroring its state-based failure pattern. In summary, DFC achieves the best domain-level average on all 5 visual domains, confirming that the distributional flow critic is particularly effective for learning from high-dimensional pixel observations.

Offline-to-online fine-tuning. Beyond pure offline evaluation, Table 2 demonstrates that the representations learned by DFC transfer effectively to the online fine-tuning setting. DFC achieves the best or tied-best fine-tuned score on 4 out of 5 OGBench domains, reaching perfect scores on three of them. Most notably, on `puzzle-4x4-play`, DFC is the only method to recover a perfect score from a modest offline starting point, while all other baselines stagnate. The sole exception is `humanoidmaze-medium-navigate`, where IFQL leads. Across all settings, DFC demonstrates that its distributional critic provides not only stronger offline policies but also a more favorable initialization for subsequent online learning.

Contribution of the Two-Stage Architecture. Table 7 compares DFC with the Direct Flow Critic (FC), which differentiates through the ODE during actor updates, and the Quantile-Only Critic (DC), which removes the flow teacher. On the 68 offline tasks, DFC wins on 57, ties on 8, and loses on 3

against FC; against DC it wins on 53 and ties on 15. The full design therefore combines an effective distribution-learning target with an efficient feed-forward critic for policy optimization.

Return-distribution fitting. With the same flow-actor backbone, DFC outperforms IQN in 10 of the 12 state-based domain groups (Table 1). Figure 4 provides a complementary comparison of the learned return distributions: DFC attains 1-Wasserstein errors of 0.071, 0.053, and 0.083, each less than half the corresponding error of the strongest baseline (0.162, 0.109, and 0.248). Together with the quantile-only ablation, these results demonstrate the benefit of learning the return distribution with a flow teacher and transferring it to a quantile critic for policy optimization.

Run time comparison. We present the time consumption comparison in Figure 3. DFC is only slightly slower than FQL in training speeds, while not being slower than most other methods such as IQL, FQL, FBRAC and IFQL in inference speeds.

Distributional methods. C51, IQN, and DFC use the same flow-actor backbone. DFC achieves a higher group mean than IQN in 10 of the 12 state-based domain groups and all 5 image-based groups (Table 1). The per-task results in Tables 6 and 5 detail both the gains and the remaining competitive cases for IQN.

Ensemble methods. Table 8 compares DFC with two- and four-critic FQL ensembles. DFC achieves the best or tied-best score on all 22 listed tasks, demonstrating its advantage over these scalar-ensemble alternatives.

Sample numbers. Table 10 ablates the common critic sample count $M \in \{51, 101\}$; we use M teacher return samples and M student quantile levels. Neither setting consistently dominates across D4RL Adroit manipulation tasks and OGBench state-based tasks, indicating limited sensitivity to this design choice.

Task	IQL	ReBRAC	IFQL	FQL	IQN	EWFM	DFC
visual-antmaze-medium-navigate-task1-v0(*)	78 ±9	54 ±15	81 ±3	32 ±3	62 ±7	43 ±14	**87** ±9
visual-antmaze-medium-navigate-task2-v0	90 ±3	**96** ±1	87 ±1	60 ±2	88 ±2	89 ±5	72 ±4
visual-antmaze-medium-navigate-task3-v0	80 ±6	**97** ±1	92 ±1	35 ±8	64 ±5	89 ±2	95 ±4
visual-antmaze-medium-navigate-task4-v0	89 ±4	**93** ±2	84 ±3	35 ±2	71 ±6	64 ±15	90 ±4
visual-antmaze-medium-navigate-task5-v0	84 ±2	**97** ±1	89 ±2	29 ±7	84 ±1	77 ±9	97 ±10
visual-antmaze-teleport-navigate-task1-v0(*)	5 ±2	2 ±0	7 ±4	2 ±1	2 ±1	5 ±2	**12** ±4
visual-antmaze-teleport-navigate-task2-v0	10 ±2	10 ±3	13 ±3	6 ±1	7 ±3	10 ±3	**15** ±2
visual-antmaze-teleport-navigate-task3-v0	7 ±7	4 ±1	8 ±9	9 ±4	6 ±4	25 ±6	**19** ±2
visual-antmaze-teleport-navigate-task4-v0	4 ±6	4 ±0	18 ±2	9 ±1	4 ±2	24 ±6	**19** ±6
visual-antmaze-teleport-navigate-task5-v0	2 ±1	2 ±1	4 ±2	1 ±1	2 ±1	4 ±2	**8** ±4
visual-cube-double-play-task1-v0(*)	**34** ±23	4 ±4	8 ±6	23 ±4	4 ±1	1 ±1	**34** ±7
visual-cube-double-play-task2-v0	3 ±1	0 ±0	0 ±0	0 ±0	0 ±0	0 ±0	**3** ±1
visual-cube-double-play-task3-v0	7 ±4	2 ±2	1 ±1	4 ±2	0 ±0	0 ±0	**14** ±1
visual-cube-double-play-task4-v0	2 ±1	0 ±0	0 ±0	0 ±0	0 ±0	0 ±0	**3** ±1
visual-cube-double-play-task5-v0	11 ±2	0 ±0	2 ±1	4 ±1	1 ±1	0 ±0	**12** ±3
visual-scene-play-task1-v0(*)	97 ±2	98 ±4	86 ±10	98 ±3	95 ±2	37 ±7	**100** ±1
visual-scene-play-task2-v0	21 ±16	30 ±15	0 ±0	86 ±8	79 ±15	18 ±14	**96** ±1
visual-scene-play-task3-v0	12 ±9	10 ±7	19 ±2	22 ±6	36 ±14	0 ±0	**64** ±13
visual-scene-play-task4-v0	1 ±0	0 ±0	0 ±0	1 ±1	0 ±0	0 ±0	**11** ±2
visual-scene-play-task5-v0	**0** ±0	**0** ±0	**0** ±0	**0** ±0	**0** ±0	**0** ±0	**0** ±0
visual-puzzle-3x3-play-task1-v0(*)	7 ±15	88 ±4	**100** ±0	94 ±1	84 ±1	11 ±13	**100** ±0
visual-puzzle-3x3-play-task2-v0	0 ±0	12 ±1	0 ±0	0 ±0	6 ±2	0 ±0	**23** ±2
visual-puzzle-3x3-play-task3-v0	0 ±0	1 ±1	2 ±1	0 ±0	1 ±0	0 ±0	**3** ±0
visual-puzzle-3x3-play-task4-v0	1 ±1	0 ±1	1 ±0	5 ±4	3 ±0	0 ±0	**12** ±4
visual-puzzle-3x3-play-task5-v0	0 ±0	0 ±0	0 ±0	1 ±2	1 ±0	0 ±0	**27** ±7

