Behavior Regularization with Flow Latent Policy for Offline Reinforcement Learning

Yulong Xia, Fuchun Sun

Department of Computer Science and Technology, Tsinghua University
xiayl23@mails.tsinghua.edu.cn, fcsun@tsinghua.edu.cn

Abstract

Expressive generative models have recently shown promise in offline reinforcement learning (RL) by capturing the complex, multimodal structure of dataset behavior. However, directly integrating these models into policy optimization introduces substantial computational and stability challenges due to the intricacies of their sampling processes. We introduce Flow Latent Policy (FLP), an offline RL framework that decouples expressivity from optimization by operating entirely in the latent space of a pre-trained, frozen flow-based behavior model. FLP learns a simple latent Gaussian policy whose samples are transformed through the flow to produce complex, behavior-aligned actions. This design enables closed-form behavior regularization via latent-space KL divergence and allows policy optimization without expensive backpropagation through the generative model. Experiments on the OGBench benchmark demonstrate that FLP achieves competitive or superior performance across diverse tasks, combining the benefits of expressive modeling and tractable optimization.

Introduction

Offline reinforcement learning (RL) aims to learn policies from static datasets without requiring online interactions. This paradigm is particularly appealing in real-world scenarios where interaction is costly or unsafe (Levine et al. 2020). A central challenge in offline RL is distribution shift: the learned policy may query actions that are poorly represented in the dataset, which often causes value overestimation and degraded policy performance (Fujimoto, Meger, and Precup 2019; Kumar et al. 2019).

A common way to mitigate this issue is behavior regularization (Wu, Tucker, and Nachum 2019), which constrains the learned policy to remain close to the dataset behavior and thereby reduces extrapolation errors. However, behavior distributions are often complex and multimodal, which conventional unimodal policies struggle to capture accurately, resulting in ineffective regularization (Wang, Hunt, and Zhou 2023; Chen et al. 2023).

To better model such complex distributions, recent studies have incorporated expressive generative models (e.g., diffusion models (Ho, Jain, and Abbeel 2020), flow models (Lipman et al. 2023)) into offline RL. These models provide significantly greater modeling capacity than traditional unimodal policies and have achieved promising empirical results. Nevertheless, integrating them into policy learning poses substantial challenges. Directly optimizing these models requires backpropagation through the multi-step generative process, which is computationally expensive and often unstable (Wang, Hunt, and Zhou 2023; He et al. 2024). Alternative approaches relying on rejection sampling (Hansen-Estruch et al. 2023) or weighted behavior cloning (Lu et al. 2023) avoid backpropagation but suffer from high sampling overhead and numerical issues; these policy extraction methods have also been shown to be less effective than gradient-based policy optimization (Park et al. 2024).

In this work, we take a different perspective: rather than directly optimizing an expressive policy, we freeze a pre-trained flow-based behavior cloning (BC) model and shift policy learning into its latent space (i.e., the base space of the flow). This BC flow establishes a mapping from a simple

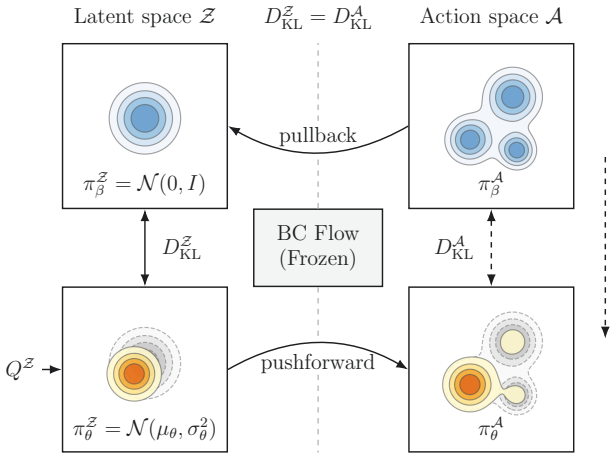

Figure 1: Illustration of FLP. An expressive BC flow is trained to capture the complexity of the action distribution, while a latent policy is optimized in the flow's latent space, enabling tractable optimization and exact KL regularization. The latent policy is tuned toward high-value regions, while the flow maps latent actions to in-distribution actions.

latent distribution (e.g., standard Gaussian) to the rich, multimodal behavior in the dataset. By strategically modifying the latent distribution, we can adjust the resulting action distribution toward higher-value behaviors for policy improvement, while naturally inheriting the expressiveness of the BC flow and remaining close to the behavior distribution for policy regularization. Crucially, in the latent space, the behavior distribution becomes simple and tractable, which is particularly appealing as it eliminates the burden of handling the original complex distribution. Building on this, we instantiate the latent policy as a simple Gaussian, from which latent actions are sampled and mapped through the frozen flow to produce final actions. We refer to our approach as Flow Latent Policy (FLP).

The design of FLP effectively decouples behavior modeling from policy optimization, yielding two main advantages: First, FLP facilitates stable and efficient policy optimization. With the flow being frozen, the need for backpropagation through the iterative generation process is eliminated, leading to improved training stability. By leveraging the inverse map of the flow, actions from the dataset can be mapped back into the latent space, where policy evaluation and optimization can be performed efficiently. Second, FLP enables exact behavior regularization. Since both the behavior prior and the latent policy are Gaussian distributions, we can compute a closed-form KL divergence directly in the latent space. Thanks to the bijectivity of the flow, this latent space KL divergence is equivalent to that in the action space, ensuring accurate alignment with the dataset behavior.

We empirically evaluate FLP on OGBench (Park et al. 2025), a recently proposed offline RL benchmark with a diverse range of tasks. Our experiments demonstrate that FLP achieves competitive or superior performance compared to state-of-the-art baselines. Extensive ablation studies further elucidate the individual contributions of each component.

In summary, our contributions are as follows:

- We introduce Flow Latent Policy (FLP), a new offline RL framework that leverages a frozen flow-based behavior model and optimizes a latent policy in its latent space, thereby combining the expressiveness of generative models with the tractability of decoupled optimization.
- We show that FLP enables closed-form behavior regularization and effective policy optimization, avoiding expensive sampling or backpropagation through the flow.
- Experiments on OGBench show that FLP achieves competitive or superior performance across diverse tasks, validating the benefits of combining expressive behavior models with latent policies.

