GUIDED FLOW POLICY: LEARNING FROM HIGH-VALUE ACTIONS IN OFFLINE REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Offline reinforcement learning often relies on behavior regularization that enforces policies to remain close to the dataset distribution. However, such approaches fail to distinguish between high-value and low-value actions. We introduce Guided Flow Policy (GFP), which couples a multi-step flow-matching policy with a distilled one-step actor. The actor directs the flow policy to focus on cloning high-value actions from the dataset rather than imitating all state-action pairs indiscriminately. In turn, the flow policy constrains the actor to remain aligned with the dataset's best transitions while maximizing the critic. This mutual guidance enables GFP to achieve state-of-the-art performance across 129 tasks from the OGBench, Minari, and D4RL benchmarks, with substantial gains on suboptimal datasets and challenging tasks.

1 INTRODUCTION

Offline Reinforcement Learning (RL) aims to learn effective policies from static datasets without further interaction with the environment S. Lange (2012); Ernst et al. (2005). This paradigm is important in domains such as robotics and logistics, where online exploration can be unsafe or costly. However, standard off-policy algorithms such as DDPG Lillicrap et al. (2015) and SAC Haarnoja et al. (2018), which are successful in online RL, tend to underperform in offline settings since the RL agent cannot interact with the environment. The main challenge is extrapolation error, corresponding to the inability to properly evaluate out-of-distribution actions Wu et al. (2019); Fujimoto et al. (2019); Kumar et al. (2019; 2020).

Two main lines of work have been proposed to address this challenge. The first one focuses on learning a critic without querying the values of actions outside the dataset Kostrikov et al. (2021). The second one, known as the Behavior-Regularized Actor–Critic (BRAC) family, mitigates these errors by forcing the learned policy to stay "close" to the unknown behavior policy that generated the dataset Fujimoto & Gu (2021); Tarasov et al. (2023); Jaques et al. (2019); Laroche et al. (2019); Wu et al. (2019). The key idea is that out-of-distribution state–action pairs are especially vulnerable to Q-value overestimation, while staying near the empirical distribution reduces extrapolation errors. Minimalist variants achieve this by simply adding a behavior cloning (BC) loss to the policy and/or value updates with respect to dataset actions Fujimoto & Gu (2021); Tarasov et al. (2023). Although this approach improves stability, it also raises a trade-off: regularizing too strictly to a potentially suboptimal dataset action may restrict the policy from exploiting higher-reward actions contained in the dataset.

Recent progress in generative modeling offers new opportunities. Flow and diffusion-based models Lipman et al. (2022); Ho et al. (2020); Song et al. (2020); Chi et al. (2023); Janner et al. (2022); Wang et al. (2022); Zhang et al. (2025) can capture complex, multimodal action distributions. However, they come at the cost of high computational overhead: iterative sampling slows inference, and recursive backpropagation complicates critic optimization. To address these challenges, Park et al. (2025) proposed a flow-matching BC model distilled into a one-step policy that also optimizes the critic, enabling expressive policy learning without the need for recursive backpropagation and iterative sampling at inference. Despite these advances, a central limitation remains: the flow-based

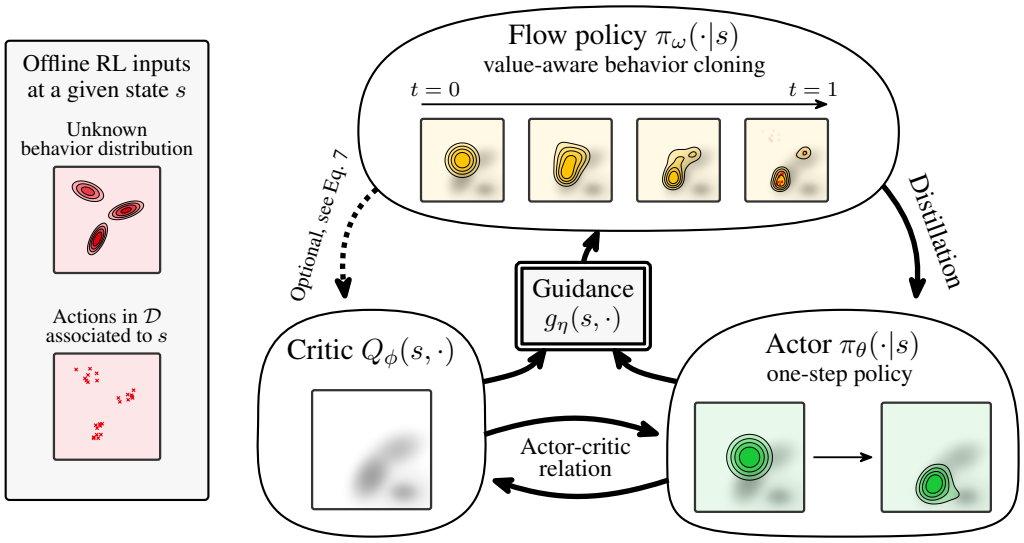


Figure 1: **Overview of the Guided Flow Policy framework.** GFP consists of three main components: (i) in yellow, a multi-step flow policy π_ω trained via value-aware BC using the guidance term g_η, (ii) in green, a one-step actor π_θ distilled from the flow policy, and (iii) in gray, a critic Q_ϕ guiding action evaluation. π_ω regularizes the actor toward high-value actions from the dataset; in turn, the actor shapes the flow and optimizes the critic following the actor–critic approach. The different components of the figure are introduced throughout the paper. Each drawing represents the probability distribution of actions $a \in \mathcal{A}$ of a policy, in a current state s, except for the gray ones, where it is the value of actions $a \in \mathcal{A}$ in state s, according to the critic.

BC component, similar to standard BC, does not incorporate reward information. An overview comparing the characteristics of prior works is presented in Tab. 1.

We build upon Fujimoto & Gu (2021); Park et al. (2025) and propose **Guided Flow Policy (GFP)**, a dual-policy framework with a bidirectional guidance mechanism between a multi-step flow-matching policy, termed Value-aware Behavior Cloning (VaBC), and a distilled one-step actor. VaBC acts as a distributional regularizer for the actor, encouraging it to remain within the support of the behavior policy. However, unlike standard behavior cloning, VaBC leverages the actor and its critic to prioritize cloning high-value actions from the dataset, rather than indiscriminately imitating all state-action pairs as done in standard BC methods. In turn, the actor optimizes the critic while being distilled toward VaBC, allowing it to align with the dataset's high-value actions in a given state, while maximizing expected returns. Fig. 1 illustrates the GFP framework in the offline RL setting.

Our contributions are threefold: (i) we introduce Guided Flow Policy (GFP), an efficient yet simple method inspired by BRAC, leveraging behavior cloning on the dataset's most promising transitions; (ii) we extensively evaluate GFP on 129 tasks from standard offline RL benchmarks, showing strong performances with substantial gains on suboptimal datasets and challenging tasks; and (iii) we re-assess two previous state-of-the-art offline RL algorithms on these benchmarks, highlighting the critical role of hyperparameter choices and subtle implementation details, aligned in the spirit with the retrospective analysis provided in Tarasov et al. (2023).

Table 1: **Characteristics of offline-RL methods.**

	Approach to avoid out-of-distribution issue	Handles suboptimal data	Expressive actor
IQL Kostrikov et al. (2021)	Critic trained only on dataset actions	✗	✗
TD3+BC Fujimoto & Gu (2021) ReBRAC Tarasov et al. (2023)	Actor regularized toward ↪ dataset actions	✗	✗
FQL Park et al. (2025)	↪ learned behavior cloning policy	✗	✓
GFP (ours)	↪ learned value-aware behavior cloning policy	✓	✓

2 BACKGROUND

Actor-critic framework in RL. RL problems are typically formalized as a Markov Decision Process (MDP) Sutton et al. (1998); Konda & Tsitsiklis (1999), defined by the tuple $(\mathcal{S}, \mathcal{A}, p, r, \rho, \gamma)$. Here, $\mathcal{S}$ denotes the state space, $\mathcal{A}$ the action space, p the transition dynamics, r the reward function, ρ the initial state distribution, and $\gamma \in [0, 1)$ the discount factor. The behavior of the agent is governed by a policy π, mapping states to probability distributions over actions. The objective is to maximize the expected discounted return $\mathbb{E}_{a_t \sim \pi(\cdot|s_t)} \left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)\right]$, i.e., the expected cumulative reward when following π in the MDP. In general, the policy π, also referred to as the actor, is trained jointly with a critic Q, which approximates the state–action value function. The Q-function is defined as $Q^\pi(s, a) = \mathbb{E}_{a \sim \pi(\cdot|s)} \left[\sum_{t=0}^{\infty} \gamma^t r\left(s_t, a_t\right) \mid s_0 = s, a_0 = a\right]$, estimating the expected return after taking action a in state s and subsequently following π.

Both actor and critic are parametrized as neural networks, with parameters θ and ϕ respectively, and optimized by alternating gradient descent steps on the two objectives:

$$\mathcal{L}^{\mathcal{A}}(\theta) = \mathbb{E}_{s \sim \mathcal{D},\, a_\theta \sim \pi_\theta(\cdot|s)} \big[- Q_\phi(s, a_\theta) \big], \tag{1}$$

$$\mathcal{L}^{\mathcal{C}}(\phi) = \mathbb{E}_{(s,a,r,s') \sim \mathcal{D},\, a' \sim \pi_\theta(\cdot|s')} \left[\left(Q_\phi(s, a) - r - \gamma Q_{\bar{\phi}}(s', a') \right)^2 \right], \tag{2}$$

where $\mathcal{L}^{\mathcal{A}}$ and $\mathcal{L}^{\mathcal{C}}$ refer to the actor and critic losses, respectively, and $\mathcal{D}$ is the set of transitions (s, a, r, s') collected during training. $Q_{\bar{\phi}}$ denotes a second target Q-function parameterized by a slowly updated set of weights $\bar{\phi}$, maintained via Polyak averaging, a common stabilization technique in actor–critic methods.

Minimalist approaches in offline RL. In offline RL, the agent learns exclusively from a static dataset $\mathcal{D}$, consisting of transitions (s, a, r, s') generated by an unknown behavior policy. In a given state s, the distribution of actions of such a behavior policy is illustrated on the left of Fig. 1. This introduces a key challenge compared to the online setting: the distributional shift S. Lange (2012); Kumar et al. (2019); Konda & Tsitsiklis (1999). Indeed, since the learned policy π_θ may select actions outside the dataset's support, value estimates for such out-of-distribution actions can be inaccurate Kumar et al. (2020); Fujimoto & Gu (2021). The BRAC approach addresses this issue by constraining the policy π_θ to remain close to the behavior policy Jaques et al. (2019); Kumar et al. (2019). However, as emphasized in Fujimoto & Gu (2021), there is no fundamental justification for preferring one divergence or distance metric over another for this purpose. A simple and effective choice is to add a BC term directly into the actor objective. Incorporating this into the actor–critic framework, the actor loss in Eq. 1 becomes:

$$\mathcal{L}^{\mathcal{A}}(\theta) = \mathbb{E}_{(s,a) \sim \mathcal{D}, a_\theta \sim \pi_\theta(\cdot|s)} \Big[- Q_\phi(s, a_\theta) + \alpha \overbrace{\|a_\theta - a\|^2}^{\text{BC term}} \Big], \tag{3}$$

where α is a hyperparameter that balances between exploiting high Q-values and staying close to the behavior policy. This objective encourages actions that both achieve high-expected returns and remain within the support of the dataset. The critic loss in Eq. 2 remains unchanged.

