Offline Inverse Reinforcement Learning with Unified Diffusion Planning

Hongmin Zhao[1,2], **Jiyuan Yin**[1,2], **Kexi Yan**[1,2], **Qinglai Wei**[1,2,3,4,†], **Jie Zhang**[1,2,†]
[1]Institute of Automation, Chinese Academy of Sciences
[2]School of Artificial Intelligence, University of Chinese Academy of Sciences
[3]Institute of Systems Engineering, Macau University of Science and Technology
[4]School of Artificial Intelligence, Shenyang University of Technology
{zhaohongmin2023,yinjiyuan2022,yankexi2022,qinglai.wei,jie.zhang}
@ia.ac.cn
† denotes the corresponding author

Abstract

Offline Inverse Reinforcement Learning (IRL) aims to recover a reward function and imitate expert behavior solely from offline demonstrations. While recent Offline IRL approaches employ approximate dynamics to mitigate distribution shift, their performance is constrained by the unstable minimax framework as well as simplified representation and utilization. To address these issues, we propose Offline Inverse Reinforcement Learning with Unified Diffusion Planner (OIDP), which leverages diffusion models to achieve stable Offline IRL. OIDP formulates a model-based conservative Q-optimization through inverse soft Q-learning, with provably concave Q-update subproblems for a fixed policy and a controllable objective discrepancy. To this end, we introduce a unified diffusion planner that models dynamics, uncertainty estimator, and policy. It captures multimodal distributions and tightens the objective gap bound, and at execution performs trajectory-guided policy sampling using the internalized dynamics knowledge. Experimental results on D4RL benchmarks demonstrate that OIDP outperforms state-of-the-art offline approaches.

1 Introduction

Reinforcement Learning (RL) is a powerful paradigm for solving sequential decision-making problems [1–3]. However, its success critically depends on the availability of carefully designed reward functions [4]. Imitation Learning (IL) circumvents this limitation by learning from expert demonstrations, and is commonly categorized into Behavior Cloning (BC) and Inverse Reinforcement Learning (IRL) [5]. BC approaches treat IL as a supervised learning task but neglect the sequential nature of decision-making, leading to rapid error accumulation under minor state deviations [6]. In contrast, IRL incorporates environmental dynamics by inferring an underlying reward function that explains expert behavior, from which an optimal policy can be derived. Traditional IRL approaches [7, 8] adopt the unstable minimax optimization framework, alternating between the reward space and the policy or Q-space. To address this issue, recent methods have proposed stable formulations that optimize exclusively in the Q-space [9]. However, in safety-critical domains such as clinical treatment [10] and autonomous driving [11], *Offline IRL*, which learns exclusively from static datasets, has become a necessary setting.

Despite recent progress, existing Offline IRL methods still face several fundamental limitations. Some approaches leverage additional non-expert data to learn approximate dynamics model in order to mitigate distribution shift [12, 13]. However, these methods rely on the minimax optimization

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

framework inherited from online IRL, in which errors in the learned model accumulate through repeated optimization across different spaces, ultimately leading to significant deviations from the desired solution [9], as illustrated in Fig. 1A. Moreover, in complex environments, commonly adopted Gaussian models impose overly restrictive assumptions on dynamic representation and lack a natural uncertainty estimator for the conservative penalty, degrading IRL performance [14]. Finally, the approximated model is discarded during execution after solving the IRL problem, which prevents the utilization of dynamics knowledge embedded from offline data, resulting in inefficient data usage. A more detailed discussion of related work is provided in Appendix A.

To address these issues, we propose **O**ffline **I**nverse Reinforcement Learning with Unified **D**iffusion **P**lanner (**OIDP**), which leverages diffusion models [15, 16] to enable a stable, model-based Offline IRL. Specifically, we employ inverse soft Q-learning [9] to formulate an offline objective solely in Q-space, which enables conservative optimization, avoids error accumulation from cross-space iterations, and recovers the corresponding reward function and policy, as shown in Fig. 1B. To support this formulation, we introduce a unified diffusion planner in which the score architecture simultaneously represents the dynamics model, the uncertainty estimator, and the energy-based policy. On the one hand, the expressive generative capacity of diffusion models captures complex and multi-modal distributions from offline data, while the score-matching residual provides a natural uncertainty estimator. On the other hand, the planner directly leverages internalized dynamics knowledge to perform diffusion trajectory planning during execution, guiding the policy to sample high-value actions [17]. To the best of our knowledge, this is the first work that employs diffusion models to develop a stable algorithm for Offline IRL with the associated theoretical analysis.

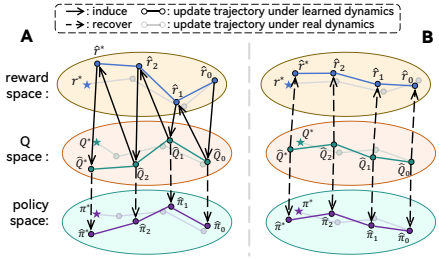


Figure 1: Comparison between minimax optimization and single-space optimization in Offline IRL. **A.** Minimax optimization repeatedly alternates across multiple spaces, potentially exacerbating model error accumulation. **B.** Q-space optimization avoids this error amplification and enables simultaneous recovery of reward and policy.

In summary, our contributions are: (1) We propose a novel and stable Offline IRL formulation that achieves conservative optimization solely within the Q-space based on an approximate dynamics model. We provide theoretical analysis showing that each Q-update subproblem is concave for a fixed policy and bounding the objective discrepancy relative to the online IRL objective. (Sec. 3.1) (2) We develop OIDP, which utilizes unified diffusion models to represent the environment dynamics, the uncertainty estimator, and the energy-based policy. This not only improves representation and provides natural uncertainty estimation for conservative optimization, but also enables the policy to leverage model-based planning to guide action sampling during execution. (Sec. 3.2, 3.3) (3) We conduct extensive evaluations on D4RL offline benchmarks. Experimental results demonstrate that OIDP outperforms state-of-the-art Offline IL algorithms, and the recovered reward function reflects meaningful ground-truth incentives. (Sec. 4)

2 Preliminary

We consider environments modeled as a Markov Decision Process (MDP), defined by the tuple $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, r, P, p_0, \gamma \rangle$. Here, $\mathcal{S}$ and $\mathcal{A}$ denote the state and action spaces, respectively; $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward function; $P : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}^+$ is the transition probability density; $p_0 : \mathcal{S} \to \mathbb{R}^+$ is the initial state distribution; and $\gamma \in [0, 1)$ is the discount factor. A stochastic policy $\pi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^+$ represents the probability distribution over actions given states. Given P and π, the discounted state occupancy measure is defined as $d_\pi(s) = (1-\gamma)\sum_{t=0}^{\infty} \gamma^t \Pr(s_t = s|P, \pi, p_0)$, and the state-action occupancy measure is $\rho_\pi(s, a) = d_\pi(s)\pi(a|s)$. In IRL, the reward function $r(\cdot, \cdot)$ is unknown.

2.1 Maximum entropy IRL

Given expert demonstrations $\mathcal{D}_E = \{(s_i, a_i, s'_i)\}_{i=1}^{|\mathcal{D}_E|}$ sampled from an expert policy π_E with empirical occupancy measure ρ_E, MaxEnt IRL [7] aims to recover the underlying reward function

and imitate the expert by solving the following minimax optimization problem:

$$\max_{r\in\mathcal{R}} \min_{\pi\in\Pi} \mathbb{E}_{\rho_E}[r(s,a)] - \mathbb{E}_{\rho_\pi}[r(s,a)] - \alpha H(\pi) - \psi(r), \tag{1}$$

where $\mathcal{R}$ and Π are the sets of reward functions and policies; $H(\pi) = \mathbb{E}_{\rho_\pi}[-\log \pi(a|s)]$ is the discounted causal entropy; $\alpha > 0$ is the temperature parameter controlling the exploration–exploitation trade-off; and $\psi_\rho(r) = \mathbb{E}_{(s,a)\sim\rho}[\zeta(r(s,a))]$ is an occupancy-weighted, pointwise separable regularizer with convex $\zeta : \mathbb{R} \to \mathbb{R}$, omitting the occupancy subscript for brevity. Despite strong theoretical foundations, the nested minimax structure introduces significant training instability [9], and its inner loop requires extensive interactions to solve a forward RL problem.