Preliminaries

Offline RL. We consider the standard formulation of an infinite-horizon discounted Markov Decision Process (MDP), formalized as $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, R, \gamma)$, where $\mathcal{S}$ denotes the state space, $\mathcal{A}$ the action space, $P : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ the transition dynamics, $R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ the reward function, and $\gamma \in [0, 1)$ the discount factor. The goal is to find a policy $\pi_\theta : \mathcal{S} \to \Delta(\mathcal{A})$ that maximizes the expected cumulative discounted return:

$$J(\theta) = \mathbb{E}_{\tau \sim \pi_\theta} \left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t) \right],$$

where $\tau = (s_0, a_0, s_1, a_1, \dots)$ is a trajectory generated by executing π_θ under the environment dynamics. In offline RL, the agent does not interact with the environment. Instead, it learns solely from a fixed dataset $\mathcal{D} = \{(s_i, a_i, r_i, s'_i)\}_{i=1}^n$, generated by an unknown behavior policy π_β. A key challenge arises from the distribution shift between the learned policy π_θ and the behavior policy π_β: since π_θ may query actions not well-represented in $\mathcal{D}$, value estimation can suffer from extrapolation error, often destabilizing policy optimization.

Behavior Regularization in Offline RL. To mitigate distribution shift, a common strategy in offline RL is to regularize the learned policy to remain close to the behavior policy. Many algorithms adopt a behavior-regularized actor-critic framework (Wu, Tucker, and Nachum 2019) with the following objectives:

$$\mathcal{L}_{\text{actor}}(\theta) = \mathbb{E}_{s \sim \mathcal{D}} \Big[- \mathbb{E}_{a \sim \pi_\theta(\cdot|s)} \left[Q_\phi(s, a) \right] + \alpha D \left(\pi_\theta(\cdot|s), \pi_\beta(\cdot|s) \right) \Big], \tag{1}$$

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

where Q_ϕ denotes the action value function parameterized by ϕ, Q_ϕ^- denotes the target network (Mnih et al. 2013), $D(\cdot, \cdot)$ is a divergence measure (e.g., KL divergence), and α controls the regularization strength. This objective trades off between optimality (selecting high-value actions) and conservativeness (staying close to π_β). In practice, exact computation of $D(\cdot, \cdot)$ is often intractable when π_β is implicit (represented only by samples). As a result, prior works approximate this via density estimation or adopt surrogate losses like behavior cloning (Tarasov et al. 2023).

Flow Matching. Flow matching (Lipman et al. 2023; Liu, Gong, and Liu 2023; Albergo, Boffi, and Vanden-Eijnden 2025) is a recent technique for generative modeling. Given a source distribution p and a target distribution q in $\Delta(\mathbb{R}^d)$, the goal is to learn a time-dependent velocity field $v_\omega : \mathbb{R}^d \times [0, 1] \to \mathbb{R}^d$, such that the solution of the ODE:

$$\frac{dx}{dt} = v_\omega(x, t),$$

i.e., the flow $f_\omega : \mathbb{R}^d \times [0, 1] \to \mathbb{R}^d$, transports p at $t = 0$ to match q at $t = 1$.

In this work, we consider the linear path (Lipman et al. 2024) with $p = \mathcal{N}(0, I)$. The corresponding velocity field is

learned by minimizing the conditional flow matching objective (Lipman et al. 2023):

$$\mathcal{L}_{\mathrm{FM}}(\omega) = \mathbb{E}_{\substack{x_0\sim\mathcal{N}(0,I)\\ x_1\sim q\\ t\sim\mathcal{U}[0,1]}}\left[\|v_\omega(x_t,t)-(x_1-x_0)\|_2^2\right], \quad (3)$$

where $x_t = (1-t)x_0 + tx_1$ is the linear interpolation between x_0 and x_1.

In offline RL, such flow models are typically adopted as state-conditional policies, where the target distribution corresponds to the action distribution conditioned on the state.

Method

In this section, we present Flow Latent Policy (FLP), an offline reinforcement learning framework that decouples the modeling of complex behaviors from the optimization of the policy. Our key idea is to leverage an expressive flow model to capture the multimodal structure of dataset actions and then perform policy learning entirely in the flow's latent space. This transforms policy optimization and behavior regularization into tractable problems, while retaining the expressiveness of the flow.

Flow Latent Policy

We start by training a state-conditioned flow model on the offline dataset using Eq. (3). After training, the resulting flow f_ω is used as a mapping from the latent space to the action space, which transforms a standard Gaussian distribution into the action distribution of the dataset. Under standard regularity conditions on the velocity field, the flow f_ω becomes bijective and differentiable.

For ease of optimization, we parameterize the latent policy as a diagonal Gaussian:

$$\pi_\theta^{\mathcal{Z}}(\cdot \mid s) = \mathcal{N}(\mu_\theta(s), \mathrm{diag}(\sigma_\theta^2(s))).$$

The corresponding action space policy $\pi_\theta^{\mathcal{A}}(a \mid s)$ is the pushforward of $\pi_\theta^{\mathcal{Z}}(z \mid s)$ through the frozen BC flow f_ω, given by:

$$\pi_\theta^{\mathcal{A}}(a \mid s) = \pi_\theta^{\mathcal{Z}}(f_\omega^{-1}(a) \mid s) \cdot \left|\det\left(\frac{df_\omega^{-1}}{da}\right)\right|.$$

We now integrate this latent policy into the standard behavior-regularized policy optimization objective in Eq. (1). We use reverse KL as the policy regularization:

$$D\left(\pi_\theta^{\mathcal{A}}(\cdot|s), \pi_\beta(\cdot|s)\right) = D_{\mathrm{KL}}(\pi_\theta^{\mathcal{A}}(\cdot \mid s)\|\pi_\beta(\cdot \mid s)),$$

where π_β is the behavior policy.

Due to the bijectivity and differentiability of the flow f_ω, the change-of-variables formula implies that the KL divergence is invariant under the flow mapping. Formally:

$$D_{\mathrm{KL}}(\pi_\theta^{\mathcal{A}}(\cdot|s)\|\pi_\beta(\cdot|s)) = D_{\mathrm{KL}}(\pi_\theta^{\mathcal{Z}}(\cdot|s)\|\mathcal{N}(0,I)),$$

where the Jacobian terms cancel out. Since both the latent policy and the behavior prior are diagonal Gaussians, the KL divergence admits a closed-form expression:

$$D_{\mathrm{KL}} = \frac{1}{2}\sum_i\Big(\sigma_{\theta,i}^2(s) + \mu_{\theta,i}^2(s) - 1 - \log\sigma_{\theta,i}^2(s)\Big).$$

Algorithm 1: Offline RL with Flow Latent Policy

Input: Offline dataset $\mathcal{D} = \{(s_i, a_i, r_i, s_i')\}_{i=1}^N$
Initialize: Flow model f_ω, latent policy $\pi_\theta^{\mathcal{Z}}(\cdot \mid s)$, latent critic $Q_\phi(s,z)$
Pretrain: Fit f_ω on $\mathcal{D}$ using Eq. (3)
Preprocess: Construct $\mathcal{D}^{\mathcal{Z}} = \{(s_i, z_i, r_i, s_i')\}_{i=1}^N$ by applying f_ω^{-1} to $\mathcal{D}$
for each training iteration **do**
  Sample minibatch $\{(s,z,r,s')\} \sim \mathcal{D}^{\mathcal{Z}}$
  Sample $z \sim \pi_\theta^{\mathcal{Z}}(\cdot \mid s)$
  Update actor $\pi_\theta^{\mathcal{Z}}$ using Eq. (4)
  Sample N latent candidates $z_i' \sim \pi_\theta^{\mathcal{Z}}(\cdot \mid s')$
  Select $z^* = \arg\max_{i=1,\dots N} Q_\phi(s', z_i')$
  Compute TD target $y = r + \gamma Q_\phi(s', z^*)$
  Update critic Q_ϕ using Eq. (5)
end for

This closed-form enables tractable and effective regularization. A short derivation of the equivalence is provided in the Supplementary Materials.