Behavior cloning with flow matching. Flow Matching (FM) Lipman et al. (2022) is a generative modeling framework that learns a continuous-time transformation, or flow, which maps a simple base distribution (in this work, a standard Gaussian) to a target data distribution. This transformation is defined through a family of intermediate, time-dependent distributions governed by an ordinary differential equation (ODE).

In the context of BC, FM is extended to a conditional setting, where the goal is to approximate a behavior policy π_ω underlying the dataset $\mathcal{D}$. This is achieved by learning a state and time dependent velocity field $v_\omega : [0, 1] \times \mathcal{S} \times \mathbb{R}^d \to \mathbb{R}^d$ that governs the dynamics of a flow, where d is the action dimension. This flow $\psi_\omega(t, s, z)$ is the solution of the family of ODEs characterized by:

$$\forall s \in \mathcal{S}, \quad \frac{d}{dt}\psi_\omega(t, s, z) = v_\omega(t, s, \psi_\omega(t, s, z)), \quad \psi_\omega(0, s, z) = z. \tag{4}$$

This flow, conditioned on the state s, maps noise samples $z \sim \mathcal{N}(0, I_d)$ into actions distributed according to $\pi_\omega(\cdot \mid s)$.

While sophisticated conditioning strategies can help enhance expressiveness (e.g., classifier-free guidance Ho & Salimans (2022)), we adopt in this work the simplest variant of conditional flow

Algorithm 1: Guided Flow Policy (GFP)

function ***Integrate*** $\mu_\omega(s, z)$
  // Explicit discrete Euler integration with M steps
  for $t = 0, 1, \ldots, M-1$ **do**
    $z \leftarrow z + \frac{1}{M} v_\omega(t/M, s, z)$
  return z
while *not converged* **do**
  Sample $\{(s, a, r, s')\} \sim \mathcal{D}$
  // Step 1 -- Train critic Q_ϕ
  $z' \sim \mathcal{N}(0, I_d), \quad a' = \mu_{\bar{\theta}}(s', z')$
  Update ϕ to minimize $\mathbb{E}[(Q_\phi(s, a) - r - \gamma Q_{\bar{\phi}}(s', a'))^2]$
  // Step 2 -- Train the distilled one-step actor π_θ
  $z \sim \mathcal{N}(0, I_d), \quad a^{\pi_\theta} = \mu_\theta(s, z),$
  $a^{\pi_\omega} = \mu_\omega(s, z)$ // Using the Integrate-μ_ω function, Line. 1
  Compute $\lambda = \frac{1}{\frac{1}{N}\sum |Q_\phi(s, a^{\pi_\theta})|}$ // Stop gradient a^{π_θ}
  Update θ to minimize $\mathbb{E}[-\lambda Q_\phi(s, a^{\pi_\theta}) + \alpha \|a^{\pi_\theta} - a^{\pi_\omega}\|_2^2]$
  // Step 3 -- Train the value-aware BC policy π_ω
  Compute $g_\eta(s, a) = \frac{\exp\left(\frac{\lambda}{\eta} Q_\phi(s, a)\right)}{\exp\left(\frac{\lambda}{\eta} Q_\phi(s, a^{\pi_\theta})\right) + \exp\left(\frac{\lambda}{\eta} Q_\phi(s, a)\right)}$ // Stop gradient a^{π_θ}
  $a_t = (1-t)\epsilon + ta$, with $\epsilon \sim \mathcal{N}(0, I_d)$ and $t \sim \mathcal{U}([0, 1])$
  Update θ to minimize $\mathbb{E}\left[g(s, a) \|v_\omega(t, s, a_t) - (a - \epsilon)\|_2^2\right]$
Output: $\pi_\theta, \pi_\omega, Q_\phi$

matching Holderrieth & Erives (2025). We further employ the optimal transport variant of FM, which uses linear interpolation with uniformly sampled time points Lipman et al. (2022). For $(s, a) \sim \mathcal{D}$, $\epsilon \sim \mathcal{N}(0, I_d)$, and $t \sim \mathcal{U}([0, 1])$, we define the interpolated point $a_t = (1-t)\epsilon + ta$, whose target velocity is $a - \epsilon$. The velocity field v_θ is then trained by least-squares regression toward this reference, yielding the conditional flow-matching BC loss Holderrieth & Erives (2025):

$$\mathcal{L}^{\text{FM-BC}}(\omega) = \mathbb{E}_{(s,a)\sim\mathcal{D}, \epsilon\sim\mathcal{N}(0,I_d), t\sim\mathcal{U}([0,1])} \Big[\|v_\omega(t, s, a_t) - (a - \epsilon)\|_2^2 \Big]. \tag{5}$$

Once the velocity field is learned, the corresponding flow $\psi_\omega : [0, 1] \times \mathcal{S} \times \mathbb{R}^d \to \mathcal{A}$ defines an approximation of the behavior policy. At inference, an action is obtained by sampling a random noise $z \sim \mathcal{N}(0, I_d)$ and integrating the flow from 0 to 1 using an ODE solver (e.g., an explicit Euler method). We denote by $\mu_\omega(s, z) := \psi_\omega(1, s, z)$ the value of the integrated flow at time 1. In this way, behavior cloning can be naturally expressed as conditional flow matching in the action space.

Flow policy for offline RL. Following the idea of Diffusion Q-Learning Wang et al. (2022), a straightforward way to train a flow policy for offline RL is to replace the BC term in the actor loss (Eq. 3) with the flow-matching BC loss (Eq. 5). However, the iterative sampling procedure makes training expensive, due to recursive backpropagation through the actor loss, and also results in slower inference at test time. To mitigate these limitations, Park et al. (2025) suggests distilling the iterative flow-matching BC policy into a one-step policy that directly maximizes the critic.

3 Guided Flow Policy

We now detail the GFP algorithm that builds on top of Fujimoto & Gu (2021); Park et al. (2025). GFP integrates a Value-aware Behavior Cloning (VaBC) flow policy with a distilled one-step actor through bidirectional guidance. VaBC leverages the actor and the critic to selectively clone high-value dataset actions, providing more targeted regularization than standard behavior cloning. The distilled actor, in turn, maximizes the critic while avoiding recursive backpropagation and iterative sampling. GFP is composed of three main components: the critic Q_ϕ, the actor π_θ, and the VaBC policy π_ω. The complete algorithm is presented in Algo. 1 and the approach is illustrated in Fig. 1.

Step 1 – Learning the critic Q_ϕ. The critic is trained using the Bellman mean-squared loss:

$$\mathcal{L}^{\mathcal{C}}(\phi) = \mathbb{E}_{(s,a,r,s')\sim\mathcal{D},\, a'\sim\pi_\theta(\cdot|s')} \Big[\big(Q_\phi(s, a) - \underbrace{(r + \gamma Q_{\bar{\phi}}(s', a'))}_{\text{Bellman target } y} \big)^2 \Big], \tag{6}$$

where $Q_{\bar{\phi}}$ denotes the target network. $y(s, r, s') := r + \gamma Q_{\bar{\phi}}(s', a')$ corresponds to the standard Bellman target in actor-critic methods, which we use by default in this work. Yet, since VaBC is designed to prioritize cloning the most promising dataset actions for a given state, we have also

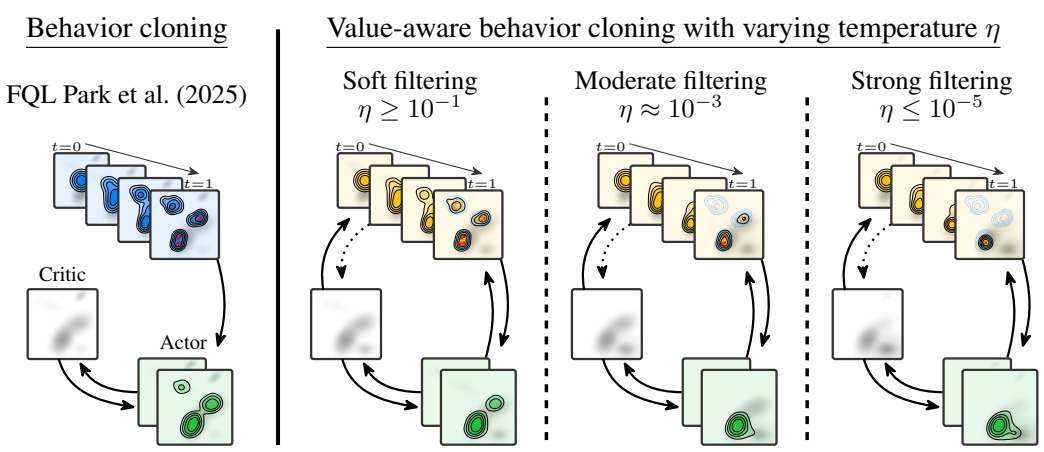


Figure 2: **Comparison of behavior cloning under different levels of guidance. Left:** Prior work (e.g., FQL, Park et al. (2025)) uses no filtering, indiscriminately imitating all state-action pairs. **Right:** In contrast, our method introduces a temperature-controlled guidance mechanism, as shown in Eq. 10, resulting in VaBC. At high temperatures, the guidance is weak, so the actor is influenced by many candidate actions. At moderate temperatures, the filtering becomes sharper, giving more weight to higher-value actions while still keeping enough regularization and exploration. At low temperatures, the filtering is very selective, concentrating almost exclusively on the highest-value actions according to the critic. However, excessive concentration at very low temperatures may allow the actor to escape the dataset's action distribution, as shown on the right in green, leading to critic overestimation and out-of-distribution issues. Importantly, VaBC cannot escape the dataset's action distribution even at very low temperatures, since it trains exclusively on in-distribution state-action pairs. The dashed blue contours in the final yellow drawings (first row) illustrate this constraint.

considered a more conservative variant of the Bellman target:

$$y^{\text{VaBC}}(s,r,s') = r + \frac{\gamma}{2}\Big(Q_{\bar{\phi}}(s', \mu_\theta(s',z)) + Q_{\bar{\phi}}(s', \mu_\omega(s',z))\Big), \quad z \sim \mathcal{N}(0, I_d), \tag{7}$$

where $\mu_\theta(s',z)$ denotes the action from the actor and $\mu_\omega(s',z)$ the action from the VaBC policy. Here, as mentioned in Sec. 2 and outlined in Line 1 of Alg. 1, $\mu(s,z)$ is the action sampled from $\pi(.|s)$ with initial input noise z. The Bellman target $y^{\text{VaBC}}(s,r,s')$ corresponds to an averaging between two estimates of the Q-value: $Q_{\bar{\phi}}(s', \mu_\theta(s',z))$ which can overestimate the real Q-value; and $Q_{\bar{\phi}}(s', \mu_\omega(s',z)$ which can underestimate the real Q-value. This choice can lead to substantial performance improvements in certain situations, as observed in Sec. 4.