Table 5: **Results on image-based tasks.** The following table presents a comprehensive evaluation of our method against key baselines on 25 image-based tasks from the OGBench benchmark. Scores are reported as mean ± SD over 4 random seeds. The best scores are highlighted in **bold** and the second-best scores are underlined. The asterisk $(*)$ denotes the default task within each environment group.

Computational overhead. DFC introduces only marginal training overhead compared to FQL, thanks to our distillation design that decouples ODE solving from gradient updates; the runtime comparison is reported in Appendix C, Figure 3. Importantly, DFC incurs *zero inference overhead*: the distributional critic is used only during training, and the deployed policy is identical to a standard flow policy, matching the latency of IQL, ReBRAC, and IFQL at test time.

Task	IQL	ReBRAC	IDQL	CAC	IFQL	FQL	C51	IQN	EWFM	DFC
antmaze-large-navigate-singletask-task1-v0(*)	48 ±9	**91** ±10	0 ±0	42 ±7	24 ±17	80 ±8	0 ±0	31 ±3	74 ±6	90 ±1
antmaze-large-navigate-singletask-task2-v0	42 ±6	**88** ±4	14 ±8	1 ±1	8 ±3	57 ±10	0 ±0	0 ±0	12 ±1	74 ±4
antmaze-large-navigate-singletask-task3-v0	72 ±7	51 ±18	26 ±8	49 ±10	52 ±17	93 ±3	0 ±0	84 ±4	7 ±4	**98** ±1
antmaze-large-navigate-singletask-task4-v0	51 ±9	84 ±7	62 ±25	17 ±6	18 ±8	80 ±4	0 ±0	11 ±7	18 ±4	**89** ±2
antmaze-large-navigate-singletask-task5-v0	54 ±22	90 ±2	2 ±2	55 ±6	38 ±18	83 ±4	0 ±0	77 ±2	89 ±6	**91** ±3
antmaze-giant-navigate-singletask-task1-v0(*)	0 ±0	**27** ±22	0 ±0	0 ±0	0 ±0	4 ±5	0 ±0	0 ±0	0 ±0	25 ±12
antmaze-giant-navigate-singletask-task2-v0	1 ±1	**16** ±17	0 ±0	0 ±0	0 ±0	9 ±7	0 ±0	0 ±0	0 ±0	1 ±1
antmaze-giant-navigate-singletask-task3-v0	0 ±0	**34** ±22	0 ±0	0 ±0	0 ±0	0 ±1	0 ±0	0 ±0	0 ±0	1 ±1
antmaze-giant-navigate-singletask-task4-v0	0 ±0	5 ±12	0 ±0	0 ±0	0 ±0	14 ±23	0 ±0	0 ±0	32 ±3	**40** ±45
antmaze-giant-navigate-singletask-task5-v0	19 ±7	**49** ±22	0 ±1	0 ±0	13 ±9	16 ±28	0 ±0	2 ±2	18 ±2	30 ±10
humanoidmaze-medium-navigate-singletask-task1-v0(*)	32 ±7	16 ±9	1 ±1	38 ±19	**69** ±19	19 ±12	0 ±0	44 ±4	0 ±0	35 ±9
humanoidmaze-medium-navigate-singletask-task2-v0	41 ±9	18 ±16	1 ±1	47 ±35	85 ±11	94 ±3	0 ±0	44 ±3	0 ±0	**99** ±1
humanoidmaze-medium-navigate-singletask-task3-v0	25 ±5	36 ±13	0 ±1	83 ±18	49 ±49	74 ±18	0 ±0	67 ±6	0 ±0	**96** ±3
humanoidmaze-medium-navigate-singletask-task4-v0	0 ±1	15 ±16	1 ±1	5 ±4	1 ±1	3 ±4	0 ±0	**28** ±2	0 ±0	3 ±1
humanoidmaze-medium-navigate-singletask-task5-v0	66 ±4	24 ±20	1 ±1	91 ±5	98 ±2	97 ±2	0 ±0	51 ±11	0 ±0	**100** ±1
humanoidmaze-large-navigate-singletask-task1-v0(*)	3 ±1	2 ±1	0 ±0	1 ±1	6 ±2	7 ±6	0 ±0	**16** ±2	0 ±0	15 ±6
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
humanoidmaze-large-navigate-singletask-task3-v0	7 ±3	8 ±4	3 ±1	2 ±3	**48** ±10	11 ±7	1 ±1	17 ±1	0 ±0	19 ±5
humanoidmaze-large-navigate-singletask-task4-v0	1 ±0	1 ±1	0 ±0	0 ±1	1 ±1	2 ±3	0 ±0	**9** ±5	0 ±0	4 ±2
humanoidmaze-large-navigate-singletask-task5-v0	1 ±1	2 ±2	0 ±0	0 ±0	0 ±0	1 ±3	1 ±1	**36** ±4	0 ±0	2 ±2
antsoccer-arena-navigate-singletask-task1-v0	14 ±5	0 ±0	44 ±12	1 ±3	61 ±25	77 ±4	7 ±1	34 ±4	3 ±1	**84** ±3
antsoccer-arena-navigate-singletask-task2-v0	17 ±7	0 ±1	15 ±12	0 ±0	75 ±3	88 ±3	3 ±3	19 ±6	8 ±3	**96** ±3
antsoccer-arena-navigate-singletask-task3-v0	6 ±4	0 ±0	0 ±0	8 ±19	14 ±22	61 ±6	0 ±0	45 ±3	8 ±3	**69** ±5
antsoccer-arena-navigate-singletask-task4-v0(*)	3 ±2	0 ±0	0 ±1	0 ±0	16 ±9	39 ±6	2 ±2	41 ±4	1 ±1	**52** ±7
antsoccer-arena-navigate-singletask-task5-v0	2 ±2	0 ±0	0 ±0	0 ±0	0 ±1	36 ±9	0 ±0	29 ±6	4 ±2	**66** ±5
cube-single-play-singletask-task1-v0	88 ±3	89 ±5	95 ±2	77 ±28	79 ±4	97 ±2	4 ±3	97 ±1	13 ±3	**100** ±0