2.2 IQ-Learn

For the soft Bellman operator $(\mathcal{B}^\pi Q)(s,a) = r(s,a) + \gamma \mathbb{E}_{s'\sim P(\cdot|s,a)} V^\pi(s')$ with $V^\pi(s) = \mathbb{E}_{a\sim\pi(\cdot|s)}[Q(s,a) - \alpha \log \pi(a|s)]$, IQ-Learn [9] introduces a bijective inverse soft Bellman operator $(\mathcal{T}^\pi Q)(s,a) = Q(s,a) - \gamma \mathbb{E}_{s'\sim P(\cdot|s,a)} V^\pi(s')$. For a fixed policy π and any reward $r \in \mathcal{R}$, $Q = (\mathcal{T}^\pi)^{-1} r$ is the unique fixed point of $\mathcal{B}^\pi$. IQ-Learn thus reformulates the MaxEnt IRL objective Eq. 1 from reward space to Q-space:

$$\max_{Q\in\Omega} \min_{\pi\in\Pi} \mathbb{E}_{\rho_E}[(\mathcal{T}^\pi Q)(s,a)] - \mathbb{E}_{\rho_\pi}[(\mathcal{T}^\pi Q)(s,a)] - \alpha H(\pi) - \psi(\mathcal{T}^\pi Q), \tag{2}$$

which is concave in Q and preserves a unique saddle point under the assumptions of IQ-Learn. For fixed Q, the optimal policy is $\pi = \frac{1}{Z_s} \exp(\frac{1}{\alpha} Q(s,a))$ with $Z_s = \sum_a \exp(\frac{1}{\alpha} Q(s,a))$. Substituting yields an optimization solely in Q-space:

$$\max_{Q\in\Omega} \mathcal{J}(Q) = \mathbb{E}_{\rho_E}[(\mathcal{T}^\pi Q)(s,a)] - \mathbb{E}_{\rho_\pi}[(\mathcal{T}^\pi Q)(s,a)] - \alpha H(\pi) - \psi(\mathcal{T}^\pi Q). \tag{3}$$

By avoiding the nested minimax, IQ-Learn substantially improves training stability. However, in offline settings, the estimation of $\mathbb{E}_{\rho_\pi}[(\mathcal{T}^\pi Q)(s,a)]$ in Eq. 3 still suffers from significant bias—addressing this is a central contribution of this work.

2.3 Diffusion models

Diffusion models (DMs) [15, 16] are generative models that represent arbitrarily complex distributions through a forward noising process and a reverse denoising process. Given the data distribution $p(\boldsymbol{x}_0)$, DMs gradually transform data into Gaussian noise via the SDE:

$$\mathrm{d}\boldsymbol{x}_k = \boldsymbol{\mu}(\boldsymbol{x}_k, k)\mathrm{d}k + \sigma(k)\mathrm{d}\boldsymbol{w}_k, \tag{4}$$

where $k \in [0, K]$, $K > 0$ is the terminal diffusion time;[1] $\boldsymbol{\mu}(\cdot,\cdot)$ is the drift coefficient; $\sigma(\cdot)$ is the diffusion coefficient; and $\{\boldsymbol{w}_k\}_{k\in[0,K]}$ is the standard Brownian motion. With an appropriate noise schedule, the terminal distribution satisfies $p(\boldsymbol{x}_K) \approx \mathcal{N}(\mathbf{0}, \boldsymbol{I})$ [16]. There exists a probability flow ODE that shares the same marginal as Eq. 4 and describes the reverse denoising process:

$$\mathrm{d}\boldsymbol{x}_k = \left[\boldsymbol{\mu}(\boldsymbol{x}_k, k) - \frac{1}{2}\sigma(k)^2 \nabla \log p(\boldsymbol{x}_k)\right] \mathrm{d}k, \tag{5}$$

where $\nabla \log p(\boldsymbol{x}_k)$ is the marginal score function, which is generally intractable. To approximate it, a neural network $\boldsymbol{s}_\phi(\boldsymbol{x}_k, k)$ is trained using denoising score matching (DSM):

$$L(\phi) = \mathbb{E}_{k,\boldsymbol{x}_0,\boldsymbol{x}_k}\left[\lambda(k) \left\|\boldsymbol{s}_\phi(\boldsymbol{x}_k, k) - \nabla \log q_k(\boldsymbol{x}_k \mid \boldsymbol{x}_0)\right\|_2^2\right], \tag{6}$$

where $k \sim \mathcal{U}[0, K]$, $\boldsymbol{x}_0 \sim p$, $\boldsymbol{x}_k \sim q_k(\cdot \mid \boldsymbol{x}_0)$, q_k is the forward perturbation kernel in Eq. 4, and $\lambda(k) > 0$ is a weight. This tractable target $\nabla \log q_k(\boldsymbol{x}_k \mid \boldsymbol{x}_0)$ equals the marginal score $\nabla \log p(\boldsymbol{x}_k)$ in expectation conditioned on $(\boldsymbol{x}_k, k)$. After training, sampling proceeds by drawing $\hat{\boldsymbol{x}}_K \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$ and numerically solving Eq. 5, yielding approximate samples $\hat{\boldsymbol{x}}_0$ from $p(\boldsymbol{x}_0)$. This formulation naturally extends to conditional generation, enabling a diffusion-based policy representation. A diffusion policy represents a stochastic policy $\pi(a|s)$ via a conditional score network $\boldsymbol{\epsilon}_\varphi(a_k, s, k)$ [18], trained by minimizing

$$L(\varphi) = \mathbb{E}\left[\lambda(k) \left\|\boldsymbol{\epsilon}_\varphi(a_k, s, k) - \nabla \log q_k(a_k|a_0)\right\|_2^2\right], \tag{7}$$

[1] We use k to distinguish the diffusion time from the MDP time index t.

where a_k is obtained by forward noising $a_0 \sim \pi(\cdot|s)$. The resulting diffusion policy $\pi_\varphi(\hat{a}_0|s)$ is induced by solving

$$\mathrm{d}\hat{a}_k = \left[\boldsymbol{\mu}(\hat{a}_k, s, k) - \frac{1}{2}\sigma(k)^2 \boldsymbol{\epsilon}_\varphi\left(\hat{a}_k, s, k\right)\right] \mathrm{d}k. \tag{8}$$

3 Method

In this section we present OIDP, an offline extension of the inverse soft Q-learning framework. To enable stable offline learning, we proceed in three tightly coupled steps. First, we formulate a model-based conservative objective in Q-space, where the policy occupancy is estimated through model rollouts, and derive an upper bound on the gap to the oracle IRL objective. Second, we instantiate both the dynamics model and the uncertainty estimator with a single score-based diffusion network, and relate to the objective gap bound. Finally, we cast the policy into the same score-network form and couple it with the diffusion dynamics into a unified diffusion planner that performs trajectory-guided sampling during execution. The overview of OIDP is illustrated in Fig. 2.

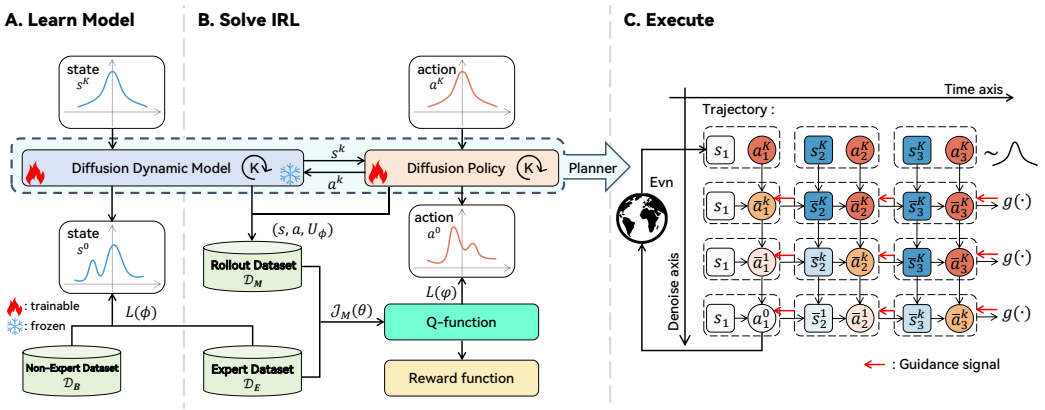


Figure 2: Overview of the OIDP framework, which comprises three components: **A.** learning a diffusion model to approximate dynamics and estimate uncertainty; **B.** optimizing the conservative objective in Q-space via model rollouts; **C.** at execution, performing trajectory-guided sampling by interleaving the diffusion policy and the diffusion dynamics to obtain high-value actions.

3.1 Model-based conservative objective in Q-space

In the offline setting, a demonstration dataset $\mathcal{D} = \mathcal{D}_E \cup \mathcal{D}_B$ is given, where $\mathcal{D}_E$ and $\mathcal{D}_B$ denote the expert dataset and the non-expert dataset, respectively. The non-expert dataset $\mathcal{D}_B$ is typically collected by aggregating trajectories independently sampled from B non-expert policies $\pi_1, \pi_2, \ldots, \pi_B$. Let π_{mix} denote the resulting mixture non-expert policy, and let ρ_{mix} denote its occupancy measure under the real dynamics model P. A naïve approximation replaces the policy-induced occupancy measure ρ_π in Eq. 3 with ρ_{mix}, yielding the following offline objective:

$$\max_{Q\in\Omega} \mathbb{E}_{\rho_E}[(\mathcal{T}^\pi Q)(s,a)] - \mathbb{E}_{\rho_{\text{mix}}}[(\mathcal{T}^\pi Q)(s,a)] - \alpha H(\pi) - \psi(\mathcal{T}^\pi Q). \tag{9}$$

This optimization is still restricted to the fixed offline dataset. When the diversity of non-expert data is limited, this approximation overfits and fails to generalize beyond the data support. A model-based approach can overcome this limitation by synthesizing on-policy transitions through model rollouts [12, 13].