Step 2 – Learning the actor π_θ. The actor π_θ is trained to maximize the Q-function while distilling the distribution of valuable actions learned by π_ω. This is achieved by minimizing the following objective:

$$\mathcal{L}^{\mathcal{A}}(\theta) = \mathbb{E}_{s\sim\mathcal{D}, z\sim\mathcal{N}(0,I_d)}\Big[-\lambda Q_\phi\left(s, \mu_\theta(s,z)\right) + \alpha \|\mu_\theta(s,z) - \mu_{\bar{\omega}}(s,z)\|_2^2\Big]. \tag{8}$$

As already stated in step 1, $\mu_c(s,z)$ is the action sampled from $\pi_c(.|s)$ with initial input noise z, where $c = \theta$ or ω. The normalization term $\lambda = \frac{1}{\frac{1}{N}\sum |Q(s,a)|}$ is based on the average absolute value of Q, estimated over mini-batches rather than over the entire dataset Fujimoto & Gu (2021).

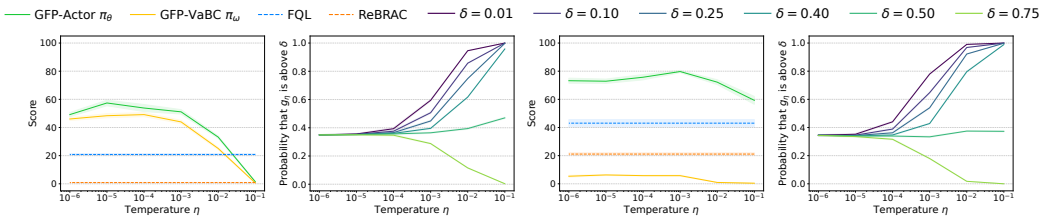
The distillation term encourages the actor to stay close to VaBC. In this way, the actor learns to select actions that maximize return while avoiding out-of-distribution actions, as it is constrained to remain near the support of high-value dataset behaviors.

Step 3 – Learning the flow policy π_ω. The VaBC policy π_ω is optimized via a flow-matching objective weighted by a value-aware guiding function:

$$\mathcal{L}^{\text{VaBC}}(\omega) = \mathbb{E}_{(s,a)\sim\mathcal{D}, \epsilon\sim\mathcal{N}(0,I_d), t\sim\mathcal{U}([0,1])}\Big[g_\eta(s,a)\, \|v_\omega(t,s,a_t) - (a-\epsilon)\|_2^2\Big], \tag{9}$$

where

$$g_\eta(s,a) := \frac{\exp\left(\frac{\lambda}{\eta} Q_\phi(s,a)\right)}{\exp\left(\frac{\lambda}{\eta} Q_\phi(s,a)\right) + \exp\left(\frac{\lambda}{\eta} Q_\phi(s, \mu_\theta(s,z))\right)}, \quad z \sim \mathcal{N}(0, I_d). \tag{10}$$



(a) Puzzle score (b) Puzzle $\mathbb{P}(g_\eta > \delta)$ (c) Cube score (d) Cube $\mathbb{P}(g_\eta > \delta)$

Figure 3: **Temperature analysis on challenging OGBench Puzzle (left) and Cube (right) tasks with suboptimal data. Plots (a) and (c):** performance scores across temperature values η for our GFP method (Actor π_θ and VaBC π_ω) compared to baselines (FQL, ReBRAC) on puzzle-4x4-noisy-task3 and cube-double-noisy-task2. **Plots (b) and (d):** probability that the guidance term g_η is above different thresholds δ as a function of temperature, illustrating how temperature controls the sharpness of value-guided filtering.

Intuitively, for a given state-action tuple (s, a) sampled from the dataset $\mathcal{D}$, $g_\eta(s, a)$ compares the quality between the dataset action a and a proposal of the actor $\mu_\theta(s, z)$. If the dataset action has a higher Q-value, this implies that $g_\eta(s, a) > 0.5$, placing greater emphasis on cloning it. Conversely, if the dataset action is worse, $g_\eta(s, a) < 0.5$, then it reduces its influence. This ensures that VaBC selectively clones high-value dataset behaviors. This makes sense because the actor itself is constrained to remain close to the dataset's action distribution.

Here, λ is the same Q-normalization factor used in the actor loss, ensuring consistent scaling across components. The parameter $\eta > 0$ is a temperature hyperparameter that controls the sharpness of the weighting: small η makes $g_\eta(s, a)$ more selective, while large η smooths the weighting. Importantly, since $g_\eta(s, a) \in (0, 1)$, VaBC avoids degeneracy during early training when the critic is unreliable, ensuring stable learning.

To the best of our knowledge, this is the first integration of value-aware behavior cloning into a BRAC framework, explicitly mitigating distributional shift by guiding the policy toward the most promising dataset actions.

Analysis of the guidance term. In Fig. 2, we illustrate how the temperature parameter controls value-guided filtering, balancing dataset fidelity with value exploitation. Lower temperatures sharpen the filter, shifting the policy from broadly imitating the dataset to emphasizing higher-value actions. Moderate values achieve the best trade-off, prioritizing promising actions while preserving diversity. In contrast, excessively low temperatures over-concentrate the VaBC policy, destabilizing training and degrading the critic by pushing the actor out of distribution.

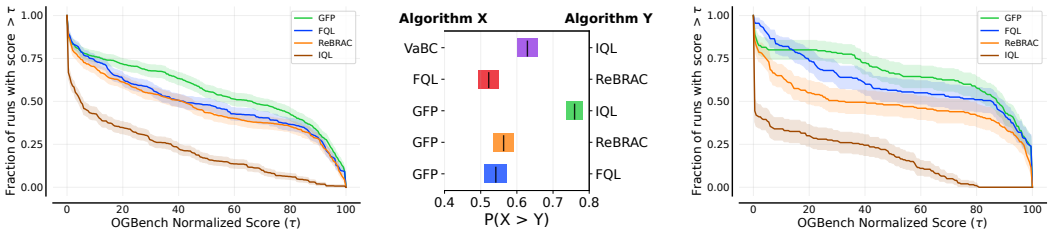
4 Experiments

We conducted extensive experiments over OGBench Park et al. (2024), D4RL Fu et al. (2020), and Minari Younis et al. (2024) benchmarks, evaluating our GFP, and prior state-of-the-art methods, IQL Kostrikov et al. (2021), ReBRAC Tarasov et al. (2023) and FQL Park et al. (2025), where needed, leading to about 13 000 runs. Our JAX-based implementation of GFP will be released after the rebuttal phase. It can complete one training run in under 30 minutes on modern GPUs.

4.1 Suboptimal datasets

We first study the impact of the temperature-controlled guidance mechanism. In Fig. 3, GFP is evaluated with varying temperature η on two challenging *noisy* benchmark tasks from OGBench, characterized as *highly suboptimal data* according to Park et al. (2024). The presence of low-quality demonstrations makes selective action emphasis decisive for effective learning, as shown in Fig. 3.

Figs. 3a and 3c demonstrate the advantages of moderate temperatures. Very low temperatures cause training instability due to over-concentration on narrow action sets, while very high temperatures fail to provide sufficient filtering of suboptimal actions. As the temperature decreases, VaBC π_ω performance improves, confirming that the value-guided filtering mechanism successfully emphasizes higher-value actions, until the temperature is too low.



(a) (b) (c)

Figure 4: **OGBench analysis**: (a) Performance profiles for 90 tasks showing the fraction of tasks where each algorithm achieves a score above threshold τ (b) Probability of improvement $\mathrm{P}(\mathrm{X} > \mathrm{Y})$ with 95% confidence intervals on the same 90 tasks (c) Performance profiles for 25 tasks from the noisy and explore environments.

Figs. 3b and 3d illustrate the filtering behavior by showing the probability that the guidance term g_η exceeds various thresholds δ (ranging from 0.01 to 0.75). At extremely low temperatures ($\eta \leq 10^{-5}$), the guidance term exhibits near-binary behavior: any slight differences between $Q(s, a)$, with a from the dataset, and $Q(s, a^{\pi_\theta})$, with $a^{\pi_\theta} \sim \pi_\theta(\cdot|s)$, in state s, result in the guidance term approaching either 1 or 0 according to Eq. 10, in this case $\mathbb{P}(g_\eta > 0.75) \approx \mathbb{P}(g_\eta > 0.01)$. As the temperature increases, the filtering becomes softer, creating more gradual transitions in the guidance values. This leads to a broader distribution of filtering probabilities across different thresholds, demonstrating how higher temperatures preserve more of the original dataset diversity. In contrast, lower temperatures create sharper distinctions between high-value and low-value actions.

4.2 Extensive offline RL benchmarks

We evaluate GFP across a comprehensive suite of robot locomotion and manipulation tasks, spanning three major benchmarks: D4RL Fu et al. (2020), its successor Minari Younis et al. (2024), and the recently proposed OGBench Park et al. (2024). For comparability with existing works, we first evaluate on D4RL's AntMaze (6 tasks) and Adroit (12 tasks). We also present results on Minari, evaluating both GFP and FQL, to facilitate the community's migration from D4RL. Minari includes all available Gym-Mujoco datasets (Hopper, HalfCheetah, and Walker, each with 3 tasks), and Adroit (12 tasks). Our most extensive evaluation focuses on OGBench, which offers substantially more complex and challenging tasks than D4RL. Following Park et al. (2025), we use the reward-based single-task variants ("-singletask") of OGBench. This yields 9 locomotion and 9 manipulation environments, each with 5 tasks, resulting in a total of 90 state-based tasks. Combined with D4RL and Minari, we evaluate on 129 tasks overall.

Tab. 2 summarizes our results grouped by environment, with detailed per-task results in the appendix (Tabs. 8, 9, 10, and 11). Together with performance profile plots and probability of improvement metrics (Fig. 4, following Agarwal et al. (2021)), it shows that GFP achieves state-of-the-art performance, with particularly substantial gains on noisy and challenging environments. For instance, on the cube-double-noisy dataset, GFP achieves an average score of 63, compared to 38 and 20 for FQL and ReBRAC, respectively. Similarly, GFP stands out in some very challenging locomotion tasks, such as humanoidmaze-large-navigate (18 vs. 7 for FQL and 13 for ReBRAC), and manipulation tasks, like cube-triple-play (16 vs. 4 for FQL and 3 for ReBRAC). For cube and humanoidmaze-medium environments, we use our conservative Bellman target y^{VaBC} defined in Eq. 7, which improves performances considerably; e.g., boosting cube-double-noisy score from 46 of 63 (see appendix, Tab. 6, for additional comparisons).

4.3 Value-aware behavior cloning

Through our bidirectional training procedure (Alg. 1), we obtain the VaBC policy π_ω as a byproduct that can also be exploited and evaluated. Since this policy is trained using only in-distribution state-action pairs from the dataset like in IQL, a direct comparison between these approaches is informative. Notably, while both methods share this fundamental in-distribution constraint, VaBC leverages the expressive power of flow matching, whereas IQL relies on Gaussian assumptions for policy extraction.

As shown in Tab. 2 and Fig. 4b, VaBC achieves good performance across benchmarks by combining the stability of BC with value-based selectivity. Rather than optimizing Q-values directly as in IQL's expectile regression, it uses critic estimates to focus on high-value actions during cloning. The strong empirical performance of this emergent VaBC policy justifies its role as a regularizer during distillation, where it steers the distilled policy toward high-value regions while avoiding the instabilities of pure Q-maximization.