cube-single-play-singletask-task2-v0(*)	85 ±8	92 ±4	96 ±2	80 ±30	73 ±3	97 ±2	7 ±4	**100** ±0	0 ±0	**100** ±0
cube-single-play-singletask-task3-v0	91 ±5	93 ±3	99 ±1	98 ±1	88 ±4	98 ±2	18 ±8	99 ±1	62 ±5	**100** ±0
cube-single-play-singletask-task4-v0	73 ±6	92 ±3	93 ±4	91 ±2	79 ±6	94 ±3	4 ±1	**99** ±1	48 ±5	**99** ±1
cube-single-play-singletask-task5-v0	78 ±9	87 ±8	90 ±6	80 ±20	77 ±7	93 ±3	2 ±0	97 ±3	0 ±0	**100** ±0
cube-double-play-singletask-task1-v0	27 ±5	45 ±6	39 ±19	21 ±8	35 ±9	61 ±9	9 ±0	70 ±14	3 ±2	**84** ±4
cube-double-play-singletask-task2-v0(*)	1 ±1	7 ±3	16 ±10	2 ±2	9 ±5	36 ±6	0 ±0	24 ±9	0 ±0	**49** ±2
cube-double-play-singletask-task3-v0	0 ±0	4 ±1	17 ±8	3 ±1	8 ±5	22 ±5	0 ±0	25 ±6	1 ±1	**31** ±1
cube-double-play-singletask-task4-v0	0 ±0	1 ±1	0 ±1	0 ±1	1 ±1	5 ±2	0 ±0	10 ±1	0 ±0	**11** ±4
cube-double-play-singletask-task5-v0	4 ±3	4 ±2	1 ±1	3 ±2	17 ±6	19 ±10	0 ±0	**81** ±8	0 ±0	28 ±11
scene-play-singletask-task1-v0	94 ±3	95 ±2	**100** ±0	**100** ±1	98 ±3	**100** ±0	17 ±3	**100** ±0	19 ±31	**100** ±0
scene-play-singletask-task2-v0(*)	12 ±3	50 ±13	33 ±14	50 ±40	0 ±0	76 ±9	2 ±1	1 ±0	19 ±2	**96** ±3
scene-play-singletask-task3-v0	32 ±7	55 ±16	94 ±4	49 ±16	54 ±19	98 ±1	0 ±1	94 ±2	2 ±2	**100** ±0
scene-play-singletask-task4-v0	0 ±1	3 ±3	4 ±3	0 ±0	0 ±0	5 ±1	2 ±1	3 ±1	0 ±0	**20** ±9
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
puzzle-3x3-play-singletask-task1-v0	33 ±6	**97** ±4	52 ±12	**97** ±2	94 ±3	90 ±4	5 ±0	71 ±3	0 ±0	**97** ±1
puzzle-3x3-play-singletask-task2-v0	4 ±3	1 ±1	0 ±1	0 ±0	1 ±2	16 ±5	0 ±0	2 ±2	0 ±0	**36** ±1
puzzle-3x3-play-singletask-task3-v0	3 ±2	3 ±1	0 ±0	0 ±0	0 ±0	10 ±3	0 ±0	0 ±0	0 ±0	**17** ±3
puzzle-3x3-play-singletask-task4-v0(*)	2 ±1	2 ±1	0 ±0	0 ±0	0 ±0	16 ±5	0 ±0	0 ±0	0 ±0	**21** ±5
puzzle-3x3-play-singletask-task5-v0	3 ±2	5 ±3	0 ±0	0 ±0	0 ±0	16 ±3	0 ±0	0 ±0	1 ±1	**29** ±5
puzzle-4x4-play-singletask-task1-v0	12 ±2	26 ±4	48 ±5	44 ±10	49 ±9	34 ±8	0 ±0	41 ±2	1 ±1	**68** ±5
puzzle-4x4-play-singletask-task2-v0	7 ±4	12 ±4	14 ±5	0 ±0	4 ±4	16 ±5	0 ±0	12 ±4	0 ±0	**24** ±2
puzzle-4x4-play-singletask-task3-v0	9 ±3	15 ±3	34 ±5	29 ±12	**50** ±14	18 ±5	0 ±0	45 ±7	0 ±0	49 ±3
puzzle-4x4-play-singletask-task4-v0(*)	5 ±2	10 ±3	**26** ±6	1 ±1	21 ±11	11 ±3	0 ±0	23 ±2	0 ±0	20 ±3
puzzle-4x4-play-singletask-task5-v0	4 ±1	7 ±3	**24** ±11	0 ±0	2 ±2	7 ±3	0 ±0	16 ±6	0 ±0	14 ±4
antmaze-umaze-v2	77	98	94	66 ±5	92 ±6	96 ±2	52 ±3	99 ±1	97 ±2	**100** ±0
antmaze-umaze-diverse-v2	54	84	80	66 ±11	62 ±12	89 ±5	67 ±3	65 ±4	97 ±2	**96** ±0
antmaze-medium-play-v2	66	**90**	84	49 ±24	56 ±15	78 ±7	1 ±1	53 ±1	82 ±3	85 ±3
antmaze-medium-diverse-v2	74	84	**85**	0 ±1	60 ±25	71 ±13	0 ±0	43 ±2	23 ±7	**85** ±3
antmaze-large-play-v2	42	52	64	0 ±0	55 ±9	84 ±7	0 ±0	66 ±7	75 ±4	**91** ±1
antmaze-large-diverse-v2	30	64	68	0 ±0	64 ±8	83 ±4	0 ±0	63 ±3	77 ±2	**95** ±1
pen-human-v1	78	**103**	76 ±10	64 ±8	71 ±12	53 ±6	69 ±8	69 ±3	2 ±3	61 ±7
pen-cloned-v1	83	**103**	64 ±7	56 ±10	80 ±11	74 ±11	67 ±9	80 ±11	15 ±3	90 ±7
pen-expert-v1	128	**152**	140 ±6	103 ±9	139 ±5	142 ±6	110 ±3	118 ±19	45 ±13	147 ±1
door-human-v1	3	−0	6 ±2	5 ±2	**7** ±2	0 ±0	0 ±0	0 ±0	0 ±0	6 ±3
door-cloned-v1	3	0	0 ±0	1 ±0	2 ±2	2 ±1	0 ±0	0 ±0	0 ±0	**6** ±5
door-expert-v1	**107**	106	105 ±1	98 ±3	104 ±2	104 ±1	104 ±1	105 ±0	21 ±13	105 ±0
hammer-human-v1	2	0	2 ±1	2 ±0	3 ±1	1 ±1	3 ±1	2 ±1	0 ±0	**4** ±1
hammer-cloned-v1	2	5	2 ±1	1 ±1	2 ±1	11 ±9	0 ±0	0 ±0	0 ±0	**14** ±13
hammer-expert-v1	129	**134**	125 ±4	92 ±11	117 ±9	125 ±3	122 ±1	121 ±7	3 ±3	128 ±0
relocate-human-v1	**0**	**0**	**0** ±0	**0** ±0	**0** ±0	**0** ±0	**0** ±0	**0** ±0	0 ±0	**0** ±0
relocate-cloned-v1	0	**2**	−0 ±0	−0 ±0	−0 ±0	−0 ±0	0 ±0	0 ±0	0 ±0	1 ±0
relocate-expert-v1	106	108	107 ±1	93 ±6	104 ±3	107 ±1	103 ±0	103 ±0	72 ±1	**109** ±0