To this end, we learn a dynamics model $\hat{P}$ from $\mathcal{D}$ and impose a conservative penalty motivated by model-based offline RL [19, 20] to prevent overestimation in regions with insufficient data coverage. The model-based conservative objective is given by:

$$\max_{Q\in\Omega} \mathcal{J}_M(Q) = \mathbb{E}_{\rho_E}[(\mathcal{T}^\pi Q)(s,a)] - \mathbb{E}_{\hat{\rho}_\pi}[(\hat{\mathcal{T}}^\pi Q - U)(s,a)] - \alpha H(\pi) - \psi(\hat{\mathcal{T}}^\pi Q - U), \tag{10}$$

where $\hat{\rho}_\pi$ denotes the occupancy measure induced by rolling out π under the learned dynamics model $\hat{P}$, and $(\hat{\mathcal{T}}^\pi Q)(s,a) = Q(s,a) - \gamma \mathbb{E}_{s' \sim \hat{P}(\cdot|s,a)} V^\pi(s')$. ρ_E is generally considered to be estimated from the expert dataset $\mathcal{D}_E$ and therefore does not rely on $\hat{P}$ in the expert expectation term $\mathbb{E}_{\rho_E}[(\mathcal{T}^\pi Q)(s,a)]$. $U(s,a) \leq 0$ is a penalty function that quantifies uncertainty in $\hat{P}(\cdot|s,a)$. This can be interpreted as defining a conservative Bellman operator $(\hat{\mathcal{B}}^\pi_M Q)(s,a) = r(s,a) + U(s,a) + \gamma \mathbb{E}_{s' \sim \hat{P}(\cdot|s,a)} V^\pi(s')$, which prevents overestimation of Q, similar to CQL [21]. Unlike model-free approaches that are bound to the fixed data support, the model-based occupancy $\hat{\rho}_\pi$ adapts to the current policy π, enabling the optimization to explore policy-relevant regions through model rollouts while the penalty ensures conservatism in unreliable regions.

We first show that the proposed offline objective $\mathcal{J}_M$ possesses a desirable property similar to that of $\mathcal{J}$ Eq. 3 [9].

Lemma 3.1. *If π is fixed during the Q-update and the regularizer $\psi(\cdot)$ is convex, then the resulting subproblem objective $\mathcal{J}_M(Q;\pi)$ is concave in Q.*

The proof is provided in Appendix C.1. As a consequence, each Q-update subproblem $\mathcal{J}_M(Q;\pi)$ is well behaved. Let $Q^*_M = \operatorname{argmax}_Q \mathcal{J}_M(Q)$ denotes the optimal solution to the global problem $\mathcal{J}_M(Q)$, and the corresponding IRL solution satisfies the saddle-point conditions:

$$\pi^*_M = \frac{1}{Z_s} \exp(\frac{1}{\alpha} Q^*_M(s,a)), \tag{11a}$$

$$r^*_M(s,a) = (\hat{\mathcal{T}}^{\pi^*_M} Q^*_M)(s,a) - U(s,a). \tag{11b}$$

The incorporation of $\hat{P}$ and $U(s,a)$ leads to an objective shift. The following lemma describes the matching gap between the proposed objective $\mathcal{J}_M$ and the original objective $\mathcal{J}$.

Lemma 3.2. *For any Q, the gap between $\mathcal{J}(Q)$ in Eq. 3 and $\mathcal{J}_M(Q)$ in Eq. 10 satisfies:*

$$\Delta\mathcal{J} = \gamma \mathbb{E}_{\rho_\pi}\left[\sum_{s'} (P(s'|s,a) - \hat{P}(s'|s,a)) \hat{V}_H(s')\right] - \mathbb{E}_{\hat{\rho}_\pi}[U(s,a)] + \psi(\hat{\mathcal{T}}^\pi Q - U) - \psi(\mathcal{T}^\pi Q), \tag{12}$$

where $\hat{V}_H(s) = \mathbb{E}_\pi\left[\sum_{t=0}^\infty \gamma^t \alpha H(\pi(\cdot|s_t)) \mid s_0 = s, \hat{P}\right]$ denotes the entropy value function induced by π and $\hat{P}$.

The proof is provided in Appendix C.2. Lemma 3.2 shows that the objective gap decomposes into a dynamics model mismatch term, a conservative penalty term, and a regularizer-shift term. To further analyze the controllability of the gap between $\mathcal{J}$ and $\mathcal{J}_M$, we introduce the following reasonable assumption:

Assumption 3.3. There exist finite constants $C_H, C_V, C_U, L_\psi, C_\psi > 0$ such that, for all policies π and states s, (i) $|H(\pi(\cdot|s))| \leq C_H$, (ii) $|V^\pi(s)| \leq C_V$ and (iii) $|U(s,a)| \leq C_U$. (iv) The pointwise regularizer ζ is L_ψ-Lipschitz and bounded by C_ψ on the reward range implied by the above bounds.

The assumptions are standard in the theoretical analysis of RL. (i) is maintained in practice by setting a target entropy as in [22]. (ii) is a standard regularity condition in soft RL theory, automatically satisfied when the optimization is restricted to a bounded function class. (iii) is standard for conservative penalty functions in model-based offline RL [19, 20]. Together, these bounds imply that the recovered rewards lie in a bounded interval, on which (iv) is satisfied by any continuously differentiable convex pointwise regularizer ζ.

Theorem 3.4. *Under Assumption 3.3, the objective gap satisfies:*

$$\begin{aligned} |\Delta\mathcal{J}| \leq & \left[\frac{\gamma(\alpha C_H + C_\psi)}{1-\gamma} + \gamma L_\psi C_V\right] \mathbb{E}_{\rho_\pi}\left[\|P(\cdot|s,a) - \hat{P}(\cdot|s,a)\|_1\right] \\ & + \mathbb{E}_{\hat{\rho}_\pi}[|U(s,a)|] + L_\psi \, \mathbb{E}_{\rho_\pi}[|U(s,a)|]. \end{aligned} \tag{13}$$

The proof is provided in Appendix C.3. Theorem 3.4 provides further insights into the roles of the learned dynamics model and the penalty function within our proposed Offline IRL formulation and provides a basis for algorithm design. Therefore, to tighten the bound on this objective discrepancy, it is crucial to obtain high-quality representations of the dynamics model and the penalty function.

3.2 Score-based dynamics and uncertainty estimation

To obtain such high-quality representations, we turn to score-based diffusion models, which can effectively capture complex and potentially multimodal state transition distributions. Furthermore, since the score function implicitly encodes the underlying data density, it inherently contains the structural information required for uncertainty quantification. We therefore employ a single score network to both approximate the environment dynamics P and formulate the conservative penalty function $U(s, a)$.

Specifically, we construct a score network $\boldsymbol{\varsigma}_\phi(s'_k, s, a, k)$. Using offline transitions $(s, a, s') \sim \mathcal{D}$, we sample $s'_k \sim q_k(\cdot \mid s')$ from the forward perturbation kernel and train the network to approximate the conditional score $\nabla \log p(s'_k \mid s, a)$ of the resulting noisy marginal via DSM:

$$L(\phi) = \mathbb{E}_{\mathcal{D},k,s'_k}\left[\lambda(k)\left\|\boldsymbol{\varsigma}_\phi(s'_k, s, a, k) - \nabla \log q_k(s'_k \mid s')\right\|_2^2\right]. \tag{14}$$

Then, we initialize a sample $\hat{s}'_K$ from a Gaussian distribution and solve the following ODE process to obtain $\hat{s}'_0$, thereby achieving the high-fidelity representation of the state transition $P(s'|s, a)$.

$$\mathrm{d}\hat{s}'_k = \left[\boldsymbol{\mu}(\hat{s}'_k, s, a, k) - \frac{1}{2}\sigma(k)^2\boldsymbol{\varsigma}_\phi\left(\hat{s}'_k, s, a, k\right)\right]\mathrm{d}k, \tag{15}$$

where $\boldsymbol{\mu}$ is the drift coefficient, and σ is the diffusion coefficient, following the standard DDPM setting [15]. For the trained score network $\boldsymbol{\varsigma}_\phi$, its theoretical residual relative to the true dynamics P directly measures fitting error (i.e., epistemic uncertainty) rather than transition dispersion (i.e., aleatoric uncertainty):

$$U_\phi(s, a) = \mathbb{E}_{k\sim\mathcal{U}[0,K]}\mathbb{E}_{s'_k\sim p(\cdot|s,a,k)}\left[\lambda(k)\|\boldsymbol{\varsigma}_\phi(s'_k, s, a, k) - \nabla \log p(s'_k|s, a)\|_2^2\right]. \tag{16}$$

This theoretical residual remains zero when the score network is Bayes-optimal, even if transitions are stochastic. This motivates using it to characterize uncertainty arising from imperfect model fitting. Its theoretical relationship with the model-mismatch error is as follows.

Lemma 3.5. *For any (s, a), let $\hat{P}_\phi(\cdot|s, a)$ be the transition distribution induced by the score network $\boldsymbol{\varsigma}_\phi$ through the reverse ODE. Then, the following results hold:*

$$\|P(\cdot|s, a) - \hat{P}_\phi(\cdot|s, a)\|_1 \lesssim C_1 \cdot \sqrt{U_\phi(s, a)} + C_2, \tag{17}$$

where C_1 and C_2 are constants depending on $\boldsymbol{\mu}$, σ and K.

The proof is provided in Appendix C.4. Substituting Lemma 3.5 into Theorem 3.4, we obtain the following corollary, which characterizes how the score-based diffusion modeling controls the objective gap bound.