4.4 Re-evaluation of prior work on OGBench

To obtain a fair comparison of GFP against prior methods, we reevaluate existing baselines on OGBench. During the development of our method, we observed that several task-specific hyperparameters (e.g., discount factor γ, minibatch size B, and critic aggregation scheme for doubled Q-learning Fujimoto et al. (2018)) have a significant impact on performance. Tab. 3 reports results under these revised settings, showing that careful tuning can substantially improve the reported scores of both ReBRAC and FQL. Since the optimal values for these hyperparameters were generally consistent across methods, we treated them as task-specific and, by default, applied the same settings to GFP, FQL, and ReBRAC (see Tabs. 4 and 5 in the appendix for detailed values).

Table 2: **Offline RL results.** GFP achieves best or near-best performance on all 129 benchmark tasks. Results are averaged over 8 seeds, with values reported from prior works Park et al. (2025); Tarasov et al. (2023); Fu et al. (2020) in *italic*, and values within 95% of the best performance are shown in bold. GFP actor π_θ represents our primary policy, while GFP RaBC π_ω is reported as a byproduct of the training procedure. Full results are provided in the appendix Tabs. 8, 9, 10, and 11.

Task Category	**Offline RL algorithms**				
	IQL	ReBRAC	FQL	**GFP** actor π_θ	**GFP** VaBC π_ω
OGBench antmaze-large-navigate-singletask (5 tasks)	*53* ± 3	**95.9** ± 0.4	88.1 ± 3.4	**93.8** ± 1.5	90.0 ± 1.3
OGBench antmaze-large-stitch-singletask (5 tasks)	30.4 ± 3.2	**89.2** ± 6.6	58.1 ± 8.7	68.9 ± 0.8	57.6 ± 3.2
OGBench antmaze-large-explore-singletask (5 tasks)	12.9 ± 1.7	82.7 ± 7.6	**87.9** ± 6.6	**91.9** ± 0.9	**89.3** ± 1.1
OGBench antmaze-giant-navigate-singletask (5 tasks)	*4* ± 1	**33.2** ± 5.7	16.3 ± 8.2	27.9 ± 8.5	0.8 ± 0.2
OGBench humanoidmaze-medium-navigate-singletask (5 tasks)	*33* ± 2	59.2 ± 12.1	*58* ± 5	**72.0** ± 2.8	35.9 ± 2.7
OGBench humanoidmaze-medium-stitch-singletask (5 tasks)	27.3 ± 2.9	61.1 ± 8.2	**63.2** ± 6.7	**66.2** ± 5.7	39.5 ± 2.1
OGBench humanoidmaze-large-navigate-singletask (5 tasks)	*2* ± 1	12.9 ± 4.2	6.5 ± 2.7	**17.8** ± 9.6	2.4 ± 1.1
OGBench antsoccer-arena-navigate-singletask (5 tasks)	*8* ± 2	55.9 ± 1.5	**60** ± 4	**57.9** ± 1.9	10.3 ± 0.7
OGBench antsoccer-arena-stitch-singletask (5 tasks)	2.8 ± 1.0	22.0 ± 1.5	28.6 ± 2.3	**30.5** ± 2.2	1.4 ± 0.3
OGBench cube-single-play-singletask (5 tasks)	*83* ± 3	*91* ± 2	***96*** ± 1	**98.8** ± 0.4	39.7 ± 4.1
OGBench cube-single-noisy-singletask (5 tasks)	53.2 ± 4.1	**98.4** ± 0.6	**100.0** ± 0.0	**100.0** ± 0.0	**99.9** ± 0.1
OGBench cube-double-play-singletask (5 tasks)	7 ± 1	12.6 ± 1.8	*29* ± 2	**47.2** ± 1.6	6.4 ± 1.0
OGBench cube-double-noisy-singletask (5 tasks)	4.5 ± 0.8	19.6 ± 2.1	38.2 ± 5.3	**63.1** ± 3.3	9.4 ± 0.8
OGBench cube-triple-play-singletask (5 tasks)	0.1 ± 0.1	2.9 ± 1.2	3.9 ± 1.5	**15.9** ± 2.0	7.6 ± 1.6
OGBench puzzle-4×4-play-singletask (5 tasks)	7 ± 1	17.1 ± 1.3	*17* ± 2	**26.1** ± 2.1	9.5 ± 1.1
OGBench puzzle-4×4-noisy-singletask (5 tasks)	0.1 ± 0.0	1.1 ± 0.3	15.6 ± 1.1	**18.8** ± 1.7	**19.3** ± 1.0
OGBench scene-play-singletask (5 tasks)	*28* ± 1	41.6 ± 3.6	***56*** ± 2	53.5 ± 2.9	**57.6** ± 1.7
OGBench scene-noisy-singletask (5 tasks)	16.0 ± 1.2	39.9 ± 2.6	**59.3** ± 1.4	**57.5** ± 0.9	**58.5** ± 1.0
D4RL antmaze (6 tasks)	*17*	*76.8*	***84*** ± 3	**83.1** ± 2.7	70.2 ± 3.0
D4RL Adroit (12 tasks)	*48*	*59*	*52* ± 1	52.8 ± 1.4	49.6 ± 1.3
Minari Adroit (12 tasks)	–	–	40.6 ± 0.4	**48.3** ± 2.3	**46.1** ± 1.7
Minari hopper (3 tasks)	–	–	79.6 ± 10.3	**91.7** ± 4.5	**91.5** ± 12
Minari halfcheetah (3 tasks)	–	–	97.8 ± 2.0	**109.1** ± 2.0	**103.1** ± 1.8
Minari walker2d (3 tasks)	–	–	**121.7** ± 1.3	**124.5** ± 0.8	**122.2** ± 1.1
Average OGBench (90 tasks)	20.7	46.4	48.9	**56.0**	N/A
Average D4RL (18 tasks)	54.0	**64.8**	**62.1**	**63.0**	N/A
Average Minari (21 tasks)	–	–	65.9	**74.1**	N/A

5 Related Work

Our work builds on key developments in offline RL. Early methods addressed distributional shift in different ways: Conservative Q-Learning (CQL) Kumar et al. (2020) penalized out-of-distribution value estimates, Implicit Q-Learning (IQL) Kostrikov et al. (2021) avoided querying such values via expectile regression, and AWR Peng et al. (2020) and AWAC Nair et al. (2020) framed offline RL as supervised learning instances. These approaches established core principles but often required complex implementations or struggled with multimodal action distributions.

A practical approach emerged with the use of BC regularization. TD3+BC Fujimoto & Gu (2021) demonstrated that adding a BC term of the form $\alpha\|a_\theta - a\|^2$ to TD3 can match state-of-the-art performance. Refining this idea, ReBRAC Tarasov et al. (2023), through careful hyperparameter tuning and architectural choices, achieved strong D4RL performance. These methods highlight that combining behavioral constraints with value optimization can be both simple and effective.

Table 3: **Impact of task-specific hyperparameters on OGBench performance.**

Task Environment	Bigger discount factor γ			
	Previously reported Park et al. (2025)		Our evaluations	
	ReBRAC	FQL	ReBRAC	FQL
antmaze-large-navigate (5 tasks)	*81* $\pm$ 5	*79* $\pm$ 3	95.9 $\pm$ 0.4	88.1 $\pm$ 3.4
humanoidmaze-large-navigate (5 tasks)	*2* $\pm$ 1	*4* $\pm$ 2	12.9 $\pm$ 4.2	6.5 $\pm$ 2.7
	Bigger minibatch size			
	Previously reported Park et al. (2025)		Our evaluations	
	ReBRAC	FQL	ReBRAC	FQL
antmaze-giant-navigate (5 tasks)	*26* $\pm$ 8	*9* $\pm$ 6	33.2 $\pm$ 5.7	16.3 $\pm$ 8.2
	Same critic aggregation for ReBRAC as used in FQL			
	Previously reported Park et al. (2025)		Our evaluations	
	ReBRAC		ReBRAC	
humanoidmaze-medium-navigate (5 tasks)	*22* $\pm$ 8		59.2 $\pm$ 12.1	
antsoccer-arena-navigate (5 tasks)	*0* $\pm$ 0		55.9 $\pm$ 1.5	
cube-double-play (5 tasks)	*12* $\pm$ 1		12.6 $\pm$ 1.8	
scene-play (5 tasks)	*41* $\pm$ 3		41.6 $\pm$ 3.6	
puzzle-4x4-play (5 tasks)	*14* $\pm$ 1		17.1 $\pm$ 1.3	

Generative models enable the learning of complex multimodal policies. Diffusion-QL Wang et al. (2022) used diffusion models to generate actions while maximizing Q-values iteratively, but at high computational cost. Flow-based methods Park et al. (2025) applied flow matching with one-step distillation, achieving strong performance efficiently. Combining flow and diffusion with RL and imitation learning is a thriving research area Janner et al. (2022); Ajay et al. (2022); Chi et al. (2023); Zheng et al. (2023); Kang et al. (2023); Chen et al. (2023); Hansen-Estruch et al. (2023); Jackson et al. (2024); Ding & Jin (2024); Jang et al. (2025).

6 Discussion and Conclusion

In this work, we revisited behavior regularization for offline RL. Conventional approaches constrain the learned policy to remain near the dataset distribution, which reduces instability but fails to distinguish between high and low-value actions. This limitation is especially problematic in suboptimal datasets, where imitating all transitions indiscriminately hinders performance.

To address this, we introduced Guided Flow Policy. GFP couples a multi-step flow-matching policy trained with value-aware behavior cloning and a distilled one-step actor through a bidirectional guidance mechanism. This design enables GFP to leverage the expressiveness of flow policies while guiding them toward high-value actions identified by the actor–critic, striking a balance between effective exploitation and robustness against distributional drift.

Our analysis provides several insights. First, although simple behavior regularized actor–critic methods, such as ReBRAC, are competitive with good hyperparameter tuning, their dependence on behavior regularization restricts their performance when data quality is imperfect. Second, while generative models such as flow or diffusion policies can represent dataset distributions more flexibly, without guidance, their expressiveness also reproduces suboptimal actions. GFP unifies these perspectives: the flow policy gains value-aware guidance from the critic, while the actor benefits from an expressive, value-aware regularization. This synergy enables GFP to consistently achieve state-of-the-art results across more than 120 offline RL tasks.

Nonetheless, GFP depends on the availability of a sufficiently accurate critic. In datasets lacking high-value actions or when the critic cannot reliably evaluate them, improvements are limited. Future research directions could explore ways to reduce reliance on the critic or extend GFP to settings with weaker or sparse reward signals.

Note on LLM usage: In this work, we only used LLMs for grammar and spelling corrections.

REFERENCES

Rishabh Agarwal, Max Schwarzer, Pablo Samuel Castro, Aaron Courville, and Marc G Bellemare. Deep reinforcement learning at the edge of the statistical precipice. *Advances in Neural Information Processing Systems*, 2021.

Anurag Ajay, Yilun Du, Abhi Gupta, Joshua Tenenbaum, Tommi Jaakkola, and Pulkit Agrawal. Is conditional generative modeling all you need for decision-making? *arXiv preprint arXiv:2211.15657*, 2022.

Huayu Chen, Cheng Lu, Zhengyi Wang, Hang Su, and Jun Zhu. Score regularized policy optimization through diffusion behavior. *arXiv preprint arXiv:2310.07297*, 2023.

Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, pp. 02783649241273668, 2023.