Table 6: **Results on state-based tasks.** The following table presents a comprehensive evaluation of our method against key baselines across 68 state-based tasks from the OGBench and D4RL benchmarks. Scores are reported as mean ± SD over 8 random seeds. The best scores are highlighted in **bold** and the second-best scores are underlined. The asterisk (∗) denotes the default task within each environment group.

Task	FQL	FC	DC	DFC
antmaze-large-navigate-singletask-task1-v0(*)	80 ±8 → 100 ±0	86 ±3 → 100 ±0	88 ±1 → 100 ±0	90 ±1 → 100 ±0
antmaze-large-navigate-singletask-task2-v0	57 ±10 → 88 ±2	72 ±16 → 88 ±3	69 ±8 → 88 ±1	74 ±4 → 89 ±1
antmaze-large-navigate-singletask-task3-v0	93 ±3 → 100 ±0	96 ±6 → 100 ±0	96 ±2 → 100 ±0	98 ±1 → 100 ±0
antmaze-large-navigate-singletask-task4-v0	80 ±4 → 97 ±1	86 ±4 → 98 ±1	88 ±2 → 98 ±0	89 ±2 → 99 ±1
antmaze-large-navigate-singletask-task5-v0	83 ±4 → 99 ±1	90 ±6 → 100 ±1	91 ±3 → 100 ±1	91 ±3 → 100 ±1
antmaze-giant-navigate-singletask-task1-v0(*)	4 ±5 → 97 ±2	16 ±21 → 98 ±3	22 ±12 → 94 ±1	25 ±12 → 98 ±2
antmaze-giant-navigate-singletask-task2-v0	9 ±7 → 98 ±0	0 ±1 → 98 ±0	1 ±1 → 100 ±0	1 ±1 → 99 ±1
antmaze-giant-navigate-singletask-task3-v0	0 ±1 → 0 ±0	0 ±0 → 0 ±0	1 ±0 → 3 ±4	1 ±1 → 10 ±3
antmaze-giant-navigate-singletask-task4-v0	14 ±23 → 95 ±1	38 ±5 → 94 ±1	39 ±44 → 94 ±1	40 ±45 → 99 ±1
antmaze-giant-navigate-singletask-task5-v0	16 ±28 → 100 ±0	0 ±6 → 100 ±0	16 ±7 → 100 ±0	30 ±10 → 100 ±0
humanoidmaze-medium-navigate-singletask-task1-v0(*)	19 ±12 → 22 ±13	40 ±17 → 0 ±1	34 ±9 → 34 ±11	35 ±9 → 59 ±32
humanoidmaze-medium-navigate-singletask-task2-v0	94 ±3 → 99 ±1	98 ±3 → 98 ±1	98 ±1 → 98 ±1	99 ±1 → 100 ±0
humanoidmaze-medium-navigate-singletask-task3-v0	74 ±18 → 79 ±16	98 ±2 → 0 ±2	90 ±10 → 90 ±4	96 ±3 → 99 ±1
humanoidmaze-medium-navigate-singletask-task4-v0	3 ±4 → 13 ±19	2 ±1 → 4 ±2	2 ±1 → 34 ±3	3 ±1 → 66 ±24
humanoidmaze-medium-navigate-singletask-task5-v0	97 ±2 → 77 ±40	100 ±3 → 100 ±4	99 ±1 → 99 ±0	100 ±1 → 100 ±0
humanoidmaze-large-navigate-singletask-task1-v0(*)	7 ±6 → 0 ±0	0 ±6 → 0 ±0	9 ±5 → 9 ±10	15 ±6 → 16 ±3
humanoidmaze-large-navigate-singletask-task2-v0	0 ±0 → 0 ±0	0 ±0 → 0 ±0	0 ±0 → 0 ±0	0 ±0 → 0 ±0
humanoidmaze-large-navigate-singletask-task3-v0	11 ±7 → 26 ±35	18 ±16 → 12 ±4	16 ±7 → 32 ±1	19 ±5 → 67 ±3
humanoidmaze-large-navigate-singletask-task4-v0	2 ±3 → 0 ±0	0 ±0 → 0 ±0	2 ±1 → 5 ±3	4 ±2 → 20 ±14
humanoidmaze-large-navigate-singletask-task5-v0	1 ±3 → 0 ±0	0 ±0 → 0 ±0	1 ±1 → 1 ±0	1 ±2 → 3 ±2
antsoccer-arena-navigate-singletask-task1-v0	77 ±4 → 99 ±1	74 ±16 → 98 ±1	80 ±6 → 100 ±1	84 ±3 → 100 ±1
antsoccer-arena-navigate-singletask-task2-v0	88 ±3 → 95 ±2	90 ±7 → 94 ±3	92 ±3 → 94 ±1	96 ±3 → 100 ±0
antsoccer-arena-navigate-singletask-task3-v0	61 ±6 → 91 ±3	60 ±18 → 90 ±4	65 ±8 → 90 ±1	69 ±5 → 94 ±2
antsoccer-arena-navigate-singletask-task4-v0(*)	39 ±6 → 71 ±4	50 ±21 → 70 ±6	50 ±9 → 68 ±1	52 ±7 → 82 ±4
antsoccer-arena-navigate-singletask-task5-v0	36 ±9 → 69 ±5	50 ±8 → 72 ±6	57 ±5 → 72 ±8	66 ±5 → 86 ±1
cube-single-play-singletask-task1-v0	97 ±2 → 100 ±0	98 ±7 → 100 ±0	98 ±2 → 100 ±0	100 ±0 → 100 ±0
cube-single-play-singletask-task2-v0(*)	97 ±2 → 100 ±0	100 ±1 → 100 ±0	100 ±0 → 100 ±0	100 ±0 → 100 ±0
cube-single-play-singletask-task3-v0	98 ±2 → 100 ±0	98 ±1 → 100 ±0	100 ±0 → 100 ±0	100 ±0 → 100 ±0
cube-single-play-singletask-task4-v0	94 ±3 → 100 ±0	96 ±4 → 100 ±0	97 ±1 → 100 ±0	99 ±1 → 100 ±0