Corollary 3.6. *Let $\bar{U}_P := \mathbb{E}_{\rho_\pi}[U_\phi(s, a)]$, and $U(s, a) := \xi\, U_\phi(s, a)$ with scaling coefficient $\xi < 0$. Under the conditions of Theorem 3.4 and Lemma 3.5, the objective gap is bounded by:*

$$|\Delta\mathcal{J}| \lesssim \left[\frac{\gamma(\alpha C_H + C_\psi + C_U)}{1-\gamma} + \gamma L_\psi C_V\right]\left(C_1\sqrt{\bar{U}_P} + C_2\right) - \xi(1 + L_\psi)\bar{U}_P. \tag{18}$$

The proof is provided in Appendix C.5. Consequently, minimizing the DSM loss $L(\phi)$ can improve score accuracy and thereby tighten the objective gap bound. Furthermore, Corollary 3.6 provides a policy-specific bound, emphasizing score accuracy in regions visited by π rather than uniformly across the entire state-action space.

In practice, the marginal score $\nabla \log p(s'_k \mid s, a)$ is unavailable, so we use the DSM residual $\widetilde{U}_\phi(s, a)$, defined as the conditional expectation of the per-sample DSM loss in Eq. 14 given (s, a). It relates to the theoretical residual $U_\phi(s, a)$ as follows:

$$\widetilde{U}_\phi(s, a) = U_\phi(s, a) + \mathbb{E}[\lambda(k)\,\mathrm{Var}(\nabla \log q_k(s'_k \mid s') \mid s'_k, s, a, k) \mid s, a], \tag{19}$$

where vector variance denotes the trace of the covariance. The second term is irreducible aleatoric uncertainty. For the Gaussian forward process, its contribution at each noise level is proportional to the expected posterior transition variance $\mathbb{E}[\mathrm{Var}(s'|s'_k, s, a, k)|s, a, k]$. By the law of total variance,

$\mathbb{E}[\mathrm{Var}(s'|s'_k, s, a, k)|s, a, k] \leq \mathrm{Var}(s'|s, a)$. Thus, compared with directly computing sample variance, conditional denoising in the DSM residual exploits information in s'_k to reduce the remaining transition uncertainty and mitigate sensitivity to aleatoric uncertainty. The DSM residual $\widetilde{U}_\phi$ provides a theoretically motivated conservative penalty, allowing a single score network to serve as both a dynamics model and an uncertainty estimator without additional ensembles.

For rollout pairs without real next states, we generate $\hat{s}' \sim \hat{P}_\phi(\cdot \mid s, a)$ through $\boldsymbol{\varsigma}_\phi$, then perturb $\hat{s}'_{k_j} \sim q_{k_j}(\cdot \mid \hat{s}')$ to compute an empirical proxy for the DSM residual:

$$\widehat{U}_\phi(s, a) = \frac{1}{N} \sum_{j=1}^{N} \lambda(k_j) \left\| \boldsymbol{\varsigma}_\phi(\hat{s}'_{k_j}, s, a, k_j) - \nabla \log q_{k_j}(\hat{s}'_{k_j} \mid \hat{s}') \right\|_2^2. \tag{20}$$

3.3 Unified diffusion planner for guided sampling

Corollary 3.6 localizes the approximation requirement to the occupancy induced by π, which in turn makes faithful sampling from the energy-based policy $\pi \propto \exp(\frac{1}{\alpha} Q(s, a))$ itself the next bottleneck. Since this distribution is intractable to sample directly in continuous spaces and is often multimodal, we train a diffusion policy following Eq. 7 to represent it effectively, where $\nabla \log p(a_k|s)$ is estimated using the learned Q-function following [18]. This approach offers a crucial structural advantage: both the dynamics model and the policy share the mathematical formulation of score networks. Consequently, this enables the construction of a unified diffusion planner during execution, which seamlessly integrates dynamics knowledge into every step of policy sampling directly within the score space, achieving trajectory-guided sampling inspired by [17] to fully leverage the learned dynamics.

The diffusion planner initializes a trajectory $\tau^K_{t:t+W} = \{s^K_{t:t+W}, a^K_{t:t+W}\}$ from a Gaussian distribution.[2] Given the current state s_t, the planner then executes Eq. 15 and Eq. 8 in single-step along the time axis, following the Markov property. Here, W denotes the planning horizon, analogous to that in model predictive control [23].

As illustrated in Fig. 2.C, the diffusion planner first executes a single denoising step through Eq. 8 to denoise the noisy action a^{k+1}_t conditioned on the predicted clean state $\bar{s}^k_t$ [3] and obtain the updated noisy action a^k_t and its corresponding clean action $\bar{a}^k_t$. Subsequently, a single denoising step through Eq. 15 is applied to denoise the noisy state s^{k+1}_{t+1} conditioned on $(\bar{s}^k_t, \bar{a}^k_t)$ and obtain the updated noisy state s^k_{t+1} and its corresponding clean state $\bar{s}^k_{t+1}$. This process is executed autoregressively along the time axis.

Since the diffusion planner performs a single-step denoising process across the entire trajectory, past data remains not fully denoised. This structure allows guidance gradients from future data to guide the sampling of past data, thereby enabling the generation of action a_t that respects long-horizon dynamics causality. Accordingly, the policy score used during sampling is estimated by

$$\nabla \log p(a^k_t|s_t) = \boldsymbol{\epsilon}_\varphi\left(a^k_t, s_t, k\right) + \kappa \nabla g(\bar{\tau}^k_{t:t+W}), \tag{21}$$

where $g(\cdot)$ is the guidance cost function. Considering the cumulative effect of noise propagation along the trajectory, where small noise in early data accumulates into large noise over the time axis, the noise level parameter $k = k(t)$ is designed to increase with the time index. The pseudocode of the guided sampling algorithm and a theoretical analysis of its performance guarantee are provided in Appendix B.1.

3.4 OIDP algorithm

We summarize the pseudocode of OIDP as shown in Appendix B.2. In the implementation, we alternate between Q-update and policy optimization to avoid the computational burden of repeatedly solving for the policy π under each Q-function, and adopt the following practices.

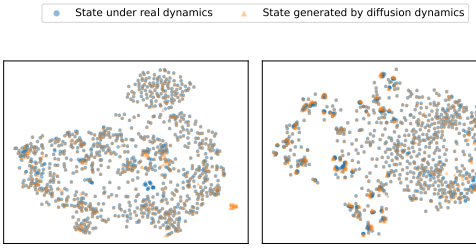
Reward regularizer. We use $\zeta(x) = \beta x^2$ with $\beta > 0$, giving $\psi_\rho(r) = \beta \mathbb{E}_{(s,a)\sim\rho}[r(s, a)^2]$. This induces the χ^2-divergence between the expert and policy occupancy measures in the original objective $\mathcal{J}$ [9] and satisfies Assumption 3.3.

[2] For clarity, superscripts denote diffusion steps and subscripts denote time steps.

[3] The bar denotes a clean estimate at the corresponding denoising step.

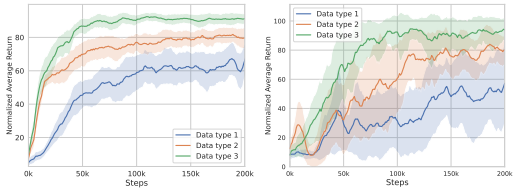
Table 1: Results on MuJoCo tasks. For each task, the experiments are carried out with three different data combinations: 1) 5k expert and 20k medium tuples, 2) 5k expert and 20k random tuples, and 3) 5k expert and 20k medium-replay tuples. Except for Hopper, which is trained for 100k steps, all other tasks are trained for 200k steps. The highest score across all algorithms is bold.

Dataset	Env	OIDP	CLARE	OMLIRL	SPRINQL	IQL	DWBC	DPBC
Exp. & Med.	Hopper	**97.1±4.8**	43.4±5.9	-0.2±0.1	59.9±28.0	20.6±14.7	58.0±19.5	41.6±10.6
	HalfCheetah	51.0±9.7	46.8±1.5	36.5±7.4	45.6±7.1	40.6±2.9	**52.7±13.2**	21.3±9.4
	Walker2d	**93.9±8.2**	56.9±11.2	6.4±11.3	73.9±23.6	0.1±0.4	64.4±27.2	48.9±10.8
	Ant	**74.6±7.8**	68.8±7.6	-16.9±6.5	37.7±13.6	1.7±32.7	56.8±10.1	58.6±7.3
Exp. & Rand.	Hopper	**94.4±7.6**	15.4±6.8	13.6±11.8	15.4±11.8	3.3±2.2	61.5±14.4	41.6±10.6
	HalfCheetah	**70.3±7.7**	2.6±0.7	2.1±5.2	6.1±1.4	0.4±0.5	35.4±10.5	21.3±9.4
	Walker2d	**65.4±27.4**	4.6±3.5	-0.2±0.0	5.8±8.2	-0.1±0.0	59.3±1.6	48.9±10.8
	Ant	54.2±10.7	8.4±8.0	-7.2±7.6	31.5±11.4	-1.4±2.1	**58.7±10.2**	58.6±7.3
Exp. & Medrep.	Hopper	**99.2±4.6**	16.2±6.2	0.2±0.1	36.1±17.8	3.6±2.1	34.2±11.2	41.6±10.6
	HalfCheetah	**66.4±6.5**	39.6±2.3	33.4±1.8	20.6±12.6	21.7±3.9	45.8±7.9	21.3±9.4
	Walker2d	**97.3±5.3**	22.3±10.1	4.2±3.1	67.4±24.2	-0.2±0.7	36.7±33.4	48.9±10.8
	Ant	**65.8±9.1**	43.4±8.8	-3.4±4.5	51.2±14.1	-27.4±13.8	49.8±12.4	58.6±7.3



(a) HalfCheetah (b) Ant

Figure 3: Visualization of a 2D projection on the state space of the real dynamics sampling data and diffusion dynamics sampling data.