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

Damien Ernst, Pierre Geurts, and Louis Wehenkel. Tree-based batch mode reinforcement learning. *Journal of Machine Learning Research*, 6, 2005.

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.

Scott Fujimoto, Herke Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In *International conference on machine learning*, pp. 1587–1596. PMLR, 2018.

Scott Fujimoto, David Meger, and Doina Precup. Off-policy deep reinforcement learning without exploration. In *International conference on machine learning*, pp. 2052–2062. PMLR, 2019.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pp. 1861–1870. Pmlr, 2018.

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.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

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

Peter Holderrieth and Ezra Erives. An introduction to flow matching and diffusion models. *arXiv preprint arXiv:2506.02070*, 2025.

Matthew Thomas Jackson, Michael Tryfan Matthews, Cong Lu, Benjamin Ellis, Shimon Whiteson, and Jakob Foerster. Policy-guided diffusion. *arXiv preprint arXiv:2404.06356*, 2024.

Yejun Jang, Hong Chul Nam, Jeong Min Park, GIMIN BAE, and Hyun Kwon. Q-guided flow q-learning. In *CoRL 2025 Workshop RemembeRL*, 2025.

Michael Janner, Yilun Du, Joshua B Tenenbaum, and Sergey Levine. Planning with diffusion for flexible behavior synthesis. *arXiv preprint arXiv:2205.09991*, 2022.

Natasha Jaques, Asma Ghandeharioun, Judy Hanwen Shen, Craig Ferguson, Agata Lapedriza, Noah Jones, Shixiang Gu, and Rosalind Picard. Way off-policy batch deep reinforcement learning of implicit human preferences in dialog. *arXiv preprint arXiv:1907.00456*, 2019.

Bingyi Kang, Xiao Jie, Derrick Du, et al. Efficient diffusion policies for offline reinforcement learning. In *Advances in Neural Information Processing Systems*, 2023.

Vijay Konda and John Tsitsiklis. Actor-critic algorithms. *Advances in neural information processing systems*, 12, 1999.

Ilya Kostrikov, Ashvin Nair, and Sergey Levine. Offline reinforcement learning with implicit q-learning. *arXiv preprint arXiv:2110.06169*, 2021.

Aviral Kumar, Justin Fu, Matthew Soh, George Tucker, and Sergey Levine. Stabilizing off-policy q-learning via bootstrapping error reduction. *Advances in neural information processing systems*, 32, 2019.

Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative q-learning for offline reinforcement learning. *Advances in neural information processing systems*, 33:1179–1191, 2020.

Romain Laroche, Paul Trichelair, and Remi Tachet Des Combes. Safe policy improvement with baseline bootstrapping. In *International conference on machine learning*, pp. 3652–3661. PMLR, 2019.

Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

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

Ashvin Nair, Murtaza Dalal, Abhishek Gupta, and Sergey Levine. AWAC: Accelerating online reinforcement learning with offline datasets. In *International Conference on Machine Learning*, pp. 6799–6808, 2020.

Seohong Park, Kevin Frans, Benjamin Eysenbach, and Sergey Levine. Ogbench: Benchmarking offline goal-conditioned rl. *arXiv preprint arXiv:2410.20092*, 2024.

Seohong Park, Qiyang Li, and Sergey Levine. Flow q-learning. *arXiv preprint arXiv:2502.02538*, 2025.

Xue Bin Peng, Aviral Kumar, Grace Zhang, and Sergey Levine. Advantage-weighted regression: Simple and scalable off-policy reinforcement learning. In *International Conference on Learning Representations*, 2020.

M. Riedmiller S. Lange, T. Gabel. Batch reinforcement learning. in: M. wiering, m. van otterlo (eds) reinforcement learning. *Adaptation, Learning, and Optimization*, 12(3):729, 2012.

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

Richard S Sutton, Andrew G Barto, et al. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge, 1998.

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, 2023.

Zhendong Wang, Jonathan J Hunt, and Mingyuan Zhou. Diffusion policies as an expressive policy class for offline reinforcement learning. *arXiv preprint arXiv:2208.06193*, 2022.

Yifan Wu, George Tucker, and Ofir Nachum. Behavior regularized offline reinforcement learning. *arXiv preprint arXiv:1911.11361*, 2019.

Omar G. Younis, Rodrigo Perez-Vicente, John U. Balis, Will Dudley, Alex Davey, and Jordan K Terry. Minari, September 2024. URL `https://doi.org/10.5281/zenodo.13767625`.

Shiyuan Zhang, Weitong Zhang, and Quanquan Gu. Energy-weighted flow matching for offline reinforcement learning. *arXiv preprint arXiv:2503.04975*, 2025.

Qinqing Zheng, Matt Le, Neta Shaul, Yaron Lipman, Aditya Grover, and Ricky TQ Chen. Guided flows for generative modeling and decision making. *arXiv preprint arXiv:2311.13443*, 2023.

A Appendix

A.1 Implementation details

Network architectures. All critic, actor, and flow neural networks use $[512, 512, 512, 512]$ multi-layer perceptrons with GeLU activations. Layer normalization is applied to the critic network.

Flow matching. The number of Euler steps, M, used in Algo. 1, Line 1, is fixed to 10 for both GFP and FQL, except on the humanoidmaze-large-navigate environment, where we set M to 30. For GFP, we employ a sinusoidal position embedding of the flow step t, with an embedding size of 64.

Doubled Q-learning. Following standard practice, two separate critic networks are trained and then aggregated to compute action values, either by taking the mean or minimum Fujimoto et al. (2018). As detailed in Sec. 4.4, we find that the aggregation function has a significant impact on performance for specific tasks. Specifically, by reevaluating ReBRAC on OGBench using the same aggregation function as GFP and FQL, we achieved substantial performance improvements.

Minibatch size. We use a minibatch size of $B = 256$ across most experiments, except on the most challenging tasks, where we evaluate each method with both $B = 256$ and $B = 1024$. The humanoidmaze-large-navigate environment is the only task where methods benefit from different batch sizes: GFP performs best with $B = 1024$, while other methods work better with $B = 256$. Note that on this task, using $\gamma = 0.999$ substantially improves the performance of ReBRAC and FQL compared to previously reported results. For Minari Gym-Mujoco, we use $B = 1024$ following the recommendation in Tarasov et al. (2023) for D4RL Gym-Mujoco.

Training and evaluation. To ensure a fair comparison with FQL, we use identical training durations: 1 M gradient steps on OGBench and 500 K steps on D4RL. For Minari, we adopt 1 M steps following standard practices. Evaluation differs according to the benchmark: D4RL and Minari scores are computed at the end of training, while OGBench scores are averaged over the final three checkpoints (800K, 900K, and 1M steps) following their official evaluation protocol. All results are reported across 8 random seeds that were not used during the hyperparameter tuning process. Tab. 4 summarizes the set of parameter values used by the different evaluated methods.

Parameter search methodology. Our hyperparameter search follows a systematic approach for each method and task. First, we conduct a logarithmic sweep over the BC coefficient α. For ReBRAC, we sweep over the actor coefficient α_1 while keeping the critic coefficient α_2 fixed at 0.01. Then, we sweep over α_2 after selecting the optimal α_1. For GFP, we fix $\eta = 10^{-3}$ and sweep over η after determining the optimal α. Hyperparameter search uses four training seeds, separate from the eight seeds used for final evaluation. On OGBench, hyperparameters are shared across the five tasks within each environment.

Table 4: **Summary of shared hyperparameters used across all methods and benchmark evaluations.** Environment-specific variations are indicated where applicable.

Hyperparameter	Value
Learning rate	0.0003
Gradient steps	1,000,000 (OGBench, Minari), 50,0000 (D4RL)
Minibatch size	256 (default), 1024 (Minari Gym-Mujoco), OGBench Tab. 5
Discount factor	0.99 (D4RL, Minari), OGBench Tab. 5
Euler integration steps	10 (default), 30 on humanoidmaze-large-navigate
Critic aggregation function	mean (default), min (D4RL-antmaze, OGBench-antmaze)
Critic target network smoothing coefficient	0.005

Table 5: **Discount factor and minibatch size for OGBench environments.** The asterisk * in some discount factors indicates cases where we modified the discount factor used in prior work, which led to significant performance improvements for the corresponding methods.

Environment	Discount factor γ	Minibatch size
antmaze-large-navigate-singletask (5 tasks)	0.995*	256
antmaze-large-stitch-singletask (5 tasks)	0.995 (except for FQL, 0.99)	256
antmaze-large-explore-singletask (5 tasks)	0.995	1024
antmaze-giant-navigate-singletask (5 tasks)	0.995	1024
humanoidmaze-medium-navigate-singletask (5 tasks)	0.995	256
humanoidmaze-medium-stitch-singletask (5 tasks)	0.999	256
humanoidmaze-large-navigate-singletask (5 tasks)	0.999*	256 (except for GFP, 1024)
antsoccer-arena-navigate-singletask (5 tasks)	0.99	256
antsoccer-arena-stitch-singletask (5 tasks)	0.99	256
cube-single-play-singletask (5 tasks)	0.99	256
cube-single-noisy-singletask (5 tasks)	0.99	256
cube-double-play-singletask (5 tasks)	0.99	256
cube-double-noisy-singletask (5 tasks)	0.99	256
cube-triple-play-singletask (5 tasks)	0.99	1024
puzzle-4×4-play-singletask (5 tasks)	0.99	256
puzzle-4×4-noisy-singletask (5 tasks)	0.99	256
scene-play-singletask	0.99	256
scene-noisy-singletask	0.99	256

A.2 Additional experiments

Modified Bellman target. As described in Sec. 3, we propose a variant of the Bellman target, y^{VaBC} (Eq. 7), that leverages the VaBC policy. Our experiments demonstrate that this modified target provides improvements in the cube and humanoid maze-medium environments. Tab. 6 presents experimental results showing average scores over 8 seeds for the selected hyperparameters, with "$\sim$" indicating configurations that were tested but not ultimately chosen.

Table 6: **Comparison of the modified Bellman target for GFP.**

Task Category	Standard target	Modified y^{VaBC} Eq. 7
cube-double-play (5 tasks)	~ 28	47.2 ± 1.6
cube-double-noisy (5 tasks)	~ 46	63.1 ± 3.3
humanoidmaze-medium-navigate (5 tasks)	~ 64	72.0 ± 2.8
humanoidmaze-medium-stitch (5 tasks)	~ 59	66.2 ± 5.7
puzzle-4x4-play (5 tasks)	26.1 ± 2.1	~ 22
scene-play (5 tasks)	53.5 ± 2.9	~ 50

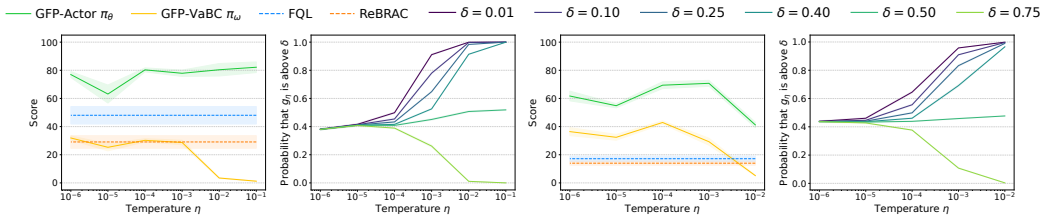
Modified guiding function. Our guiding function g_η, defined in Eq. 10, compares a dataset action a to a^{π_θ}, which is an action sampled from the actor. Since we also sample actions using the VaBC flow policy (Algo. 1, Line 10), we can alternatively use a^{π_ω} for a more conservative guiding function, leading to a modified guided function:

$$g_\eta^{min}(s,a) = \frac{\exp\left(\frac{\lambda}{\eta} Q_\phi(s,a)\right)}{\exp\left(\frac{\lambda}{\eta} Q_\phi(s,a)\right) + \exp\left(\frac{\lambda}{\eta}\ \min\left(Q_\phi(s,a^{\pi_\theta}), Q_\phi(s,a^{\pi_\omega})\right)\right)} \tag{11}$$

This modified guiding function is more conservative because it only filters out action a when both policies produce more valuable alternatives. Initially, we employed $g_\eta^{min}(s,a)$ for evaluation on OGBench antmaze, humanoidmaze, and antsoccer environments. However, we found no significant performance difference, so all remaining tasks use the standard g_η defined in Eq. 10.