For policy optimization, since actions $a \sim \pi_\theta^{\mathcal{A}}(\cdot \mid s)$ are obtained by sampling latent actions $z \sim \pi_\theta^{\mathcal{Z}}(\cdot \mid s)$ and applying the deterministic flow mapping $a = f_\omega(z)$, the corresponding term becomes:

$$\mathbb{E}_{a\sim\pi_\theta^{\mathcal{A}}(\cdot|s)}[Q(s,a)] = \mathbb{E}_{z\sim\pi_\theta^{\mathcal{Z}}(\cdot|s)}[Q(s, f_\omega(z))].$$

Although evaluating $Q(s, f_\omega(z))$ is feasible, doing so incurs substantial computational overhead during training: each Q-value query would require a forward pass through the multi-step generation process, and policy optimization would involve backpropagating gradients through the frozen flow to update the latent policy. To circumvent this, we introduce a latent critic $Q_\phi(s,z)$ to approximate $Q(s, f_\omega(z))$. This allows efficient training without repeated flow inference or gradient backpropagation through f_ω, while preserving equivalence to the original objective.

Overall, the latent policy is trained with the following loss:

$$\mathcal{L}_{\mathrm{actor}}(\theta) = \mathbb{E}_{s\sim\mathcal{D}}\Big[-\mathbb{E}_{z\sim\pi_\theta^{\mathcal{Z}}(\cdot|s)}Q_\phi(s,z) + \alpha\, D_{\mathrm{KL}}\big(\pi_\theta^{\mathcal{Z}}(\cdot \mid s)\,\|\,\mathcal{N}(0,I)\big)\Big]. \quad (4)$$

Practical Algorithm

We now detail how the above formulation is instantiated into a practical algorithm for offline RL. The training proceeds in two main phases: first, pre-training the behavior cloning flow model f_ω; then, iteratively optimizing the latent policy $\pi_\theta^{\mathcal{Z}}$ and the latent critic Q_ϕ. A full description of the training procedure is provided in Algorithm 1.

To facilitate efficient training, we leverage the invertibility of the flow model to map dataset actions back into the latent space. Specifically, for each transition $(s,a,r,s') \in \mathcal{D}$, we compute the latent action via $z = f_\omega^{-1}(a)$. This yields a transformed dataset, denoted $\mathcal{D}^{\mathcal{Z}} = \{(s,z,r,s')\}$, which is then used to train the latent critic Q_ϕ.

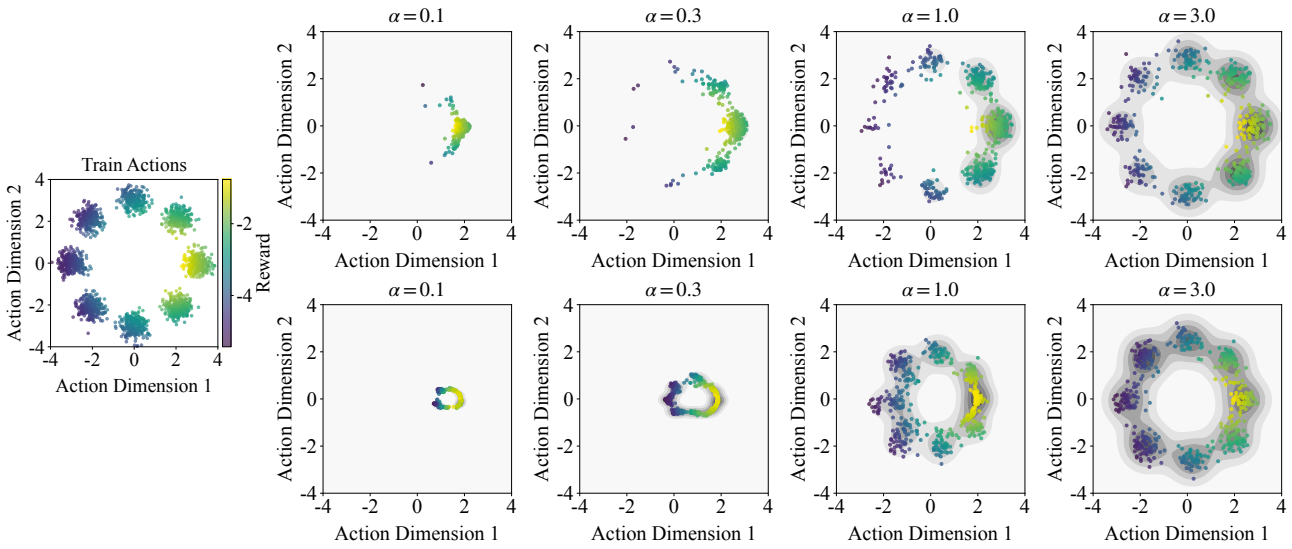

Figure 2: Toy experiment with FLP (top row) and FQL (bottom row) under different regularization strengths. The leftmost panel shows the synthetic behavior dataset.

To enhance action quality, we adopt a best-of-N sampling strategy (Ghasemipour, Schuurmans, and Gu 2021). Given a state s, we draw N latent actions $\{z_i\}_{i=1}^N \sim \pi_\theta^{\mathcal{Z}}(\cdot \mid s)$, and select the one with the highest value:

$$z^* = \arg\max_{i=1,\dots,N} Q_\phi(s, z_i).$$

The best-of-N sampling can be viewed as a global non-parametric conservative policy improvement operator to the current policy. This may help avoid suboptimal behavior induced by local optima and reduce sensitivity to hyper-parameter choices. With effective regularization, the sampled candidates $\{z_i\}_{i=1}^N$ are likely to remain in-distribution. Moreover, the computational overhead is modest because the best-of-N only scores the N latent candidates with the latent critic and decodes the chosen one, without incurring additional iterative action generation steps.