cube-single-play-singletask-task5-v0	93 ±3 → 99 ±1	94 ±4 → 100 ±1	97 ±1 → 98 ±0	100 ±0 → 100 ±0
cube-double-play-singletask-task1-v0	61 ±9 → 97 ±3	78 ±20 → 96 ±4	80 ±8 → 96 ±0	84 ±4 → 100 ±0
cube-double-play-singletask-task2-v0(*)	36 ±6 → 95 ±2	50 ±6 → 94 ±3	49 ±3 → 94 ±1	49 ±2 → 98 ±2
cube-double-play-singletask-task3-v0	22 ±5 → 95 ±3	26 ±7 → 94 ±4	29 ±3 → 98 ±0	31 ±1 → 100 ±1
cube-double-play-singletask-task4-v0	5 ±2 → 29 ±23	8 ±4 → 32 ±33	9 ±3 → 19 ±23	11 ±4 → 34 ±16
cube-double-play-singletask-task5-v0	19 ±10 → 97 ±4	24 ±18 → 98 ±6	28 ±11 → 96 ±2	28 ±11 → 100 ±1
scene-play-singletask-task1-v0	100 ±0 → 100 ±0	100 ±1 → 100 ±0	100 ±1 → 100 ±0	100 ±0 → 100 ±0
scene-play-singletask-task2-v0(*)	76 ±9 → 100 ±0	92 ±6 → 100 ±0	94 ±4 → 96 ±1	96 ±3 → 100 ±0
scene-play-singletask-task3-v0	98 ±1 → 100 ±0	100 ±3 → 100 ±0	99 ±1 → 100 ±0	100 ±0 → 100 ±0
scene-play-singletask-task4-v0	5 ±1 → 0 ±0	6 ±3 → 0 ±0	12 ±6 → 13 ±8	20 ±9 → 20 ±12
scene-play-singletask-task5-v0	0 ±0 → 0 ±0	0 ±0 → 0 ±0	0 ±0 → 0 ±0	0 ±0 → 0 ±0
puzzle-3x3-play-singletask-task1-v0	90 ±4 → 97 ±2	90 ±13 → 100 ±0	91 ±5 → 100 ±0	97 ±1 → 100 ±0
puzzle-3x3-play-singletask-task2-v0	16 ±5 → 0 ±0	8 ±4 → 0 ±0	21 ±2 → 21 ±25	36 ±1 → 37 ±1
puzzle-3x3-play-singletask-task3-v0	10 ±3 → 0 ±0	8 ±6 → 0 ±0	12 ±4 → 12 ±10	17 ±3 → 16 ±1
puzzle-3x3-play-singletask-task4-v0(*)	16 ±5 → 0 ±0	14 ±6 → 0 ±0	16 ±5 → 17 ±13	21 ±5 → 22 ±5
puzzle-3x3-play-singletask-task5-v0	16 ±3 → 0 ±0	20 ±7 → 0 ±0	23 ±5 → 23 ±17	29 ±5 → 30 ±6
puzzle-4x4-play-singletask-task1-v0	34 ±8 → 100 ±0	20 ±7 → 100 ±0	44 ±5 → 100 ±0	68 ±5 → 100 ±0
puzzle-4x4-play-singletask-task2-v0	16 ±5 → 0 ±0	14 ±3 → 0 ±0	18 ±2 → 18 ±14	24 ±2 → 24 ±3
puzzle-4x4-play-singletask-task3-v0	18 ±5 → 100 ±0	16 ±8 → 100 ±0	31 ±5 → 100 ±0	49 ±3 → 100 ±0
puzzle-4x4-play-singletask-task4-v0(*)	11 ±3 → 53 ±50	8 ±1 → 60 ±7	14 ±2 → 60 ±47	20 ±3 → 100 ±0
puzzle-4x4-play-singletask-task5-v0	7 ±3 → 0 ±0	4 ±3 → 0 ±0	9 ±3 → 9 ±7	14 ±4 → 13 ±3
antmaze-umaze-v2	96 ±2 → 99 ±1	98 ±4 → 100 ±1	99 ±2 → 98 ±0	100 ±0 → 100 ±0
antmaze-umaze-diverse-v2	89 ±5 → 100 ±0	80 ±8 → 100 ±0	90 ±4 → 100 ±0	96 ±0 → 100 ±0
antmaze-medium-play-v2	78 ±7 → 98 ±2	74 ±14 → 98 ±3	78 ±5 → 98 ±0	85 ±3 → 100 ±0
antmaze-medium-diverse-v2	71 ±13 → 95 ±1	70 ±8 → 96 ±1	76 ±4 → 96 ±0	85 ±3 → 99 ±2
antmaze-large-play-v2	84 ±7 → 93 ±6	86 ±7 → 94 ±8	89 ±4 → 86 ±0	91 ±1 → 98 ±0
antmaze-large-diverse-v2	83 ±4 → 89 ±2	86 ±3 → 88 ±3	90 ±2 → 88 ±3	95 ±1 → 97 ±1
pen-human-v1	53 ±6 → 134 ±2	50 ±21 → 135 ±3	58 ±12 → 135 ±2	61 ±7 → 135 ±2
pen-cloned-v1	74 ±11 → 142 ±8	81 ±13 → 143 ±11	82 ±7 → 134 ±8	90 ±7 → 144 ±6
pen-expert-v1	142 ±6 → 158 ±3	142 ±9 → 157 ±5	144 ±3 → 155 ±2	147 ±1 → 161 ±1
door-human-v1	0 ±0 → 83 ±1	0 ±0 → 83 ±1	4 ±2 → 83 ±3	6 ±3 → 82 ±2
door-cloned-v1	2 ±1 → 78 ±20	2 ±2 → 69 ±27	6 ±2 → 64 ±23	6 ±5 → 82 ±22
door-expert-v1	104 ±1 → 106 ±0	105 ±2 → 106 ±0	105 ±1 → 106 ±0	105 ±0 → 106 ±0
hammer-human-v1	1 ±1 → 114 ±5	1 ±2 → 112 ±6	2 ±1 → 112 ±6	4 ±1 → 121 ±8
hammer-cloned-v1	11 ±9 → 129 ±8	3 ±7 → 126 ±11	12 ±11 → 126 ±9	14 ±13 → 130 ±7
hammer-expert-v1	125 ±3 → 132 ±0	126 ±4 → 132 ±0	127 ±1 → 132 ±0	128 ±0 → 133 ±0
relocate-human-v1	0 ±0 → 10 ±8	0 ±0 → 11 ±11	0 ±0 → 11 ±10	0 ±0 → 10 ±6
relocate-cloned-v1	−0 ±0 → 31 ±27	0 ±0 → 44 ±35	1 ±0 → 28 ±33	1 ±0 → 36 ±19
relocate-expert-v1	107 ±1 → 106 ±1	107 ±2 → 107 ±2	108 ±2 → 108 ±3	109 ±0 → 110 ±0