Minari MuJoCo reference scores. Since reference scores for the MuJoCo locomotion tasks are not yet available in Minari Younis et al. (2024), we use the reference scores reported in its predecessor D4RL Fu et al. (2020).

Temperature analysis. To complete the analysis presented in Sec. 4.1, we conducted experiments on two additional tasks beyond those shown in Fig. 3. We tested humanoidmaze-medium-stitch as a locomotion task and cube-triple-play as a non-noisy manipulation task. The results of this extended temperature analysis are reported in Fig. 5, illustrating how temperature controls the sharpness of value-guided filtering.



(a) Humanoidmaze score (b) Humanoidmaze $\mathbb{P}(g_\eta > \delta)$ (c) Cube-triple score (d) Cube-triple $\mathbb{P}(g_\eta > \delta)$

Figure 5: **Temperature analysis on two additional very challenging tasks.** Plots (a) and (c), performance scores across temperature values η for our GFP method (Actor π_θ and VaBC π_ω) compared to baselines (FQL, ReBRAC) on humanoidmaze-medium-stitch-task1 and cube-triple-play-task1. Plots (b) and (d), probability that the guidance term g_η is above different thresholds δ as a function of temperature, illustrating how temperature controls the sharpness of value-guided filtering.

Table 7: **Task-specific hyperparameters for offline RL.**

Task	GFP (**ours**) $(\alpha,\ \eta)$
D4RL antmaze-umaze	$(1e{-}1, 1e{-}3)$
D4RL antmaze-umaze-diverse	$(1e{-}1, 1e{-}3)$
D4RL antmaze-medium-play	$(3e{-}2, 1e{-}3)$
D4RL antmaze-medium-diverse	$(3e{-}2, 1e{-}3)$
D4RL antmaze-large-play	$(3e{-}2, 1e{-}5)$
D4RL antmaze-large-diverse	$(3e{-}2, 1e{-}5)$
D4RL pen-human-v1	$(3e{+}0, 1e{-}4)$
D4RL pen-cloned-v1	$(3e{+}0, 1e{-}5)$
D4RL pen-expert-v1	$(1e{+}0, 1e{-}3)$
D4RL door-human-v1	$(1e{+}1, 1e{-}2)$
D4RL door-cloned-v1	$(1e{+}1, 1e{-}2)$
D4RL door-expert-v1	$(1e{+}1, 1e{-}2)$
D4RL hammer-human-v1	$(1e{+}1, 1e{-}5)$
D4RL hammer-cloned-v1	$(1e{+}1, 1e{-}5)$
D4RL hammer-expert-v1	$(1e{+}1, 1e{-}2)$
D4RL relocate-human-v1	$(1e{+}1, 1e{-}4)$
D4RL relocate-cloned-v1	$(1e{+}1, 1e{-}4)$
D4RL relocate-expert-v1	$(1e{+}1, 1e{-}4)$

(a) D4RL

Task	FQL α	GFP (**ours**) $(\alpha,\ \eta)$
Minari pen-human-v2	1e+4	$(3e{+}0, 1e{-}6)$
Minari pen-cloned-v2	3e+3	$(1e{+}0, 1e{-}4)$
Minari pen-expert-v2	1e+3	$(3e{-}1, 1e{-}6)$
Minari door-human-v2	3e+4	$(1e{-}2, 1e{-}2)$
Minari door-cloned-v2	3e+4	$(3e{-}2, 1e{-}6)$
Minari door-expert-v2	3e+4	$(3e{+}0, 1e{-}3)$
Minari hammer-human-v2	3e+4	$(1e{+}0, 1e{-}2)$
Minari hammer-cloned-v2	3e+4	$(3e{+}0, 1e{-}5)$
Minari hammer-expert-v2	1e+4	$(1e{+}0, 1e{-}5)$
Minari relocate-human-v2	3e+3	$(3e{+}0, 1e{-}5)$
Minari relocate-cloned-v2	3e+4	$(3e{+}0, 1e{-}4)$
Minari relocate-expert-v2	3e+4	$(3e{+}0, 1e{-}3)$
Minari halfcheetah-simple-v0	1e+2	$(3e{-}2, 1e{-}6)$
Minari halfcheetah-medium-v0	1e+2	$(1e{-}1, 1e{-}3)$
Minari halfcheetah-expert-v0	1e+3	$(3e{+}0, 1e{-}2)$
Minari hopper-simple-v0	3e+2	$(3e{+}0, 1e{-}3)$
Minari hopper-medium-v0	3e+2	$(3e{+}0, 1e{-}3)$
Minari hopper-expert-v0	1e+3	$(3e{+}0, 1e{-}2)$
Minari walker2d-simple-v0	1e+3	$(3e{+}0, 1e{-}1)$
Minari walker2d-medium-v0	3e+2	$(3e{-}1, 1e{-}2)$
Minari walker2d-expert-v0	3e+2	$(3e{+}0, 1e{-}3)$

(b) Minari

A.3 Complete results over 129 tasks

This section presents the comprehensive experimental results across all 129 tasks from OGBench (Tabs. 8 and 9), D4RL (Tab. 10), and Minari (Tab. 11) benchmarks. We evaluate our proposed GFP method against state-of-the-art baselines including IQL, ReBRAC, and FQL, with results averaged over 8 random seeds per task. The evaluation encompasses approximately 13,000 individual training runs, providing detailed performance comparisons across diverse offline reinforcement learning scenarios including navigation, manipulation, and locomotion tasks. The used hyperparameters are stated in Tabs. 7 and 12.

Table 8: **Offline RL full results on OGBench:** Models were trained with 8 random seeds and evaluated over 100 episodes, following the setup of prior work Park et al. (2024; 2025). Scores are averaged across seeds; values within 95% of the best performance are shown in bold, while *italics* indicate scores reported from prior work Park et al. (2025). GFP actor π_θ is our primary policy, while GFP RaBC π_ω is reported as a byproduct of training. The $\pm$ symbol denotes the standard deviation over seeds.