The latent critic is updated to minimize:

$$\mathcal{L}_{\text{critic}}(\phi) = \mathbb{E}_{(s,z,r,s')\sim\mathcal{D}^{\mathcal{Z}}}\Big[(Q_\phi(s,z) - y)^2\Big], \qquad (5)$$

where

$$y = r + \gamma Q_\phi(s', z^*).$$

In summary, FLP provides a principled framework for offline RL by optimizing a simple Gaussian policy in the latent space of an expressive pretrained flow model. This enables exact behavior regularization and effective policy learning, without requiring backpropagation through the generative model or sacrificing expressiveness.

Discussion on Regularization

A key implication of our formulation is that KL regularization can be computed exactly in the latent space while remaining mathematically equivalent to that in the original action space. Beyond ease of implementation, this also allows us to retain the desirable properties of KL-based regularization. In particular, reverse KL divergence imposes a strong penalty on assigning probability mass to low-density regions of the behavior distribution. In the latent space, this reduces to keeping the latent policy close to the standard Gaussian prior; after mapping through the flow, the resulting actions naturally remain in-distribution. Crucially, KL divergence depends solely on probability density and is agnostic of the geometric distance between modes. This enables the policy to reallocate probability mass across distant modes as long as these modes are well supported.

In contrast, geometry-aware regularizers, such as Wasserstein metrics, couple regularization strength to raw distances in the action space. Large regularization weights make it costly for the policy to move between distant modes, effectively trapping it in the original mode. Small weights allow for more flexibility, but often encourage interpolation between modes through low-density regions, which can increase the risk of value overestimation. Consequently, these methods are sensitive to the layout of the action distribution, which makes the suitable regularization strength vary widely across tasks and may require more tuning.

We include a 2D toy experiment to illustrate these effects in Figure 2. KL-based regularization allows the policy to move between modes while remaining in data-supported regions, whereas geometry-aware methods either remain confined to the original mode or traverse low-density regions.

Related Work

Offline RL. A central challenge in offline RL is the distribution shift between the dataset and learned policy. Existing approaches tackling this challenge can be broadly categorized into several directions. Behavior regularization methods constrain the learned policy to stay close to the behav-

Task Category	BC	IQL	ReBRAC	IDQL	SRPO	CAC	FAWAC	FBRAC	IFQL	FQL	FLP
antmaze-large	11 ± 1	53 ± 3	**81** ± 5	21 ± 5	11 ± 4	33 ± 4	6 ± 1	60 ± 6	28 ± 5	**79** ± 3	75 ± 2
antmaze-giant	0 ± 0	4 ± 1	**26** ± 8	0 ± 0	0 ± 0	0 ± 0	0 ± 0	4 ± 4	3 ± 2	9 ± 6	**25** ± 4
humanoid-m	2 ± 1	33 ± 2	22 ± 8	1 ± 0	1 ± 1	53 ± 8	19 ± 1	38 ± 5	60 ± 14	58 ± 5	**72** ± 5
humanoid-l	1 ± 0	2 ± 1	2 ± 1	1 ± 0	0 ± 0	0 ± 0	0 ± 0	2 ± 0	11 ± 2	4 ± 2	**13** ± 3
antsoccer	1 ± 0	8 ± 2	0 ± 0	12 ± 4	1 ± 0	2 ± 4	12 ± 0	16 ± 1	33 ± 6	60 ± 2	**69** ± 2
cube-single	5 ± 1	83 ± 3	91 ± 2	**95** ± 2	80 ± 5	85 ± 9	81 ± 4	79 ± 7	79 ± 2	**96** ± 1	**93** ± 1
cube-double	2 ± 1	7 ± 1	12 ± 1	15 ± 6	2 ± 1	6 ± 2	5 ± 2	15 ± 3	14 ± 3	29 ± 2	**69** ± 2
scene	5 ± 1	28 ± 1	41 ± 3	46 ± 3	20 ± 1	40 ± 7	30 ± 3	45 ± 5	30 ± 3	**56** ± 2	**58** ± 0
puzzle-3x3	2 ± 0	9 ± 1	21 ± 1	10 ± 2	18 ± 1	19 ± 0	6 ± 2	14 ± 4	19 ± 1	**30** ± 1	**32** ± 6
puzzle-4x4	0 ± 0	7 ± 1	14 ± 1	**29** ± 3	10 ± 3	15 ± 3	1 ± 0	13 ± 1	25 ± 5	17 ± 2	20 ± 2

Table 1: Offline RL results on 5 locomotion tasks and 5 manipulation tasks from OGBench. We report mean $\pm$ standard deviation of success rates on 50 environment rollouts over 8 random seeds, averaged across the last three evaluation epochs (800K, 900K, 1M), as suggested in (Park, Li, and Levine 2025). FLP matches or exceeds the best baseline performance on the majority of the benchmarked tasks. Numbers within 5% of the maximum score are denoted in bold. The baseline results are adapted from (Park, Li, and Levine 2025).

Task	ReBRAC	Cal-QL	RLPD	IFQL	FQL	FLP
humanoid-m	$16 \pm 20 \rightarrow 1 \pm 1$	$0 \pm 0 \rightarrow 0 \pm 0$	$0 \pm 0 \rightarrow 8 \pm 10$	$56 \pm 35 \rightarrow 82 \pm 20$	$12 \pm 7 \rightarrow 22 \pm 12$	$78 \pm 3 \rightarrow$ **97** ± 3
antsoccer	$0 \pm 0 \rightarrow 0 \pm 0$	$0 \pm 0 \rightarrow 0 \pm 0$	$0 \pm 0 \rightarrow 0 \pm 0$	$26 \pm 15 \rightarrow 39 \pm 10$	$28 \pm 8 \rightarrow$ **86** ± 5	$52 \pm 8 \rightarrow$ **88** ± 5
cube-double	$6 \pm 5 \rightarrow 28 \pm 28$	$0 \pm 0 \rightarrow 0 \pm 0$	$0 \pm 0 \rightarrow 0 \pm 0$	$12 \pm 9 \rightarrow 40 \pm 5$	$40 \pm 11 \rightarrow 92 \pm 3$	$81 \pm 6 \rightarrow$ **99** ± 1
scene	$55 \pm 10 \rightarrow$ **100** ± 0	$0 \pm 0 \rightarrow 0 \pm 0$	$0 \pm 0 \rightarrow$ **100** ± 0	$0 \pm 1 \rightarrow 60 \pm 39$	$82 \pm 11 \rightarrow$ **100** ± 1	$94 \pm 3 \rightarrow$ **100** ± 1
puzzle-4x4	$8 \pm 4 \rightarrow 14 \pm 35$	$0 \pm 0 \rightarrow 0 \pm 0$	$0 \pm 0 \rightarrow$ **100** ± 1	$23 \pm 6 \rightarrow 19 \pm 33$	$8 \pm 3 \rightarrow 38 \pm 52$	$11 \pm 4 \rightarrow$ **99** ± 2