(a) HalfCheetah (b) Walker2d

Figure 4: Results for three cumulative non-expert datasets with increasing size and coverage: 20k random (Type 1), +20k medium (Type 2), and +20k medium-replay (Type 3).

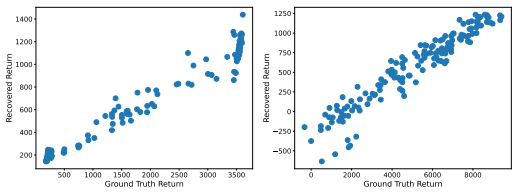
Action selection during execution. The inherent variance of the diffusion policy can hinder execution [24]. To mitigate this, we use the Q-function as the guidance cost g in trajectory-guided sampling to generate N_{cand} candidate actions a, and execute the one with the best Q-value.

4 Experiments

In this section, we conduct experiments to answer the following questions: (1) How does OIDP perform compared to baseline algorithms? (2) How effectively can diffusion models approximate the environmental dynamics? (3) What happens if we augment the state-action space coverage of non-expert data while maintaining the expert dataset? (4) How well does OIDP recover the ground-truth reward function? (5) How does each component contribute to OIDP's performance?

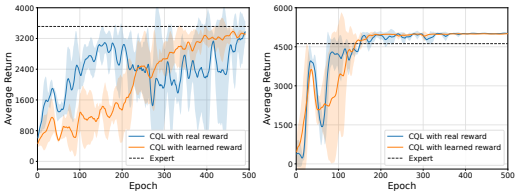
Experimental setup. We compare OIDP with state-of-the-art (SOTA) algorithms in Offline IRL and BC: 1) CLARE [13] and OMLIRL [12], advanced model-based Offline IRL algorithms using the minimax optimization framework; 2) IQL [9] and SPRINQL [25], which similarly optimize in Q-space; 3) DWBC [26], a SOTA BC algorithm that also incorporates non-expert data. Moreover, to demonstrate the limitations of relying solely on diffusion policies, we also consider a baseline algorithm, namely DPBC, by directly cloning expert behavior through diffusion policies. A more detailed introduction to these algorithms is provided in Appendix A. We evaluate them on four MuJoCo and four Adroit tasks across different dataset types from the D4RL benchmark [27]. More detailed experimental setup and hyperparameters for all algorithms are provided in Appendix D.

Metric. We use normalized return as the metric, i.e. $100 \times \frac{\text{method return} - \text{random return}}{\text{expert return} - \text{random return}}$ [25], and compute the mean and 95% confidence interval over five random seeds. The performance of different algorithms is evaluated using the average normalized returns from the last 10 evaluations.



(a) Hopper (b) HalfCheetah

Figure 5: The correlation between the ground truth returns and OIDP recovered returns.



(a) Hopper (b) Walker2d

Figure 6: Training CQL with the reward learned by OIDP and the ground-truth reward.

4.1 Performance

To address the first question, we present the performance comparison between OIDP and the baseline algorithms, as shown in Table 1. The results demonstrate that OIDP outperforms the baseline algorithms on almost all tasks, particularly on lower-quality data (e.g., *random*). In experiments where OIDP does not achieve the best performance, we notice the performance gap relative to leading baseline algorithms remains minimal. The results of Adroit tasks are provided in Appendix E.1.

To address the second question, we train dynamics models on the *random* dataset and evaluate them on held-out *medium-replay* transitions in two tasks. Table 2 quantitatively compares L2 prediction errors: the Diffusion model achieves lower errors than a Gaussian dynamics model, a common parameterization in model-based methods. Figure 3 qualitatively visualizes the overlap between real and diffusion-generated transition samples in a joint 2D t-SNE [28] embedding.

Table 2: L2 prediction errors of diffusion and Gaussian dynamics models on held-out transitions.

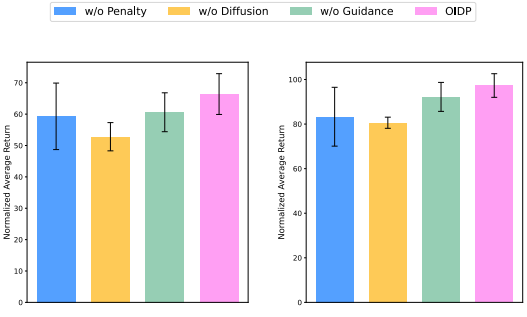
Env	Diffusion	Gaussian
HalfCheetah	**2.013**	7.023
Ant	**0.820**	7.545

To address the third question, we evaluate OIDP using three different compositions of non-expert data, as shown in Fig. 4. As the amount of non-expert data increases, the coverage of the non-expert state-action space gradually expands, leading to significant performance improvements. This matches Corollary 3.6, where broader data coverage facilitates more accurate score estimation.

To address the fourth question, we follow inverse-Q reward evaluation protocols [9, 25] and compare ground-truth and learned-reward returns on trajectories generated by progressively perturbing trained policies. Their Pearson correlation exceeds 0.95 in Fig. 5, suggesting that the learned reward preserves trajectory-quality rankings. We then relabel the offline dataset and train an independent CQL policy [21]. Fig. 6 shows that the learned reward supports downstream optimization with more stable training, possibly because its conservatism guides safer policy search.

4.2 Ablation study

To address the last question, we evaluate the contribution of key modules through several ablation studies. These include removing the penalty function (w/o Penalty), replacing all diffusion models with Gaussian models (w/o Diffusion), and disabling diffusion trajectory-guided sampling during execution (w/o Guidance). As shown in Fig. 7, the results demonstrate that removing the penalty function leads to performance degradation and increased variance, supporting the role of conservatism in mitigating extrapolation errors. The expressive capacity of diffusion models benefits continuous-space tasks. Moreover, by leveraging diffusion



(a) HalfCheetah (b) Walker2d

Figure 7: The normalized average return of the downstream policy with ablation variants.

trajectory-guided sampling, the policy can generate high-value actions, which enhances performance during execution, and this mechanism remains optional depending on specific requirements. Additional experiments, including a controlled evaluation of the uncertainty estimator, hyperparameter ablations, and computational cost comparisons, are provided in Appendix E.

5 Conclusion

This paper proposes OIDP, a method that leverages diffusion models for model-based Offline IRL. Compared to the minimax optimization framework, OIDP performs model-based conservative optimization solely in the Q-space, avoiding cross-space model error accumulation and providing theoretical analysis. To achieve its optimization objective, OIDP introduces a unified diffusion planner that integrates dynamics modeling, uncertainty estimation, and policy representation: it captures multimodal distributions and can tighten the objective gap bound, and utilizes internalized dynamics knowledge for trajectory-guided policy execution. Extensive experiments on the D4RL benchmark demonstrate that OIDP outperforms existing offline methods and can recover a meaningful reward function.

The limitations of our method are that it currently does not support image input tasks. Additionally, the diffusion-based design introduces non-negligible computational overhead during execution. A direction for future work is to leverage advanced world models to address complex real-world tasks. Moreover, another promising direction for future work is to further investigate the relationship between the two diffusion models based on Theorem 3.4.

Acknowledgments and Disclosures

This work was supported in part by the National Key R&D Program of China under Grant 2024YFB4709100; in part by the Strategic Priority Research Program of the Chinese Academy of Sciences under Grant XDA 0380000; in part by the National Natural Science Foundation of China under Grants 62425310, U2570206, and 4254106; the Beijing Natural Science Foundation under Grant L257011; and the Science and Technology Innovation Project of the China Academy of Chinese Medical Sciences under Grant CI2023C005YG. The authors declare no competing interests.

References

[1] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

[2] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *Nature*, 529(7587):484–489, 2016.

[3] Oriol Vinyals, Igor Babuschkin, Wojciech M Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David H Choi, Richard Powell, Timo Ewalds, Petko Georgiev, et al. Grandmaster level in starcraft ii using multi-agent reinforcement learning. *Nature*, 575(7782):350–354, 2019.

[4] David Silver, Satinder Singh, Doina Precup, and Richard S Sutton. Reward is enough. *Artificial intelligence*, 299:103535, 2021.

[5] Ahmed Hussein, Mohamed Medhat Gaber, Eyad Elyan, and Chrisina Jayne. Imitation learning: A survey of learning methods. *ACM Computing Surveys (CSUR)*, 50(2):1–35, 2017.

[6] Stéphane Ross and Drew Bagnell. Efficient reductions for imitation learning. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 661–668. JMLR Workshop and Conference Proceedings, 2010.