Task Category	Offline RL algorithms				
	IQL	ReBRAC	FQL	GFP actor π_θ	GFP VaBC π_ω
antmaze-large-navigate-singletask-task1-v0	*48* ± 9	**97.7** ± 0.5	92.6 ± 1.8	**95.4** ± 0.8	92.0 ± 2.3
antmaze-large-navigate-singletask-task2-v0	*42* ± 6	**92.2** ± 1.6	80.0 ± 10.4	**92.2** ± 3.0	**88.5** ± 1.6
antmaze-large-navigate-singletask-task3-v0	*72* ± 7	**98.5** ± 1.8	93.5 ± 1.6	**95.6** ± 2.7	**94.4** ± 3.0
antmaze-large-navigate-singletask-task4-v0	*51* ± 9	**94.4** ± 0.9	82.3 ± 15.5	**90.6** ± 2.6	85.6 ± 3.7
antmaze-large-navigate-singletask-task5-v0	*54* ± 22	**96.5** ± 0.9	**92.3** ± 1.5	**95.0** ± 1.3	89.8 ± 1.7
antmaze-large-stitch-singletask-task1-v0	28.2 ± 7.8	**88.4** ± 16.2	71.8 ± 28.4	**90.3** ± 2.1	85.0 ± 9.7
antmaze-large-stitch-singletask-task2-v0	5.5 ± 4.1	**85.2** ± 5.9	25.4 ± 25.3	**82.9** ± 2.3	69.5 ± 7.2
antmaze-large-stitch-singletask-task3-v0	83.4 ± 2.5	**98.0** ± 0.8	88.4 ± 3.5	**93.2** ± 3.4	90.2 ± 1.5
antmaze-large-stitch-singletask-task4-v0	8.8 ± 2.9	**79.4** ± 29.9	23.7 ± 24.6	0.2 ± 0.6	1.2 ± 3.1
antmaze-large-stitch-singletask-task5-v0	26.2 ± 14.9	**95.1** ± 1.7	81.5 ± 7.2	77.9 ± 7.1	42.2 ± 8.7
antmaze-large-explore-singletask-task1-v0	0.4 ± 1.0	**91.8** ± 6.1	**91.0** ± 10.9	**92.8** ± 4.1	**92.3** ± 1.0
antmaze-large-explore-singletask-task2-v0	0.0 ± 0.0	86.4 ± 5.2	**90.8** ± 3.4	**88.5** ± 3.6	84.1 ± 3.1
antmaze-large-explore-singletask-task3-v0	54.8 ± 7.7	**99.1** ± 0.6	**97.5** ± 0.9	**98.0** ± 0.5	92.8 ± 1.1
antmaze-large-explore-singletask-task4-v0	9.2 ± 4.1	54.4 ± 30.5	**89.0** ± 2.9	**87.1** ± 2.3	87.0 ± 2.8
antmaze-large-explore-singletask-task5-v0	0.0 ± 0.1	81.8 ± 23.7	71.1 ± 32.0	**93.1** ± 2.1	**90.2** ± 9.7
antmaze-giant-navigate-singletask-task1-v0	*0* ± 0	**17.5** ± 3.8	0.8 ± 2.1	12.6 ± 15.1	0.1 ± 0.1
antmaze-giant-navigate-singletask-task2-v0	*1* ± 1	44.9 ± 6.7	23.2 ± 15.3	**52.2** ± 26.5	1.2 ± 0.7
antmaze-giant-navigate-singletask-task3-v0	*0* ± 0	2.5 ± 1.3	0.9 ± 1.1	**13.7** ± 10.2	0.2 ± 0.2
antmaze-giant-navigate-singletask-task4-v0	*0* ± 0	**20.0** ± 20.0	9.9 ± 10.8	17.8 ± 19.9	0.4 ± 0.3
antmaze-giant-navigate-singletask-task5-v0	19 ± 7	**81.4** ± 6.1	46.9 ± 25.5	43.2 ± 35.7	2.0 ± 0.7
humanoidmaze-medium-navigate-singletask-task1-v0	*32* ± 7	34.1 ± 16.4	*19* ± 12	**83.5** ± 3.7	28.8 ± 6.5
humanoidmaze-medium-navigate-singletask-task2-v0	*41* ± 9	75.8 ± 29.	***94*** ± 3	**91.2** ± 6.3	60.2 ± 9.4
humanoidmaze-medium-navigate-singletask-task3-v0	*25* ± 5	69.5 ± 25.7	*74* ± 18	**86.3** ± 10.7	28.9 ± 4.1
humanoidmaze-medium-navigate-singletask-task4-v0	*0* ± 1	**19.5** ± 17.5	*3* ± 4	3.0 ± 6.6	1.3 ± 2.0
humanoidmaze-medium-navigate-singletask-task5-v0	*66* ± 4	**97.0** ± 0.9	***97*** ± 2	**95.8** ± 1.3	60.1 ± 6.1
humanoidmaze-medium-stitch-singletask-task1-v0	26.4 ± 3.0	29.1 ± 18.3	48.0 ± 25.4	**77.9** ± 9.2	28.7 ± 6.5
humanoidmaze-medium-stitch-singletask-task2-v0	27.9 ± 9.9	**94.4** ± 1.9	87.5 ± 3.7	**95.2** ± 1.8	49.2 ± 6.3
humanoidmaze-medium-stitch-singletask-task3-v0	30.0 ± 4.4	56.6 ± 24.5	**85.4** ± 17.8	55.2 ± 33.9	44.3 ± 3.2
humanoidmaze-medium-stitch-singletask-task4-v0	3.7 ± 1.6	**33.1** ± 28.4	0.8 ± 0.8	3.6 ± 9.5	14.9 ± 9.0
humanoidmaze-medium-stitch-singletask-task5-v0	48.5 ± 4.6	92.5 ± 3.1	**94.1** ± 2.5	**98.9** ± 0.5	60.2 ± 7.4
humanoidmaze-large-navigate-singletask-task1-v0	*3* ± 1	27.8 ± 13.4	19.8 ± 13.1	**57.2** ± 23.7	4.5 ± 2.9
humanoidmaze-large-navigate-singletask-task2-v0	*0* ± 0	**0.5** ± 0.7	0.0 ± 0.1	0.1 ± 0.2	0.0 ± 0.0
humanoidmaze-large-navigate-singletask-task3-v0	7 ± 3	**25.9** ± 8.5	8.8 ± 4.0	14.6 ± 16.2	5.5 ± 2.8
humanoidmaze-large-navigate-singletask-task4-v0	*1* ± 0	**8.3** ± 14.4	1.6 ± 1.8	3.7 ± 4.1	0.7 ± 0.5
humanoidmaze-large-navigate-singletask-task5-v0	*1* ± 1	1.9 ± 1.5	2.2 ± 3.9	**13.1** ± 15.5	1.5 ± 0.9
antsoccer-arena-navigate-singletask-task1-v0	*14* ± 5	62.1 ± 3.6	***77*** ± 4	**77.0** ± 1.7	17.0 ± 2.8
antsoccer-arena-navigate-singletask-task2-v0	*17* ± 7	78.5 ± 2.8	***88*** ± 3	**91.2** ± 2.2	16.8 ± 2.6
antsoccer-arena-navigate-singletask-task3-v0	*6* ± 4	55.5 ± 1.7	***61*** ± 6	51.9 ± 4.9	7.8 ± 2.9
antsoccer-arena-navigate-singletask-task4-v0	*3* ± 2	34.8 ± 5.0	***39*** ± 6	**40.2** ± 4.2	5.2 ± 2.1
antsoccer-arena-navigate-singletask-task5-v0	*2* ± 2	**48.5** ± 6.1	*36* ± 9	29.1 ± 8.9	4.6 ± 2.1
antsoccer-arena-stitch-singletask-task1-v0	5.3 ± 3.3	44.6 ± 5.0	**53.4** ± 3.5	**51.8** ± 3.5	3.0 ± 1.2
antsoccer-arena-stitch-singletask-task2-v0	5.6 ± 1.9	30.0 ± 6.0	49.1 ± 8.1	**53.0** ± 7.6	3.0 ± 1.4
antsoccer-arena-stitch-singletask-task3-v0	1.3 ± 1.7	15.9 ± 2.4	**19.3** ± 2.7	**18.7** ± 1.4	0.4 ± 0.3
antsoccer-arena-stitch-singletask-task4-v0	0.4 ± 0.5	14.8 ± 4.2	20.0 ± 6.4	**26.1** ± 4.7	0.1 ± 0.2
antsoccer-arena-stitch-singletask-task5-v0	1.3 ± 1.8	**4.8** ± 1.6	1.2 ± 0.4	2.9 ± 2.9	0.6 ± 0.4
cube-single-play-singletask-task1-v0	*88* ± 3	*89* ± 5	***97*** ± 2	**99.1** ± 0.4	42.1 ± 5.9
cube-single-play-singletask-task2-v0	*85* ± 8	*92* ± 4	***97*** ± 2	**99.4** ± 0.7	38.8 ± 6.4
cube-single-play-singletask-task3-v0	*91* ± 5	*93* ± 3	***98*** ± 2	**99.4** ± 0.5	48.5 ± 9.1
cube-single-play-singletask-task4-v0	*73* ± 6	*92* ± 3	*94* ± 3	**99.1** ± 0.7	32.8 ± 9.3
cube-single-play-singletask-task5-v0	*78* ± 9	*87* ± 8	***93*** ± 3	**97.0** ± 1.6	36.3 ± 5.5
cube-single-noisy-singletask-task1-v0	52.3 ± 7.2	**99.2** ± 1.1	**100.0** ± 0.0	**100.0** ± 0.0	**99.9** ± 0.2
cube-single-noisy-singletask-task2-v0	55.3 ± 8.0	**96.0** ± 3.5	**100.0** ± 0.1	**100.0** ± 0.1	**99.9** ± 0.2
cube-single-noisy-singletask-task3-v0	34.3 ± 8.1	**97.4** ± 1.6	**100.0** ± 0.0	**100.0** ± 0.0	**100.0** ± 0.0
cube-single-noisy-singletask-task4-v0	63.2 ± 7.5	**99.7** ± 0.5	**100.0** ± 0.1	**100.0** ± 0.0	**99.9** ± 0.1
cube-single-noisy-singletask-task5-v0	60.9 ± 11.7	**99.8** ± 0.2	**100.0** ± 0.1	**99.9** ± 0.2	**99.8** ± 0.3
cube-double-play-singletask-task1-v0	*27* ± 5	43.0 ± 8.9	*61* ± 9	**76.1** ± 4.6	28.5 ± 5.0
cube-double-play-singletask-task2-v0	*1* ± 1	16.2 ± 5.0	*36* ± 6	**53.3** ± 8.9	1.8 ± 1.0
cube-double-play-singletask-task3-v0	*0* ± 0	1.3 ± 0.4	*22* ± 5	**43.3** ± 8.9	0.4 ± 0.4
cube-double-play-singletask-task4-v0	*0* ± 0	0.4 ± 0.3	*5* ± 2	**7.1** ± 3.1	0.8 ± 0.6
cube-double-play-singletask-task5-v0	*4* ± 3	2.0 ± 1.0	*19* ± 10	**56.3** ± 11.3	0.7 ± 0.6
cube-double-noisy-singletask-task1-v0	20.8 ± 3.4	51.3 ± 9.5	77.1 ± 8.0	**89.5** ± 4.5	32.2 ± 3.9
cube-double-noisy-singletask-task2-v0	0.0 ± 0.1	21.1 ± 4.3	43.1 ± 10.5	75.7 ± 7.6	5.8 ± 1.3
cube-double-noisy-singletask-task3-v0	0.8 ± 1.0	8.0 ± 3.4	26.3 ± 5.8	**75.0** ± 4.4	3.2 ± 1.2
cube-double-noisy-singletask-task4-v0	0.2 ± 0.2	6.5 ± 1.8	15.5 ± 3.9	**41.8** ± 4.6	1.6 ± 0.9
cube-double-noisy-singletask-task5-v0	0.5 ± 0.5	11.2 ± 3.3	29.0 ± 7.9	**33.4** ± 7.6	3.9 ± 1.5

Table 9: **Offline RL full results on OGBench.** Models were trained with 8 random seeds and evaluated over 100 episodes, following the setup of prior work Park et al. (2024; 2025). Scores are averaged across seeds; values within 95% of the best performance are shown in bold, while *italics* indicate scores reported from prior work Park et al. (2025). GFP actor π_θ is our primary policy, while GFP RaBC π_ω is reported as a byproduct of training. The $\pm$ symbol denotes the standard deviation over seeds.

Task Category	Offline RL algorithms				
	IQL	ReBRAC	FQL	GFP actor π_ω	GFP VaBC π_θ
cube-triple-play-singletask-task1-v0	0.4 ± 0.3	14.0 ± 5.8	17.2 ± 7.3	**54.8** ± 6.2	32.4 ± 8.4
cube-triple-play-singletask-task2-v0	0.0 ± 0.0	0.1 ± 0.1	0.8 ± 0.2	**6.6** ± 6.3	0.6 ± 0.7
cube-triple-play-singletask-task3-v0	1.3 ± 0.6	0.3 ± 0.3	1.3 ± 0.6	**14.9** ± 9.9	3.2 ± 2.1
cube-triple-play-ingletask-task4-v0	0.0 ± 0.0	0.0 ± 0.0	0.3 ± 0.4	**2.5** ± 1.7	0.8 ± 0.2
cube-triple-play-singletask-task5-v0	0.1 ± 0.2	0.3 ± 0.5	0.1 ± 0.2	**0.6** ± 0.5	**1.0** ± 0.9
scene-play-singletask-task1-v0	*94* ± 3	***95*** ± 2	***100*** ± 0	**99.8** ± 0.4	**99.8** ± 0.2
scene-play-singletask-task2-v0	*12* ± 3	*50* ± 13	*76* ± 9	**89.0** ± 4.1	**93.0** ± 5.1
scene-play-singletask-task3-v0	*32* ± 7	*55* ± 16	***98*** ± 1	78.0 ± 13.2	**93.5** ± 5.1
scene-play-singletask-task4-v0	*0* ± 0	*3* ± 3	***5*** ± 1	0.6 ± 0.6	1.8 ± 1.3
scene-play-singletask-task5-v0	*0* ± 0	*0* ± 0	*0* ± 0	0.0 ± 0.0	0.0 ± 0.0
scene-noisy-singletask-task1-v0	74.2 ± 5.4	94.8 ± 3.7	**100.0** ± 0.0	**99.9** ± 0.2	**99.9** ± 0.2
scene-noisy-singletask-task2-v0	0.1 ± 0.2	18.1 ± 5.9	87.4 ± 3.7	**94.2** ± 2.0	**97.4** ± 1.9
scene-noisy-singletask-task3-v0	5.7 ± 1.1	81.1 ± 5.5	**94.4** ± 3.7	**93.3** ± 4.3	**95.2** ± 3.2
scene-noisy-singletask-task4-v0	0.0 ± 0.1	5.6 ± 3.4	**14.8** ± 4.6	0.0 ± 0.0	0.1 ± 0.2
scene-noisy-singletask-task5-v0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0
puzzle-4×4-play-singletask-task1-v0	*12* ± 2	45.4 ± 3.7	*34* ± 8	**50.0** ± 8.1	16.2 ± 3.3
puzzle-4×4-play-singletask-task2-v0	*7* ± 4	2.7 ± 1.0	***16*** ± 5	9.9 ± 2.5	7.0 ± 1.8
puzzle-4×4-play-singletask-task3-v0	*9* ± 3	27.8 ± 4.0	*18* ± 5	**46.2** ± 3.8	10.2 ± 2.9
puzzle-4×4-play-singletask-task4-v0	*5* ± 2	9.1 ± 2.1	*11* ± 3	**17.2** ± 2.5	7.4 ± 1.9
puzzle-4×4-play-singletask-task5-v0	*4* ± 1	0.8 ± 0.7	**7** ± 3	**7.3** ± 3.6	**6.6** ± 1.8
puzzle-4×4-noisy-singletask-task1-v0	0.1 ± 0.1	3.9 ± 1.2	**41.0** ± 3.8	38.5 ± 3.6	**39.6** ± 4.8
puzzle-4×4-noisy-singletask-task2-v0	0.0 ± 0.1	0.4 ± 0.4	**5.9** ± 1.7	0.7 ± 0.5	3.5 ± 1.1
puzzle-4×4-noisy-singletask-task3-v0	0.1 ± 0.2	0.9 ± 0.5	20.8 ± 2.7	**51.1** ± 6.5	44.0 ± 4.7
puzzle-4×4-noisy-singletask-task4-v0	0.0 ± 0.0	0.4 ± 0.4	**6.5** ± 1.6	3.0 ± 1.6	**6.3** ± 2.2
puzzle-4×4-noisy-singletask-task5-v0	0.0 ± 0.0	0.0 ± 0.1	**3.7** ± 1.7	0.7 ± 0.7	**3.1** ± 2.0