Table 2: Offline-to-online RL results on selected representative tasks from OGBench. The results are averaged over 8 seeds. Numbers before $\rightarrow$ correspond to the performance at the end of offline RL training (1M steps), and numbers after $\rightarrow$ correspond to the performance at the end of online training (2M steps). Numbers within 5% of the maximum score are denoted in bold. The baseline results are adapted from (Park, Li, and Levine 2025).

ior policy using divergence penalties or imitation losses, as in BEAR (Kumar et al. 2019), BRAC (Wu, Tucker, and Nachum 2019), TD3+BC (Fujimoto and Gu 2021), and ReBRAC (Tarasov et al. 2023). Conservative value methods such as CQL (Kumar et al. 2020) and EDAC (An et al. 2021) mitigate value overestimation by penalizing out-of-distribution actions. In-sample learning methods, such as IQL (Hansen-Estruch et al. 2023), XQL (Garg et al. 2023) and IVQ (Xu et al. 2023), learn the in-sample optimal value function using only actions in the dataset, completely avoiding querying unseen actions. Model-based methods (e.g., MOPO (Yu et al. 2020), MOReL (Kidambi et al. 2020)) leverage learned dynamics models to generate synthetic rollouts for policy optimization. Another line of work recasts offline RL as sequence modeling (Chen et al. 2021; Yamagata, Khalil, and Santos-Rodriguez 2023).

Expressive Generative Models in Offline RL. Recent work has explored the incorporation of expressive generative models into offline RL. These approaches can be grouped according to whether the expressive model is used to represent the target policy or the behavior policy.

Target policy-based methods. Methods such as Diffusion-QL (Wang, Hunt, and Zhou 2023), DiffCPS (He et al. 2024), Consistency-AC (Ding and Jin 2024) and EQL (Zhang et al. 2024) directly optimize the expressive policy with reparameterized policy gradients. While these approaches are capable of modeling complex action distributions, they often incur a high computational cost and training instability. A recent line of work explores diffusion-based policies and obtains tractable KL regularization by leveraging the structure of the diffusion process (Gao et al. 2025). Other approaches, such as QGPO (Lu et al. 2023), DAC (Fang et al. 2025), and QIPO (Zhang, Zhang, and Gu 2025), sidestep backpropagation through time via guided sampling or weighted regression. However, these approaches often lead to less effective policy extraction compared to direct optimization (Park et al. 2024).

Behavior policy-based methods. Alternatively, expressive generative models have been employed to model the behavior policy, either for rejection sampling with a learned Q function (Chen et al. 2023; Hansen-Estruch et al. 2023) or as a regularizer for a lightweight one-step policy. For example, SRPO (Chen et al. 2024) and DTPO (Chen, Wang, and Zhou 2024) use a frozen diffusion model to derive regularization loss for a Gaussian policy, while FQL (Park, Li, and Levine 2025) distills a full flow model into a one-step flow model. These methods improve optimization tractability but at the cost of reduced expressiveness and risk generating out-of-distribution actions.

Our work also falls into this family of behavior policy-based methods, but instead of relying on sampling-based extraction or only using the generative model as a regularizer, we directly perform policy optimization on top of a frozen expressive behavior model, retaining its expressive-

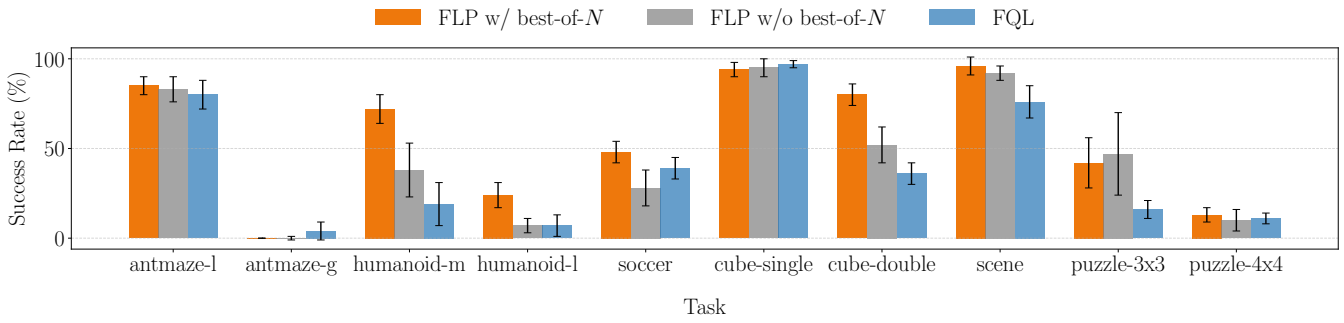

Figure 3: Ablation of FLP with and without best-of-N sampling on the default tasks of each environment.

ness while improving tractability.

Latent Policy in Offline RL. The use of generative models to construct latent policies has been explored in several prior works, primarily to impose implicit behavior regularization. PLAS (Zhou, Bajracharya, and Held 2021) and LAPO (Chen et al. 2022) leverage variational autoencoders (VAEs) to learn a latent action space, perform policy learning in the latent space, and decode latent actions back to the original space. Akimov et al. (2023) utilize Normalizing Flows to construct a tanh-squashed Gaussian latent space and extract policies with weighted regression. These methods mainly rely on heuristic mechanisms to avoid generating out-of-distribution actions. In contrast, our work presents a more principled formulation by using the closed-form KL regularization.

Experiments

Experimental Setup

We evaluate FLP on a diverse suite of offline reinforcement learning tasks from OGBench (Park et al. 2025). Our experiments aim to address the following questions:

(1) How does FLP perform compared to baselines on offline and offline-to-online RL tasks?

(2) How do different components of FLP contribute to its performance?

(3) How sensitive is FLP to main hyperparameters?

Benchmark. OGBench (Park et al. 2025) is a recently introduced benchmark that comprises a diverse set of robotic control tasks. In our experiments, we focus on ten state-based environments, including five locomotion and five manipulation domains. Each environment provides a dataset of one million transitions and defines five task variants with distinct success criteria; rewards are relabeled accordingly, resulting in a total of 50 tasks.