[7] Brian D Ziebart, Andrew L Maas, J Andrew Bagnell, Anind K Dey, et al. Maximum entropy inverse reinforcement learning. In *Aaai*, volume 8, pages 1433–1438. Chicago, IL, USA, 2008.

[8] Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. *Advances in neural information processing systems*, 29, 2016.

[9] Divyansh Garg, Shuvam Chakraborty, Chris Cundy, Jiaming Song, and Stefano Ermon. Iq-learn: Inverse soft-q learning for imitation. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 4028–4039. Curran Associates, Inc., 2021.

[10] Siqi Liu, Kay Choong See, Kee Yuan Ngiam, Leo Anthony Celi, Xingzhi Sun, and Mengling Feng. Reinforcement learning for clinical decision support in critical care: comprehensive review. *Journal of medical Internet research*, 22(7):e18477, 2020.

[11] B Ravi Kiran, Ibrahim Sobh, Victor Talpaert, Patrick Mannion, Ahmad A Al Sallab, Senthil Yogamani, and Patrick Pérez. Deep reinforcement learning for autonomous driving: A survey. *IEEE transactions on intelligent transportation systems*, 23(6):4909–4926, 2021.

[12] Siliang Zeng, Chenliang Li, Alfredo Garcia, and Mingyi Hong. When demonstrations meet generative world models: A maximum likelihood framework for offline inverse reinforcement learning. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 65531–65565. Curran Associates, Inc., 2023.

[13] Sheng Yue, Guanbo Wang, Wei Shao, Zhaofeng Zhang, Sen Lin, Ju Ren, and Junshan Zhang. CLARE: Conservative model-based reward learning for offline inverse reinforcement learning. In *The Eleventh International Conference on Learning Representations*, 2023.

[14] Gengyu Zhang and Yan Yan. Towards accurate and robust dynamics and reward modeling for model-based offline inverse reinforcement learning. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 611–618. IEEE, 2024.

[15] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Proceedings of Advances in Neural Information Processing Systems*, volume 33, pages 6840–6851, 2020.

[16] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *Proceedings of International Conference on Learning Representations*, 2021.

[17] Boyuan Chen, Diego Martí Monsó, Yilun Du, Max Simchowitz, Russ Tedrake, and Vincent Sitzmann. Diffusion forcing: Next-token prediction meets full-sequence diffusion. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 24081–24125. Curran Associates, Inc., 2024. doi: 10.52202/079017-0759.

[18] Xiaoyi Dong, Jian Cheng, and Xi Sheryl Zhang. Maximum entropy reinforcement learning with diffusion policy. In *Forty-second International Conference on Machine Learning*, 2025.

[19] Rahul Kidambi, Aravind Rajeswaran, Praneeth Netrapalli, and Thorsten Joachims. Morel: Model-based offline reinforcement learning. *Advances in neural information processing systems*, 33:21810–21823, 2020.

[20] Tianhe Yu, Garrett Thomas, Lantao Yu, Stefano Ermon, James Y Zou, Sergey Levine, Chelsea Finn, and Tengyu Ma. Mopo: Model-based offline policy optimization. *Advances in Neural Information Processing Systems*, 33:14129–14142, 2020.

[21] Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative q-learning for offline reinforcement learning. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1179–1191. Curran Associates, Inc., 2020.

[22] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 1861–1870. PMLR, 10–15 Jul 2018.

[23] Carlos E. García, David M. Prett, and Manfred Morari. Model predictive control: Theory and practice—a survey. *Automatica*, 25(3):335–348, 1989. ISSN 0005-1098.

[24] Chen-Hao Chao, Chien Feng, Wei-Fang Sun, Cheng-Kuang Lee, Simon See, and Chun-Yi Lee. Maximum entropy reinforcement learning via energy-based normalizing flow. *Advances in Neural Information Processing Systems*, 37:56136–56165, 2024.

[25] Huy Hoang, Tien Mai, and Pradeep Varakantham. Sprinql: Sub-optimal demonstrations driven offline imitation learning. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 136837–136872. Curran Associates, Inc., 2024. doi: 10.52202/079017-4348.

[26] Haoran Xu, Xianyuan Zhan, Honglei Yin, and Huiling Qin. Discriminator-weighted offline imitation learning from suboptimal demonstrations. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 24725–24742. PMLR, 17–23 Jul 2022.

[27] 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.

[28] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9:2579–2605, 2008.

[29] Alex James Chan and Mihaela van der Schaar. Scalable bayesian inverse reinforcement learning. In *International Conference on Learning Representations*, 2021.

[30] Konrad Zolna, Alexander Novikov, Ksenia Konyushkova, Caglar Gulcehre, Ziyu Wang, Yusuf Aytar, Misha Denil, Nando De Freitas, and Scott Reed. Offline learning from demonstrations and unlabeled experience. *arXiv preprint arXiv:2011.13885*, 2020.

[31] Mengjiao Yang, Sergey Levine, and Ofir Nachum. TRAIL: Near-optimal imitation learning with suboptimal data. In *International Conference on Learning Representations*, 2022.

[32] Geon-Hyeong Kim, Seokin Seo, Jongmin Lee, Wonseok Jeon, HyeongJoo Hwang, Hongseok Yang, and Kee-Eung Kim. DemoDICE: Offline imitation learning with supplementary imperfect demonstrations. In *International Conference on Learning Representations*, 2022.

[33] Harshit Sikchi, Qinqing Zheng, Amy Zhang, and Scott Niekum. Dual rl: Unification and new methods for reinforcement and imitation learning. *arXiv preprint arXiv:2302.08560*, 2023.

[34] Harshit Sikchi, Caleb Chuck, Amy Zhang, and Scott Niekum. A dual approach to imitation learning from observations with offline datasets. In Pulkit Agrawal, Oliver Kroemer, and Wolfram Burgard, editors, *Proceedings of The 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pages 1125–1147. PMLR, 06–09 Nov 2025.

[35] Quang Anh Pham, Janaka Chathuranga Brahmanage, Tien Anh Mai, and Akshat Kumar. IOSTOM: Offline imitation learning from observations via state transition occupancy matching. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[36] Michael Janner, Yilun Du, Joshua Tenenbaum, and Sergey Levine. Planning with diffusion for flexible behavior synthesis. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 9902–9915. PMLR, 17–23 Jul 2022.

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

[38] Zhendong Wang, Jonathan J Hunt, and Mingyuan Zhou. Diffusion policies as an expressive policy class for offline reinforcement learning. In *Deep Reinforcement Learning Workshop NeurIPS 2022*, 2022.

[39] 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.

[40] Zihan Ding and Chi Jin. Consistency models as a rich and efficient policy class for reinforcement learning. *CoRR*, abs/2309.16984, 2023.

[41] Donghyeon Ki, JunHyeok Oh, Seong-Woong Shim, and Byung-Jun Lee. Prior-guided diffusion planning for offline reinforcement learning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[42] Michael Psenka, Alejandro Escontrela, Pieter Abbeel, and Yi Ma. Learning a diffusion model policy from rewards via q-score matching. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

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

[44] Onur Celik, Zechu Li, Denis Blessing, Ge Li, Daniel Palenicek, Jan Peters, Georgia Chalvatzaki, and Gerhard Neumann. DIME: Diffusion-based maximum entropy reinforcement learning. In *Forty-second International Conference on Machine Learning*, 2025.

[45] Sitan Chen, Sinho Chewi, Jerry Li, Yuanzhi Li, Adil Salim, and Anru Zhang. Sampling is as easy as learning the score: theory for diffusion models with minimal data assumptions. In *The Eleventh International Conference on Learning Representations*, 2023.

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

[47] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.

[48] Sham Kakade and John Langford. Approximately optimal approximate reinforcement learning. In *Proceedings of the nineteenth international conference on machine learning*, pages 267–274, 2002.

A Related work.

Offline IRL To avoid the costly interactions with the environment, Offline IRL aims to infer the expert's reward function and policy from a pre-collected dataset. AVRIL [29] employs a variational Bayesian framework to jointly learn the approximate posterior distribution over the reward function and policy from the expert demonstration dataset. IQ-Learn [9] introduces an inverse Bellman operator, which can implicitly represent the reward function and policy by learning a soft Q-function, thus avoiding minimax optimization. However, the inherent distribution shift challenge in Offline IRL makes it difficult for the reward function and policy recovered solely from the expert demonstration dataset to extrapolate to out-of-distribution states. To alleviate this issue, some works employ the available non-expert demonstration dataset to augment training. ORIL [30] first learns the reward function through contrastive learning between expert and non-expert demonstration data, and then employs Offline RL to recover the policy. Both Offline ML-IRL [12] and CLARE [13] employ Gaussian distributions to model the dynamics, incorporating conservatism into policy optimization and reward optimization respectively. Nevertheless, these works primarily concentrate on the minimax optimization framework. Recently, SPRINQL [25] based on IQ-Learn weights the loss with different data levels, achieving stable training under the joint data distribution. Unlike previous works, our approach employs more expressive diffusion models than Gaussian models to uniformly represent dynamics and uncertainty, achieving stable training under a model-based conservative Q optimization framework.