Table 7: **Offline-to-online results and ablation of the DFC critic architecture.** Scores are reported as mean ± SD over 8 seeds. We compare the full model against its key components: a Direct Flow Critic without distillation (FC) and a Quantile-Only Critic without the flow target (DC). FQL is included for baseline performance reference. The → symbol separates offline and fine-tuned scores. The performance to the left of → is the result under offline setting while that to the right is the result after one million steps of online fine-tuning. The asterisk $(*)$ denotes the default task within each environment group.

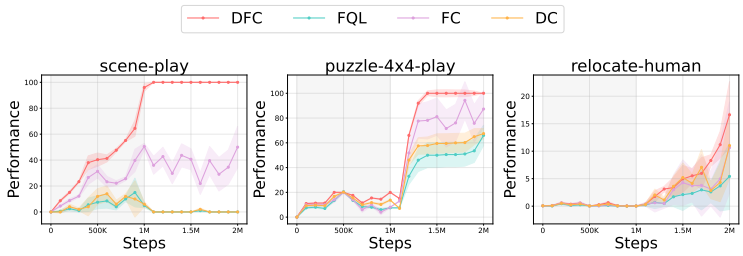


Figure 2: **Learning curves for offline-to-online RL.** Curves show the mean across 8 seeds, with colored bands indicating ± one SD. The gray background marks the offline phase.

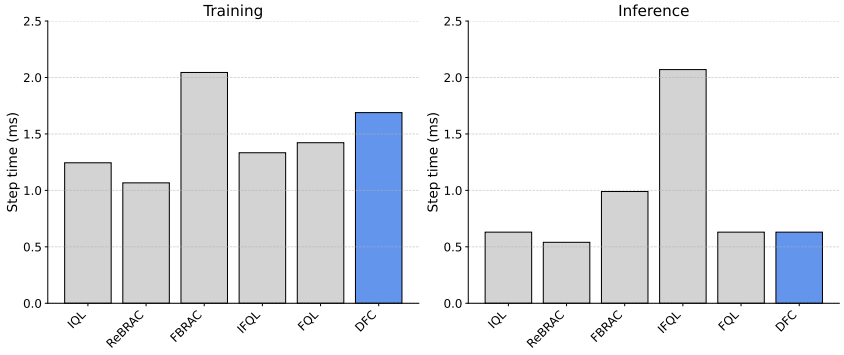


Figure 3: **Run time comparison.** DFC is only slightly slower than FQL in training speeds, while not being slower than most other methods in inference speeds.