Baselines. For offline RL, we compare with several representative baseline methods including three Gaussian-based offline RL algorithms (BC, IQL (Kostrikov, Nair, and Levine 2021), ReBRAC (Tarasov et al. 2023)), three diffusion-based algorithms (IDQL (Hansen-Estruch et al. 2023), SRPO (Chen et al. 2024), CAC (Ding and Jin 2024)), and four flow-based algorithms (FAWAC (Park, Li, and Levine 2025), FBRAC (Park, Li, and Levine 2025), IFQL (Park, Li, and Levine 2025), FQL (Park, Li, and Levine 2025)). For offline-to-online RL, we consider ReBRAC (Tarasov et al. 2023), Cal-QL (Nakamoto et al. 2023), RLPD (Ball et al. 2023), IFQL and FQL (Park, Li, and Levine 2025). All baseline results are adopted from the FQL paper (Park, Li, and Levine 2025), where the authors performed extensive hyperparameter tuning for these methods.

Evaluation protocol. We follow the evaluation setup in (Park, Li, and Levine 2025). For FLP, we use a fixed number of samples $N = 4$ for best-of-N sampling, and tune the regularization coefficient α on the default task of each environment with a budget comparable to that of the baselines. For offline experiments, we train for 1M steps and report the mean success rate over the last three evaluation epochs (800K, 900K, 1M steps). For offline-to-online experiments, we first train for 1M offline steps, then continue online fine-tuning for another 1M steps, using the same hyperparameters as in the offline phase. We report the performance at 1M (end of offline phase) and 2M steps (end of online phase).

Main Results

Offline RL. Table 1 summarizes the offline RL results on the 50 state-based tasks. FLP achieves competitive performance on most of the tasks, often matching or surpassing the best-performing baselines. The complete results for all 50 task variants are provided in the Supplementary Materials.

Among the baselines, SRPO also decouples behavior modeling from policy optimization and uses KL divergence regularization, but its target policy is a Gaussian policy, which limits the expressiveness. IDQL and IFQL, on the other hand, adopt a similar best-of-N rollout strategy; compared to these methods, FLP achieves strong performance with lower inference overhead.

We found that a single choice of $\alpha = 1$ works well across most tasks, with the only exception being puzzle-4x4, whereas the baselines typically require hyperparameters that vary considerably across tasks. This relative insensitivity to hyperparameters can be partly attributed to the geometry-invariant property of KL divergence discussed in the Method section and also the best-of-N sampling for alleviating sub-optimality. This empirically demonstrates that, while tuning

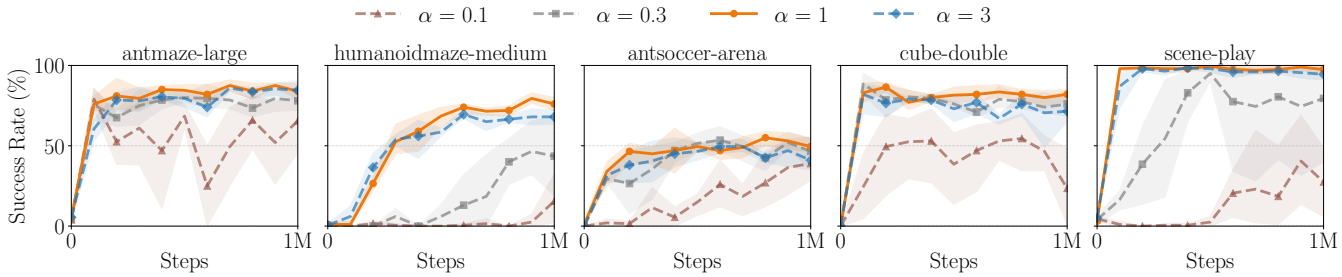

Figure 4: Sensitivity of KL regularization coefficient α.

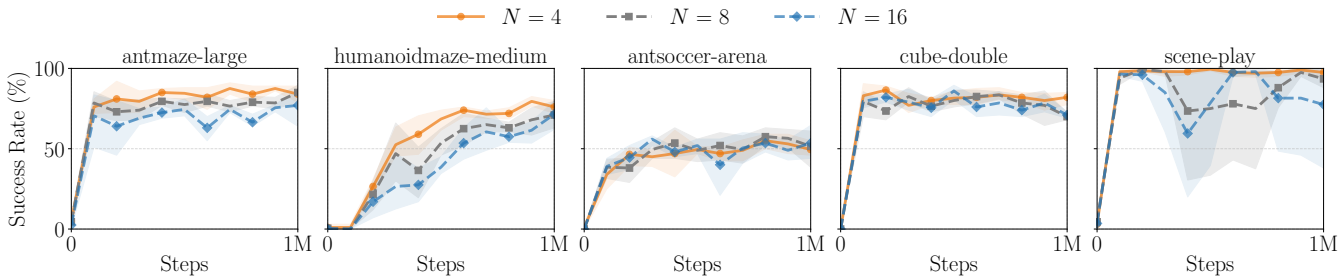

Figure 5: Sensitivity of best-of-N number of samples.

is still needed for the optimal hyperparameters, the range of adjustment is likely to be narrower across tasks.

Offline-to-online RL. For the offline-to-online results in Table 2, although FLP is not specifically designed for online fine-tuning, it often achieves large performance gains during the online phase. Note that the flow model remains frozen throughout this stage, which might appear restrictive for online exploration. We hypothesize that the offline datasets already provide sufficiently diverse and relevant behaviors, which FLP can effectively leverage during fine-tuning.

Ablation Study

We further evaluate our method without best-of-N sampling on the default tasks of each environment and retune the regularization strength accordingly. As shown in Figure 3, drawing multiple samples yields performance gains on several challenging tasks (e.g., humanoid-medium and cube-double) by encouraging the policy to explore a broader range of action candidates and thereby make better use of the expressiveness of the flow.

Sensitivity Analysis

We first investigate the effect of the KL regularization coefficient (α) on performance while keeping the number of samples at $N = 4$. The results of representative tasks are shown in Figure 4. When α is too small (e.g., 0.1 or 0.3), the learned policy tends to deviate from the data distribution and becomes less stable, whereas moderate values lead to more stable performance across tasks.

Next, we analyze the impact of the number of samples N used in best-of-N sampling with α fixed at 1. As shown in Figure 5, $N = 4$ yields the best and most stable performance. Increasing N does not consistently bring additional benefits, as larger values may exacerbate off-distribution tendencies and reduce stability.

Limitations

We identify two key limitations of the proposed framework. First, the ability to achieve exact KL regularization depends on a well-trained flow model; if the behavior model fails to capture the data distribution, this advantage is compromised. Second, while FLP leverages the expressiveness of the BC flow to capture the complex behavior distribution, the latent policy itself remains a unimodal Gaussian, which can be insufficient to represent the target policy when the Q-function is highly multimodal.