BC Different from Offline IRL, BC aims to directly imitate the expert policy without recovering the reward function. State-of-the-art BC methods also incorporate non-expert data to capture environmental dynamics. TRAIL [31] trains a feature encoder on the non-expert demonstration dataset to extract a latent action space, which is then used for downstream expert policy imitation. DemoDICE [32] and DWBC [26] both train discriminators to identify high-quality state-action pairs from non-expert data and extract policies through weighted behavior cloning. ReCOIL [33] proposes a discriminator-free approach that optimizes the dual problem of the state-action occupancy matching. DILO [34] and IOSTOM [35] focus on the problem of Offline IL without expert actions by learning state-to-state functions and then deriving policies from the non-expert dataset to match expert behaviors. However, these approaches cannot obtain a meaningful reward function, which concisely represents the expert intention and can be applied to broad downstream tasks.

Diffusion models for decision making The multimodal distribution modeling capability of diffusion models (DMs) makes them well-suited for decision-making tasks with continuous state-action spaces. Early works have employed DMs as trajectory generators [36, 17] or policy representations [37] for behavior cloning. Diffusion Policy [37] trains a DM to directly approximate multimodal expert behaviors. Diffusion Forcing [17] combines DMs with next-token prediction models to generate trajectories autoregressively. Then, many works [38–41] have focused on applying DMs to Offline RL. Diffusion-QL [38] incorporates a Q-based action maximization term into the diffusion loss. PGDP [41] proposes a prior-guided sampling framework that replaces the standard Gaussian prior

with a learnable distribution and also incorporates a value regularizer. More recently, DMs have been applied in online RL to promote exploration and policy representation. QSM [42] employs a score network to directly match the gradient of the Q-function, avoiding differentiation on the whole diffusion chain. DIPO [43] uses the gradient of the Q-function to update actions in the replay buffer, followed by behavior cloning. To encourage exploration, MaxEnt DP [18] and DIME [44] employ DMs in MaxEnt RL without explicitly adding exploration noise. The above approaches train DMs to represent policies under the reward-free or the ground-truth reward setting. In contrast, we focus on using DMs in Offline IRL, which has not been considered before. Similar to MaxEnt DP, we train the DM to approximate the exponential of the Q-function. However, we also incorporate the DM that approximates environmental dynamics and serves as the uncertainty estimator, which mitigates the distribution shift issue in Offline IRL and enables dynamics knowledge guided sampling during execution.

B Algorithmic pseudocode and theoretical analysis.

B.1 The pseudocode of guided sampling algorithm and a theoretical analysis

Algorithm 1 Diffusion Planner Sampling with Guidance

Input: Model ς_ϕ ϵ_φ, scheduling matrix $\mathcal{K}$, guidance cost function $g(\cdot)$, initial state s_1.
Initialize Sample $\tau_{1:W}^M = \{s_{1:W}^M, a_{1:W}^M\}$ from a Gaussian distribution.
for row $m = M-1, \ldots, 0$ **do**
  for $t = 1, \ldots, W$ **do**
    Get k for $\mathcal{K}_{m,t}$
    Calculate a_t^m and $\bar{a}_t^m$ conditioned on a_t^{m+1}, $\bar{s}_t^m$ and k by Eq. 8.
    if $t < W$ **then**
      Calculate s_{t+1}^m and $\bar{s}_{t+1}^m$ conditioned on s_{t+1}^{m+1}, $\bar{s}_t^m$, $\bar{a}_t^m$ and k by Eq. 15.
    end if
  end for
  $a_1^m \leftarrow$AddGuidance$(a_1^m, \kappa \nabla g(\bar{\tau}_{1:1+W}^m))$ by Eq. 8 and 21.
end for
Return a_1^0

Theorem B.1. *Let the base diffusion policy before guidance be $\pi_b(a|s) \approx \frac{1}{Z_s}\exp(\frac{1}{\alpha}Q_\theta(s,a))$. During the execution phase, the diffusion planner employs an W-step trajectory cost function $g(\tau)$ for guidance, inducing an effective policy $\pi_g(a|s)$. Let $g(s,a)$ denote the marginal expected return of the W-step planning given (s,a), and let $Q^{\pi_b}(s,a)$ denote the action-value function of the base policy in the real environment. Assume the planning error is bounded: $\|g(s,a) - Q^{\pi_b}(s,a)\|_\infty \leq \varepsilon_p$, which arises from three sources: (i) dynamics model mismatch between $\hat{P}$ and P, (ii) truncation of the W-step rollout relative to the infinite-horizon return, and (iii) the bias of approximating Q^{π_b} with the learned Q_θ. For sufficiently small guidance scales $\kappa > 0$, the expected return of the guided policy in the real environment relative to the base policy satisfies:*

$$J(\pi_g) - J(\pi_b) \geq \frac{1}{1-\gamma}\mathbb{E}_{s\sim d_{\pi_g}}\left[\kappa \cdot \mathit{Var}_{a\sim\pi_b}(g(s,a)) - \mathcal{O}(\kappa^2) - 2\varepsilon_p\right]. \tag{22}$$

The proof is provided in Appendix C.6. Theorem B.1 profoundly elucidates the pivotal role of the diffusion planner during the execution phase. Incorporating the guidance function $g(\tau)$ essentially performs dynamics-aware W-step lookahead planning. The theorem demonstrates that, provided the expected variance gain $\kappa \cdot \mathrm{Var}(g)$ induced by trajectory guidance outweighs the bounded planning error $2\varepsilon_p$, the planner guarantees a strict performance improvement over the base policy π_b. Moreover, dynamics mismatch and the approximation bias of Q_θ are important sources of ε_p. Motivated by Theorem 3.4 and Corollary 3.6, training a high-fidelity score network and choosing an appropriate planning horizon help reduce ε_p, thereby improving the lower bound on performance improvement in Eq. 22.

B.2 The pseudocode of OIDP

Algorithm 2 Offline Inverse Reinforcement Learning with Unified Diffusion Planner (OIDP)

Input: offline dataset $\mathcal{D}_E$, $\mathcal{D}_B$, replay buffer $\mathcal{D}_M$, parameters θ, ϕ, φ, η_P, η_Q, η_π, and loop lengths N_{outer}, N_{inner}, T_{exec}.
for $j = 1$ **to** N_{outer} **do**
  Sample data from $\mathcal{D} = \mathcal{D}_E \cup \mathcal{D}_B$.
  Calculate $L(\phi)$ by Eq. 14.
  Update $\hat{P}_\phi$ by $\phi \leftarrow \phi - \eta_P \nabla_\phi L(\phi)$.
end for
for $i = 1$ **to** N_{inner} **do**
  if need sampling **then**
    Roll out the policy π_φ in the model $\hat{P}_\phi$ by Eq. 15, 20 and 8.
    Store sampling data in $\mathcal{D}_M$.
  end if
  Sample data from $\mathcal{D}_E$ and $\mathcal{D}_M$.
  Calculate $\mathcal{J}_M(\theta)$ by Eq. 10.
  Update Q_θ by $\theta \leftarrow \theta + \eta_Q \nabla_\theta \mathcal{J}_M(\theta)$.
  Calculate $L(\varphi)$ by Eq. 7.
  Update π_φ by $\varphi \leftarrow \varphi - \eta_\pi \nabla_\varphi L(\varphi)$.
end for
if need execution **then**
  for $t = 1$ **to** T_{exec} **do**
    Sample action a_t by algorithm 1.
    Execute the action a_t.
  end for
end if

C Theoretic proofs.

C.1 Proof of Lemma 3.1

Proof. We recall that

$$\begin{aligned}
&\mathcal{J}_M(Q;\pi) = \mathbb{E}_{\rho_E}[(\mathcal{T}^\pi Q)(s,a)] - \mathbb{E}_{\hat{\rho}_\pi}[(\hat{\mathcal{T}}^\pi Q)(s,a) - U(s,a)] - \alpha H(\pi) - \psi(\hat{\mathcal{T}}^\pi Q - U) \\
&= \mathbb{E}_{\rho_E}[(\mathcal{T}^\pi Q)(s,a)] - \mathbb{E}_{\hat{\rho}_\pi}[Q(s,a) - \gamma \mathbb{E}_{s' \sim \hat{P}(\cdot|s,a)} V^\pi(s') - U(s,a) + \alpha H(\pi)] - \psi(\hat{\mathcal{T}}^\pi Q - U) \\
&= \mathbb{E}_{\rho_E}[(\mathcal{T}^\pi Q)(s,a)] - (1-\gamma)\mathbb{E}_{\hat{p}_0}[V^\pi(s)] + \mathbb{E}_{\hat{\rho}_\pi}[U(s,a)] - \psi(\hat{\mathcal{T}}^\pi Q - U),
\end{aligned} \tag{23}$$

where $\hat{p}_0 = p_0$ is the initial state distribution. For fixed π, the soft value function $V^\pi(\cdot)$ admits the form [22]

$$V^\pi(s) = \mathbb{E}_{a \sim \pi(\cdot|s)}[Q(s,a)] + \alpha H(\pi(\cdot|s)). \tag{24}$$

Since π is fixed, V^π is affine in Q. Then the first term $(\mathcal{T}^\pi Q)(s,a) = Q(s,a) - \gamma \mathbb{E}_{s' \sim P(\cdot|s,a)} V^\pi(s')$ is affine, hence concave in Q. As $\gamma \in [0,1)$, the second term $-(1-\gamma)\mathbb{E}_{\hat{p}_0}[V^\pi(s)]$ is affine, hence concave in Q. As π is fixed and $U(\cdot,\cdot)$ is independent of Q, the third term $\mathbb{E}_{\hat{\rho}_\pi}[U(s,a)]$ is concave in Q. Since $\hat{\mathcal{T}}^\pi Q$ is affine in Q by the same argument as $\mathcal{T}^\pi Q$ and U is independent of Q, $\hat{\mathcal{T}}^\pi Q - U$ is affine in Q. If the regularizer $\psi(\cdot)$ is convex, the last term $-\psi(\hat{\mathcal{T}}^\pi Q - U)$ is also concave in Q (composition of a convex function with an affine function is convex; negation gives concavity).