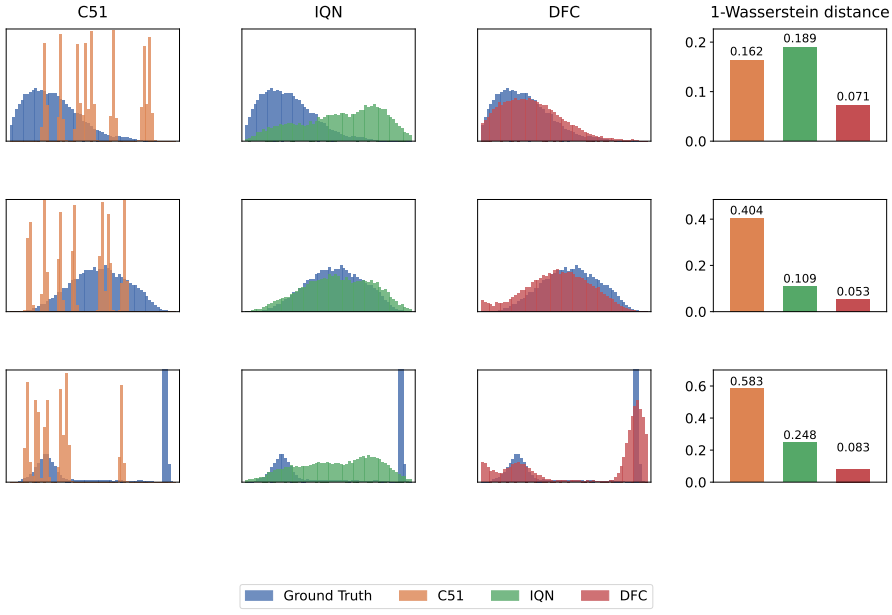


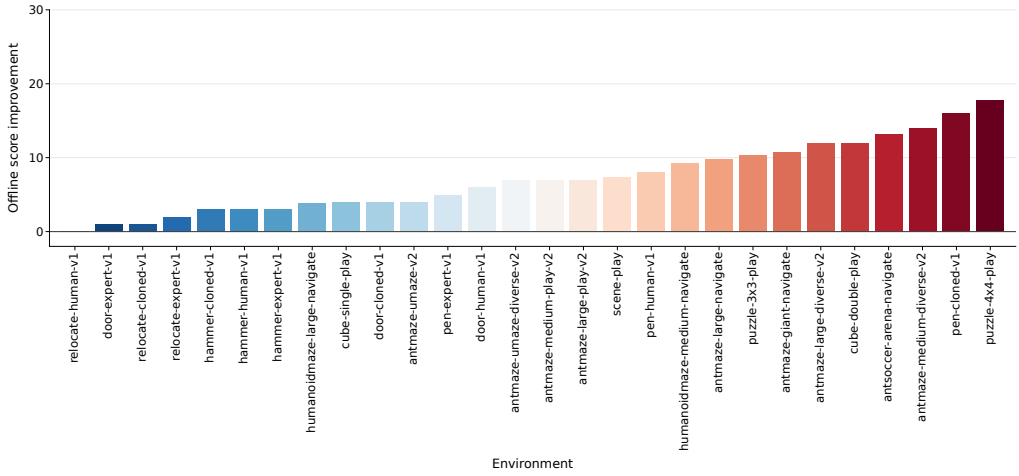
Figure 4: **Return distribution visualization.** We visualize the return distribution of C51, IQN and DFC. The displayed DFC distributions more closely match the Monte Carlo reference, with 1-Wasserstein errors below half those of the strongest baseline in all three comparisons.

Task	FQL ($\#Q = 2$)	FQL ($\#Q = 4$)	DFC
pen-human-v1	53 ± 6	39 ± 12	$\mathbf{61} \pm 7$
pen-cloned-v1	74 ± 11	76 ± 15	$\mathbf{90} \pm 7$
pen-expert-v1	142 ± 6	145 ± 2	$\mathbf{147} \pm 1$
door-human-v1	6 ± 3	0 ± 0	$\mathbf{6} \pm 3$
door-cloned-v1	2 ± 1	1 ± 1	$\mathbf{6} \pm 5$
door-expert-v1	104 ± 1	105 ± 0	$\mathbf{105} \pm 0$
hammer-human-v1	3 ± 1	1 ± 1	$\mathbf{4} \pm 1$
hammer-cloned-v1	11 ± 9	4 ± 1	$\mathbf{14} \pm 13$
hammer-expert-v1	125 ± 3	127 ± 1	$\mathbf{128} \pm 0$
relocate-human-v1	0 ± 0	0 ± 0	$\mathbf{0} \pm 0$
relocate-cloned-v1	0 ± 0	0 ± 0	$\mathbf{1} \pm 0$
relocate-expert-v1	107 ± 1	109 ± 1	$\mathbf{109} \pm 0$
antmaze-large-navigate-v0	80 ± 8	56 ± 12	$\mathbf{90} \pm 1$
antmaze-giant-navigate-v0	4 ± 4	17 ± 4	$\mathbf{25} \pm 12$
humanoidmaze-medium-navigate-v0	19 ± 12	22 ± 14	$\mathbf{35} \pm 9$
humanoidmaze-large-navigate-v0	7 ± 6	6 ± 5	$\mathbf{14} \pm 6$
antsoccer-arena-navigate-v0	39 ± 6	47 ± 6	$\mathbf{52} \pm 7$
cube-single-play-v0	97 ± 2	98 ± 2	$\mathbf{100} \pm 0$
cube-double-play-v0	36 ± 6	46 ± 3	$\mathbf{49} \pm 2$
scene-play-v0	76 ± 9	83 ± 4	$\mathbf{96} \pm 3$
puzzle-3x3-play-v0	16 ± 5	19 ± 5	$\mathbf{21} \pm 5$
puzzle-4x4-play-v0	11 ± 3	6 ± 3	$\mathbf{20} \pm 3$

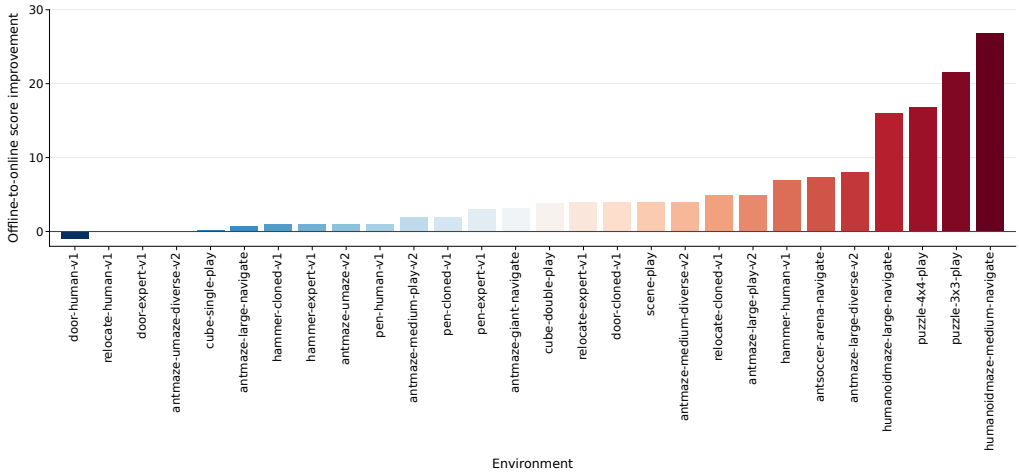
Table 8: **Ablation study of ensemble methods.** Scores are reported as mean $\pm$ SD over 8 seeds. We compare the DFC model against FQL with ensembled value models. DFC achieves the best or tied-best score on all listed tasks. The best scores are highlighted in **bold**.