Conclusion

In this work, we introduced Flow Latent Policy (FLP), a framework for offline reinforcement learning that decouples policy optimization from behavior modeling by operating in the latent space of a frozen flow-based behavior model. This formulation enables tractable optimization and exact KL regularization while inheriting the expressiveness of the flow. Our approach avoids the computational cost of back-propagating through complex generative models and instead leverages a latent critic for efficient policy improvement. Experiments on OGBench show that FLP achieves competitive or superior performance across a wide range of offline and offline-to-online tasks. These results suggest that latent-space optimization, combined with expressive behavior models, is a viable direction for advancing offline RL.

Acknowledgments

This work is supported by the National Key Research and Development Program of China (No. 2024YFB4711102) and the Joint Funds of the National Natural Science Foundation of China (No. U22A2057 and No. U22B2042).

References

Akimov, D.; Kurenkov, V.; Nikulin, A.; Tarasov, D.; and Kolesnikov, S. 2023. Let Offline RL Flow: Training Conservative Agents in the Latent Space of Normalizing Flows. arXiv:2211.11096.

Albergo, M. S.; Boffi, N. M.; and Vanden-Eijnden, E. 2025. Stochastic Interpolants: A Unifying Framework for Flows and Diffusions. arXiv:2303.08797.

An, G.; Moon, S.; Kim, J.-H.; and Song, H. O. 2021. Uncertainty-Based Offline Reinforcement Learning with Diversified Q-Ensemble. In Ranzato, M.; Beygelzimer, A.; Dauphin, Y.; Liang, P.; and Vaughan, J. W., eds., *Advances in Neural Information Processing Systems*, volume 34, 7436–7447. Curran Associates, Inc.

Ball, P. J.; Smith, L.; Kostrikov, I.; and Levine, S. 2023. Efficient Online Reinforcement Learning with Offline Data. In Krause, A.; Brunskill, E.; Cho, K.; Engelhardt, B.; Sabato, S.; and Scarlett, J., eds., *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, 1577–1594. PMLR.

Chen, H.; Lu, C.; Wang, Z.; Su, H.; and Zhu, J. 2024. Score Regularized Policy Optimization through Diffusion Behavior. In *The Twelfth International Conference on Learning Representations*.

Chen, H.; Lu, C.; Ying, C.; Su, H.; and Zhu, J. 2023. Offline Reinforcement Learning via High-Fidelity Generative Behavior Modeling. In *The Eleventh International Conference on Learning Representations*.

Chen, L.; Lu, K.; Rajeswaran, A.; Lee, K.; Grover, A.; Laskin, M.; Abbeel, P.; Srinivas, A.; and Mordatch, I. 2021. Decision Transformer: Reinforcement Learning via Sequence Modeling. In Ranzato, M.; Beygelzimer, A.; Dauphin, Y.; Liang, P.; and Vaughan, J. W., eds., *Advances in Neural Information Processing Systems*, volume 34, 15084–15097. Curran Associates, Inc.

Chen, T.; Wang, Z.; and Zhou, M. 2024. Diffusion Policies Creating a Trust Region for Offline Reinforcement Learning. In Globerson, A.; Mackey, L.; Belgrave, D.; Fan, A.; Paquet, U.; Tomczak, J.; and Zhang, C., eds., *Advances in Neural Information Processing Systems*, volume 37, 50098–50125. Curran Associates, Inc.

Chen, X.; Ghadirzadeh, A.; Yu, T.; Wang, J.; Gao, A. Y.; Li, W.; Bin, L.; Finn, C.; and Zhang, C. 2022. LAPO: Latent-Variable Advantage-Weighted Policy Optimization for Offline Reinforcement Learning. In Koyejo, S.; Mohamed, S.; Agarwal, A.; Belgrave, D.; Cho, K.; and Oh, A., eds., *Advances in Neural Information Processing Systems*, volume 35, 36902–36913. Curran Associates, Inc.

Ding, Z.; and Jin, C. 2024. Consistency Models as a Rich and Efficient Policy Class for Reinforcement Learning. In *The Twelfth International Conference on Learning Representations*.

Fang, L.; Liu, R.; Zhang, J.; Wang, W.; and Jing, B. 2025. Diffusion Actor-Critic: Formulating Constrained Policy Iteration as Diffusion Noise Regression for Offline Reinforcement Learning. In *The Thirteenth International Conference on Learning Representations*.

Fujimoto, S.; and Gu, S. S. 2021. A Minimalist Approach to Offline Reinforcement Learning. In Ranzato, M.; Beygelzimer, A.; Dauphin, Y.; Liang, P.; and Vaughan, J. W., eds., *Advances in Neural Information Processing Systems*, volume 34, 20132–20145. Curran Associates, Inc.

Fujimoto, S.; Meger, D.; and Precup, D. 2019. Off-Policy Deep Reinforcement Learning without Exploration. In Chaudhuri, K.; and Salakhutdinov, R., eds., *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, 2052–2062. PMLR.

Gao, C.-X.; Wu, C.; Cao, M.; Xiao, C.; Yu, Y.; and Zhang, Z. 2025. Behavior-Regularized Diffusion Policy Optimization for Offline Reinforcement Learning. In Singh, A.; Fazel, M.; Hsu, D.; Lacoste-Julien, S.; Berkenkamp, F.; Maharaj, T.; Wagstaff, K.; and Zhu, J., eds., *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, 18630–18657. PMLR.

Garg, D.; Hejna, J.; Geist, M.; and Ermon, S. 2023. Extreme Q-Learning: MaxEnt RL without Entropy. In *The Eleventh International Conference on Learning Representations*.

Ghasemipour, S. K. S.; Schuurmans, D.; and Gu, S. S. 2021. EMaQ: Expected-Max Q-Learning Operator for Simple Yet Effective Offline and Online RL. In Meila, M.; and Zhang, T., eds., *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, 3682–3691. PMLR.

Hansen-Estruch, P.; Kostrikov, I.; Janner, M.; Kuba, J. G.; and Levine, S. 2023. IDQL: Implicit Q-Learning as an Actor-Critic Method with Diffusion Policies. arXiv:2304.10573.

He, L.; Shen, L.; Zhang, L.; Tan, J.; and Wang, X. 2024. DiffCPS: Diffusion Model based Constrained Policy Search for Offline Reinforcement Learning. arXiv:2310.05333.

Ho, J.; Jain, A.; and Abbeel, P. 2020. Denoising Diffusion Probabilistic Models. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M.; and Lin, H., eds., *Advances in Neural Information Processing Systems*, volume 33, 6840–6851. Curran Associates, Inc.

Kidambi, R.; Rajeswaran, A.; Netrapalli, P.; and Joachims, T. 2020. MOReL: Model-Based Offline Reinforcement Learning. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M.; and Lin, H., eds., *Advances in Neural Information Processing Systems*, volume 33, 21810–21823. Curran Associates, Inc.