Table 10: **Offline RL full results on D4RL.** For each task, models were trained with 8 random seeds and evaluated at the end of training. Reported values are the average normalized scores over the final 100 evaluation episodes, with $\pm$ denoting the standard deviation across seeds. *Italics* indicate scores from prior work Fu et al. (2020); Fujimoto & Gu (2021); Tarasov et al. (2023); Park et al. (2025), and bold denotes values within 95% of the best performance. GFP actor π_θ is our primary policy, while GFP RaBC π_ω is reported as a byproduct of training.

Task Category	Offline RL algorithms							
	BC	CQL	IQL	TD3 + BC	ReBRAC	FQL	GFP actor π_ω	GFP VaBC π_θ
D4RL antmaze-umaze	*55*	*74.0*	*87.5*	*78.6*	***97.8*** ± 1.0	***96*** ± 2	**96.8** ± 1.9	**94.9** ± 2.0
D4RL antmaze-umaze-diverse	*47*	*84.0*	*62.2*	*71.4*	***88.3*** ± 13.0	***89*** ± 5	**91.9** ± 2.7	90.1 ± 3.8
D4RL antmaze-medium-play	*0*	*61.2*	*71.2*	*3.0*	***84.0*** ± 4.2	*78* ± 7	**81.9** ± 5.2	57.4 ± 9.1
D4RL antmaze-medium-diverse	*1*	*53.7*	*70.0*	*10.6*	***76.3*** ± 13.5	*71* ± 13	61.6 ± 20.9	45.6 ± 9.5
D4RL antmaze-large-play	*0*	*15.8*	*39.6*	*0.0*	*60.4* ± 26.1	***84*** ± 7	**82.6** ± 5.4	62.6 ± 8.8
D4RL antmaze-large-diverse	*0*	*14.9*	*47.5*	*0.2*	*54.4* ± 25.1	***83*** ± 4	**84.1** ± 5.4	70.6 ± 4.7
D4RL pen-human-v1	*71*	*37.5*	*71.5*	*81.8*	***103.5***	*53*	64.6 ± 5.4	67.4 ± 6.9
D4RL pen-cloned-v1	*52*	*39.2*	*37.3*	*61.4*	***91.8***	*74*	77.1 ± 10.4	70.5 ± 4.2
D4RL pen-expert-v1	*110*	*107.0*	*133.6*	*146*	***154.1***	*142*	140.4 ± 4.7	123.2 ± 5.4
D4RL door-human-v1	*2*	***9.9***	*4.3*	*−0.1*	*0.0*	*0.0*	0.3 ± 0.3	4.1 ± 2.4
D4RL door-cloned-v1	*−0*	*0.4*	***1.6***	*0.1*	*1.1*	*2*	**1.6** ± 1.9	0.6 ± 0.6
D4RL door-expert-v1	***105***	***101.5***	***105.3***	*84.6*	***104.6***	***104***	**104.1** ± 0.6	**103.1** ± 0.9
D4RL hammer-human-v1	*3*	***4.4***	*1.4*	*0.4*	*0.2*	*1*	**4.4** ± 4.9	2.5 ± 1.1
D4RL hammer-cloned-v1	*1*	*2.1*	*2.1*	*0.8*	*6.7*	***11***	**12.4** ± 5.4	2.5 ± 0.9
D4RL hammer-expert-v1	*127*	*86.7*	***129.6***	*117.0*	***133.8***	*125*	123.6 ± 2.0	116.6 ± 4.1
D4RL relocate-human-v1	*0*	*0.20*	*0.1*	*−0.2*	*0.0*	*0*	**0.5** ± 0.3	0.0 ± 0.0
D4RL relocate-cloned-v1	*−0*	*−0.1*	*−0.2*	*−0.1*	***1.9***	*−0*	1.6 ± 0.7	0.1 ± 0.1
D4RL relocate-expert-v1	***108***	*95.0*	***106.5***	***107.3***	***106.6***	***107***	**103.2** ± 3.7	**104.0** ± 3.1

Table 11: **Offline RL full results on Minari.** For each task, models were trained using 8 different random seeds, and evaluation was performed at the end of training. The reported values represent the average normalized score, computed over the final 100 evaluation episodes and averaged across the 8 seeds. GFP actor π_θ represents our primary policy, while GFP VaBC π_ω is reported as a byproduct of our training procedure.

Task Category	Offline RL algorithms		
	FQL	GFP actor π_θ	GFP VaBC π_ω
Minari pen-human-v2	11.5 ± 4.9	50.1 ± 6.3	**54.4** ± 6.7
Minari pen-cloned-v2	41.8 ± 3.7	**60.4** ± 4.2	54.0 ± 5.5
Minari pen-expert-v2	92.7 ± 6.2	**115.9** ± 4.5	108.7 ± 2.8
Minari door-human-v2	1.1 ± 0.5	0.4 ± 0.1	**2.7** ± 1.9
Minari door-cloned-v2	**0.4** ± 0.2	0.3 ± 0.3	0.1 ± 0.1
Minari door-expert-v2	**102.0** ± 0.9	94.1 ± 20.9	99.0 ± 10.3
Minari hammer-human-v2	1.0 ± 0.6	2.7 ± 0.8	**3.1** ± 0.8
Minari hammer-cloned-v2	1.0 ± 0.6	**22.9** ± 19.5	5.9 ± 4.1
Minari hammer-expert-v2	121.2 ± 4.2	**130.2** ± 5.4	119.0 ± 6.7
Minari relocate-human-v2	−0.0 ± 0.0	**0.1** ± 0.2	−0.0 ± 0.1
Minari relocate-cloned-v2	0.0 ± 0.0	**0.3** ± 0.2	0.0 ± 0.0
Minari relocate-expert-v2	**103.7** ± 1.2	**102.1** ± 5.3	**105.9** ± 1.2
Minari halfcheetah-simple-v0	59.2 ± 0.3	**72.5** ± 0.5	64.4 ± 0.4
Minari halfcheetah-medium-v0	100.2 ± 6.7	**121.3** ± 5.8	108.6 ± 5.3
Minari halfcheetah-expert-v0	**134.1** ± 2.3	**133.4** ± 1.3	**136.4** ± 0.8
Minari hopper-simple-v0	57.2 ± 5.9	**91.6** ± 4.3	**87.4** ± 6.6
Minari hopper-medium-v0	**81.9** ± 23.9	**79.6** ± 14.5	**78.2** ± 24.2
Minari hopper-expert-v0	99.6 ± 10.9	**103.9** ± 10.6	**108.8** ± 13.1
Minari walker2d-simple-v0	**89.4** ± 0.7	**90.0** ± 0.9	**89.7** ± 0.9
Minari walker2d-medium-v0	**127.6** ± 3.0	**133.7** ± 1.1	126.1 ± 3.5
Minari walker2d-expert-v0	**148.0** ± 1.8	**149.9** ± 2.0	**150.8** ± 0.4

Table 12: **Task-specific hyperparameters for offline RL on OGBench.**

Task Category	Offline RL algorithms			
	IQL α	ReBRAC $(\alpha_1,\ \alpha_2)$	FQL α	GFP (**ours**) $(\alpha,\ \eta)$
antmaze-large-navigate-singletask-task{1,2,3,4,5}-v0	1e+1	(1e−2, 1e−2)	1e+1	(3e−1, 1e−4)
antmaze-large-stitch-singletask-task{1,2,3,4,5}-v0	1e+1	(1e−2, 1e−2)	3e+0	(3e−2, 1e−6)
antmaze-large-explore-singletask-task{1,2,3,4,5}-v0	1e+0	(1e−3, 1e−1)	1e+0	(1e−2, 1e−6)
antmaze-large-giant-singletask-task{1,2,3,4,5}-v0	–	(1e−2, 1e−2)	3e+1	(1e−1, 1e−1)
humanoidmaze-medium-navigate-singletask-task{1,2,3,4,5}-v0	–	(1e−2, 1e−2)	3e+1	(3e−1, 1e−3)
humanoidmaze-medium-stitch-singletask-task{1,2,3,4,5}-v0	1e+1	(1e−2, 1e−2)	1e+2	(3e−1, 1e−3)
humanoidmaze-large-navigate-singletask-task{1,2,3,4,5}-v0	–	(1e−2, 0e+0)	1e+2	(3e−1, 1e−4)
antsoccer-arena-navigate-singletask-task{1,2,3,4,5}-v0	–	(1e−2, 0e+0)	–	(1e−1, 1e−2)
antsoccer-arena-stitch-singletask-task{1,2,3,4,5}-v0	1e+0	(1e−2, 1e−3)	1e+1	(1e−1, 1e−2)
cube-single-play-singletask-task{1,2,3,4,5}-v0	1	-	-	(1e+1, 1e−1)
cube-single-noisy-singletask-task{1,2,3,4,5}-v0	3e+0	(1e−1, 1e−1)	3e+1	(1e+1, 1e−3)
cube-double-play-singletask-task{1,2,3,4,5}-v0	–	(1e−1, 3e−1)	–	(1e+0, 1e−2)
cube-double-noisy-singletask-task{1,2,3,4,5}-v0	3e−1	(1e−2, 1e−2)	1e+1	(1e−1, 1e−4)
cube-triple-play-singletask-task{1,2,3,4,5}-v0	1e+0	(1e−1, 1e−3)	3e+2	(1e+0, 1e−5)
scene-play-singletask-task{1,2,3,4,5}-v0	–	(1e−1, 1e−3)	–	(1e+1, 1e−3)
scene-noisy-singletask-task{1,2,3,4,5}-v0	1e+1	(3e−3, 0e+0)	3e+1	(1e+0, 1e−4)
puzzle-4×4-play-singletask-task{1,2,3,4,5}-v0	–	(1e−1, 0e+0)	–	(3e+0, 1e−5)
puzzle-4×4-noisy-singletask-task{1,2,3,4,5}-v0	1e+0	(3e−2, 1e−2)	3e+2	(3e+0, 1e−3)