Task	ReBRAC		IQL	
	Vanilla	DF Critic	Vanilla	DF Critic
antmaze-large-navigate-singletask-v0	91 ± 10	91 ± 1	48 ± 9	67 ± 4
antmaze-giant-navigate-singletask-v0	27 ± 22	74 ± 5	0 ± 0	0 ± 0
humanoidmaze-medium-navigate-singletask-v0	16 ± 9	23 ± 1	32 ± 7	43 ± 3
humanoidmaze-large-navigate-singletask-v0	2 ± 1	5 ± 10	3 ± 1	5 ± 1
antsoccer-arena-navigate-singletask-v0	0 ± 0	23 ± 10	3 ± 2	14 ± 1
cube-single-play-singletask-v0	92 ± 4	100 ± 0	85 ± 8	100 ± 0
cube-double-play-singletask-v0	7 ± 3	8 ± 0	1 ± 1	4 ± 1
scene-play-singletask-v0	50 ± 13	99 ± 1	12 ± 3	64 ± 10
puzzle-3x3-play-singletask-v0	2 ± 1	98 ± 0	2 ± 1	4 ± 1
puzzle-4x4-play-singletask-v0	10 ± 3	3 ± 10	5 ± 2	11 ± 2

Table 9: **Ablation study of critic structure.** Scores are reported as mean $\pm$ SD over 8 seeds. We integrate the DFC critic into ReBRAC and IQL. DFC improves ReBRAC on 8 of 10 tasks and IQL on 9 of 10, with one tie for each algorithm.



(a) Offline score improvement over FQL



(b) Offline-to-online score improvement over FQL

Figure 5: **Task-level details for the opening performance overview.** Table-based plots corresponding to Figure 1(b–c), sorted independently by DFC minus FQL score. The 28 comparisons comprise 10 OGBench groups (five tasks per group), six D4RL AntMaze tasks, and twelve D4RL Adroit tasks. Offline scores are taken from Table 6, averaging the five task means within each OGBench group. Fine-tuned scores are taken from Table 7, averaging the five task means within each OGBench group without rounding the resulting differences to integers. OGBench and AntMaze use success scores; Adroit uses normalized returns. Colors follow the original blue-to-red rank gradient.

Task	$M = 51$	$M = 101$
pen-human-v1	61 ± 7	$\mathbf{84} \pm 3$
pen-cloned-v1	$\mathbf{90} \pm 7$	77 ± 5
pen-expert-v1	$\mathbf{147} \pm 1$	130 ± 12
door-human-v1	6 ± 3	$\mathbf{8} \pm 2$
door-cloned-v1	$\mathbf{6} \pm 5$	1 ± 1
door-expert-v1	105 ± 0	$\mathbf{106} \pm 0$
hammer-human-v1	4 ± 1	3 ± 1
hammer-cloned-v1	$\mathbf{14} \pm 13$	1 ± 1
hammer-expert-v1	$\mathbf{128} \pm 0$	123 ± 1
relocate-human-v1	$\mathbf{0} \pm 0$	$\mathbf{0} \pm 0$
relocate-cloned-v1	$\mathbf{1} \pm 0$	0 ± 0
relocate-expert-v1	$\mathbf{109} \pm 0$	107 ± 1
antmaze-large-navigate-singletask-v0	90 ± 1	$\mathbf{93} \pm 2$
antmaze-giant-navigate-singletask-v0	$\mathbf{25} \pm 12$	20 ± 6
humanoidmaze-medium-navigate-singletask-v0	$\mathbf{35} \pm 9$	33 ± 8
humanoidmaze-large-navigate-singletask-v0	14 ± 6	14 ± 5
antsoccer-arena-navigate-singletask-v0	$\mathbf{52} \pm 7$	50 ± 3
cube-single-play-singletask-v0	100 ± 0	100 ± 0
cube-double-play-singletask-v0	49 ± 2	$\mathbf{74} \pm 8$
scene-play-singletask-v0	96 ± 3	$\mathbf{99} \pm 1$
puzzle-3x3-play-singletask-v0	21 ± 5	$\mathbf{52} \pm 3$
puzzle-4x4-play-singletask-v0	$\mathbf{20} \pm 3$	18 ± 5

Table 10: **Ablation study of critic sample count.** Scores are reported as mean $\pm$ SD over 8 seeds. We vary the common sample count $M \in \{51, 101\}$ used for the teacher return samples and student quantile levels. The best scores are highlighted in **bold**.

D Additional Related Work

Concurrent work on flow-based value learning. Concurrent with our work, Value Flows (Dong et al., 2026) and floq (Agrawalla et al., 2026) investigate flow-based value learning. Value Flows models the full return distribution using distributional flow matching. It further estimates return variance through a flow derivative ODE and uses this information to reweight learning toward transitions with greater return uncertainty. floq studies iterative computation for scalar Q-function approximation: it trains a velocity field toward bootstrapped scalar value targets and varies the integration depth to scale computation. It also trains a noise-independent, distilled scalar critic for policy extraction, avoiding differentiation through the integrated flow critic. DFC couples a distributional flow teacher to a quantile-conditioned one-step critic for optimizing a flow policy. These methods share the use of flow models for value learning, with different emphases on return-distribution modeling, uncertainty-based weighting, and iterative value computation.