Kostrikov, I.; Nair, A.; and Levine, S. 2021. Offline Reinforcement Learning with Implicit Q-Learning. arXiv:2110.06169.

Kumar, A.; Fu, J.; Soh, M.; Tucker, G.; and Levine, S. 2019. Stabilizing Off-Policy Q-Learning via Bootstrapping Error Reduction. In Wallach, H.; Larochelle, H.; Beygelzimer, A.; d'Alché-Buc, F.; Fox, E.; and Garnett, R., eds., *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Kumar, A.; Zhou, A.; Tucker, G.; and Levine, S. 2020. Conservative Q-Learning for Offline Reinforcement Learning. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M.; and Lin, H., eds., *Advances in Neural Information Processing Systems*, volume 33, 1179–1191. Curran Associates, Inc.

Levine, S.; Kumar, A.; Tucker, G.; and Fu, J. 2020. Offline Reinforcement Learning: Tutorial, Review, and Perspectives on Open Problems. arXiv:2005.01643.

Lipman, Y.; Chen, R. T. Q.; Ben-Hamu, H.; Nickel, M.; and Le, M. 2023. Flow Matching for Generative Modeling. In *The Eleventh International Conference on Learning Representations*.

Lipman, Y.; Havasi, M.; Holderrieth, P.; Shaul, N.; Le, M.; Karrer, B.; Chen, R. T. Q.; Lopez-Paz, D.; Ben-Hamu, H.; and Gat, I. 2024. Flow Matching Guide and Code. arXiv:2412.06264.

Liu, X.; Gong, C.; and Liu, Q. 2023. Flow Straight and Fast: Learning to Generate and Transfer Data with Rectified Flow. In *The Eleventh International Conference on Learning Representations*.

Lu, C.; Chen, H.; Chen, J.; Su, H.; Li, C.; and Zhu, J. 2023. Contrastive Energy Prediction for Exact Energy-Guided Diffusion Sampling in Offline Reinforcement Learning. In Krause, A.; Brunskill, E.; Cho, K.; Engelhardt, B.; Sabato, S.; and Scarlett, J., eds., *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, 22825–22855. PMLR.

Mnih, V.; Kavukcuoglu, K.; Silver, D.; Graves, A.; Antonoglou, I.; Wierstra, D.; and Riedmiller, M. 2013. Playing Atari with Deep Reinforcement Learning. arXiv:1312.5602.

Nakamoto, M.; Zhai, S.; Singh, A.; Sobol Mark, M.; Ma, Y.; Finn, C.; Kumar, A.; and Levine, S. 2023. Cal-QL: Calibrated Offline RL Pre-Training for Efficient Online Fine-Tuning. In Oh, A.; Naumann, T.; Globerson, A.; Saenko, K.; Hardt, M.; and Levine, S., eds., *Advances in Neural Information Processing Systems*, volume 36, 62244–62269. Curran Associates, Inc.

Park, S.; Frans, K.; Eysenbach, B.; and Levine, S. 2025. OGBench: Benchmarking Offline Goal-Conditioned RL. In *The Thirteenth International Conference on Learning Representations*.

Park, S.; Frans, K.; Levine, S.; and Kumar, A. 2024. Is Value Learning Really the Main Bottleneck in Offline RL? In Globerson, A.; Mackey, L.; Belgrave, D.; Fan, A.; Paquet, U.; Tomczak, J.; and Zhang, C., eds., *Advances in Neural Information Processing Systems*, volume 37, 79029–79056. Curran Associates, Inc.

Park, S.; Li, Q.; and Levine, S. 2025. Flow Q-Learning. In Singh, A.; Fazel, M.; Hsu, D.; Lacoste-Julien, S.; Berkenkamp, F.; Maharaj, T.; Wagstaff, K.; and Zhu, J., eds., *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, 48104–48127. PMLR.

Tarasov, D.; Kurenkov, V.; Nikulin, A.; and Kolesnikov, S. 2023. Revisiting the Minimalist Approach to Offline Reinforcement Learning. In Oh, A.; Naumann, T.; Globerson, A.; Saenko, K.; Hardt, M.; and Levine, S., eds., *Advances in Neural Information Processing Systems*, volume 36, 11592–11620. Curran Associates, Inc.

Wang, Z.; Hunt, J. J.; and Zhou, M. 2023. Diffusion Policies as an Expressive Policy Class for Offline Reinforcement Learning. In *The Eleventh International Conference on Learning Representations*.

Wu, Y.; Tucker, G.; and Nachum, O. 2019. Behavior Regularized Offline Reinforcement Learning. arXiv:1911.11361.

Xu, H.; Jiang, L.; Li, J.; Yang, Z.; Wang, Z.; Chan, V. W. K.; and Zhan, X. 2023. Offline RL with No OOD Actions: In-Sample Learning via Implicit Value Regularization. In *The Eleventh International Conference on Learning Representations*.

Yamagata, T.; Khalil, A.; and Santos-Rodriguez, R. 2023. Q-learning Decision Transformer: Leveraging Dynamic Programming for Conditional Sequence Modelling in Offline RL. In Krause, A.; Brunskill, E.; Cho, K.; Engelhardt, B.; Sabato, S.; and Scarlett, J., eds., *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, 38989–39007. PMLR.

Yu, T.; Thomas, G.; Yu, L.; Ermon, S.; Zou, J. Y.; Levine, S.; Finn, C.; and Ma, T. 2020. MOPO: Model-based Offline Policy Optimization. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M.; and Lin, H., eds., *Advances in Neural Information Processing Systems*, volume 33, 14129–14142. Curran Associates, Inc.

Zhang, R.; Luo, Z.; Sjölund, J.; Schön, T. B.; and Mattsson, P. 2024. Entropy-regularized Diffusion Policy with Q-Ensembles for Offline Reinforcement Learning. In Globerson, A.; Mackey, L.; Belgrave, D.; Fan, A.; Paquet, U.; Tomczak, J.; and Zhang, C., eds., *Advances in Neural Information Processing Systems*, volume 37, 98871–98897. Curran Associates, Inc.

Zhang, S.; Zhang, W.; and Gu, Q. 2025. Energy-Weighted Flow Matching for Offline Reinforcement Learning. In Yue, Y.; Garg, A.; Peng, N.; Sha, F.; and Yu, R., eds., *International Conference on Representation Learning*, volume 2025, 17943–17970.

Zhou, W.; Bajracharya, S.; and Held, D. 2021. PLAS: Latent Action Space for Offline Reinforcement Learning. In Kober, J.; Ramos, F.; and Tomlin, C., eds., *Proceedings of the 2020 Conference on Robot Learning*, volume 155 of *Proceedings of Machine Learning Research*, 1719–1735. PMLR.