The proof is completed. □

C.2 Proof of Lemma 3.2

Proof. We recall that

$$\mathcal{J}(Q) = \mathbb{E}_{\rho_E}[(\mathcal{T}^\pi Q)(s,a)] - \mathbb{E}_{\rho_\pi}[(\mathcal{T}^\pi Q)(s,a)] - \alpha H(\pi) - \psi(\mathcal{T}^\pi Q), \tag{25a}$$

$$\mathcal{J}_M(Q) = \mathbb{E}_{\rho_E}[(\mathcal{T}^\pi Q)(s,a)] - \mathbb{E}_{\hat{\rho}_\pi}[(\hat{\mathcal{T}}^\pi Q)(s,a) - U(s,a)] - \alpha H(\pi) - \psi(\hat{\mathcal{T}}^\pi Q - U). \tag{25b}$$

We can get the gap as below:

$$\begin{aligned}\Delta\mathcal{J} &= \mathcal{J}(Q) - \mathcal{J}_M(Q) \\ &= \underbrace{\mathbb{E}_{\hat{\rho}_\pi}[(\hat{\mathcal{T}}^\pi Q)(s,a)] - \mathbb{E}_{\rho_\pi}[(\mathcal{T}^\pi Q)(s,a)]}_{E_1\text{: dynamics mismatch}} \underbrace{-\mathbb{E}_{\hat{\rho}_\pi}[U(s,a)]}_{E_2\text{: conservative penalty}} \underbrace{+\psi(\hat{\mathcal{T}}^\pi Q - U) - \psi(\mathcal{T}^\pi Q)}_{E_3\text{: regularizer shift}}.\end{aligned} \tag{26}$$

Recall $\rho_\pi(s,a) = (1-\gamma)\pi(a|s)\sum_{t=0}^{\infty}\gamma^t \Pr(s_t = s|P,\pi,p_0)$. For simplicity, let $P_t^\pi(s)$ represent $\Pr(s_t = s|P,\pi,p_0)$.

$$\begin{aligned}E_1 &= (1-\gamma)\sum_{t=0}^{\infty}\sum_{s,a}\gamma^t \hat{P}_t^\pi(s)\pi(a|s)(\hat{\mathcal{T}}^\pi Q)(s,a) \\ &\quad - (1-\gamma)\sum_{t=0}^{\infty}\sum_{s,a}\gamma^t P_t^\pi(s)\pi(a|s)(\mathcal{T}^\pi Q)(s,a) \\ &= (1-\gamma)\sum_{t=0}^{\infty}\sum_{s,a}\gamma^t \pi(a|s)\Bigg[Q(s,a)\left(\hat{P}_t^\pi - P_t^\pi\right) \\ &\quad - \gamma\sum_{s'}V^\pi(s')\left(\hat{P}_t^\pi \hat{P}(s'|s,a) - P_t^\pi P(s'|s,a)\right)\Bigg]\end{aligned} \tag{27}$$

Using the recurrence relation of the state distribution $P_{t+1}^\pi(s') = \sum_{s,a}P_t^\pi(s)P(s'|s,a)\pi(a|s)$, we obtain the following result:

$$\begin{aligned}E_1 &= (1-\gamma)\sum_{t=0}^{\infty}\gamma^t\Bigg[\sum_{s,a}\pi(a|s)Q(s,a)\left(\hat{P}_t^\pi(s) - P_t^\pi(s)\right) \\ &\quad - \gamma\sum_{s'}V^\pi(s')\left(\hat{P}_{t+1}^\pi(s') - P_{t+1}^\pi(s')\right)\Bigg]\end{aligned} \tag{28}$$

Transform the summation index in the second term by setting $k = t+1$, then:

$$\begin{aligned}E_1 &= (1-\gamma)\Bigg[\sum_{t=0}^{\infty}\gamma^t\sum_{s,a}\pi(a|s)Q(s,a)\left(\hat{P}_t^\pi(s) - P_t^\pi(s)\right) \\ &\quad - \sum_{k=1}^{\infty}\gamma^k\sum_{s'}V^\pi(s')\left(\hat{P}_k^\pi(s') - P_k^\pi(s')\right)\Bigg]\end{aligned} \tag{29}$$

Since the initial distributions are identical $\hat{P}_0^\pi(s) = P_0^\pi(s) = p_0$, the summation of the second term can be extended to $k = 0$:

$$\begin{aligned}E_1 &= (1-\gamma)\sum_{t=0}^{\infty}\gamma^t\left[\sum_{s,a}\pi(a|s)Q(s,a)\left(\hat{P}_t^\pi(s) - P_t^\pi(s)\right) - \sum_{s}V^\pi(s)\left(\hat{P}_t^\pi(s) - P_t^\pi(s)\right)\right] \\ &= (1-\gamma)\sum_{t=0}^{\infty}\gamma^t\sum_{s}\left[\sum_{a}\pi(a|s)Q(s,a) - V^\pi(s)\right]\left(\hat{P}_t^\pi(s) - P_t^\pi(s)\right)\end{aligned} \tag{30}$$

Substituting the identity for entropy-regularized policies $\sum_a \pi(a|s)Q(s,a) = V^\pi(s) - \alpha H(\pi(\cdot|s))$ into the equation yields:

$$E_1 = (1-\gamma)\sum_{t=0}^{\infty}\gamma^t\sum_{s}\alpha H(\pi(\cdot|s))\left(P_t^\pi(s) - \hat{P}_t^\pi(s)\right) \tag{31}$$

We define the state occupancy measure as $d_\pi(s) = (1-\gamma)\sum_{t=0}^{\infty}\gamma^t\Pr(s_t = s|P,\pi,p_0)$ and entropy value function $\hat{V}_H(s) = \mathbb{E}_\pi\left[\sum_{t=0}^{\infty}\gamma^t\alpha H(\pi(\cdot|s_t)) \mid s_0 = s\right]$, with $\hat{V}_H(s) = \alpha H(\pi(\cdot|s)) + \gamma\mathbb{E}_{a\sim\pi(\cdot|s),s'\sim\hat{P}(\cdot|s,a)}[\hat{V}_H(s')]$. Then plugging them into the Eq. 31:

$$E_1 = \sum_s \hat{V}_H(s)(d_\pi(s) - \hat{d}_\pi(s)) - \gamma\sum_{s,a,s'}\pi(a|s)\hat{P}(s'|s,a)\hat{V}_H(s')(d_\pi(s) - \hat{d}_\pi(s)) \tag{32}$$

Since the state occupancy has the following recursive relationship:

$$d_\pi(s') - \hat{d}_\pi(s') = \gamma \sum_{s,a} \pi(a|s) \left[(d_\pi(s) - \hat{d}_\pi(s)) \hat{P}(s'|s,a) + d_\pi(s)(P(s'|s,a) - \hat{P}(s'|s,a)) \right] \tag{33}$$

By rearranging terms, we obtain:

$$\begin{aligned} &\sum_{s,a} \pi(a|s) \hat{P}(s'|s,a)(d_\pi(s) - \hat{d}_\pi(s)) \\ &= \frac{1}{\gamma}(d_\pi(s') - \hat{d}_\pi(s')) - \sum_{s,a} \pi(a|s) d_\pi(s)(P(s'|s,a) - \hat{P}(s'|s,a)) \end{aligned} \tag{34}$$

By plugging Eq. 34 into Eq. 32, we obtain:

$$\begin{aligned} E_1 &= \gamma \sum_{s'} \hat{V}_H(s') \sum_{s,a} \pi(a|s) d_\pi(s)(P(s'|s,a) - \hat{P}(s'|s,a)) \\ &= \gamma \mathbb{E}_{\rho_\pi} \left[\sum_{s'} (P(s'|s,a) - \hat{P}(s'|s,a)) \hat{V}_H(s') \right]. \end{aligned} \tag{35}$$

By plugging Eq. 35 into Eq. 26, we obtain the gap between two objectives as below:

$$\begin{aligned} \Delta\mathcal{J} = \mathcal{J}(Q) - \mathcal{J}_M(Q) =& \gamma \mathbb{E}_{\rho_\pi} \left[\sum_{s'} (P(s'|s,a) - \hat{P}(s'|s,a)) \hat{V}_H(s') \right] \\ &- \mathbb{E}_{\hat{\rho}_\pi}[U(s,a)] + \psi(\hat{\mathcal{T}}^\pi Q - U) - \psi(\mathcal{T}^\pi Q). \end{aligned} \tag{36}$$

The proof is completed. □

C.3 Proof of Theorem 3.4

Proof. According to Eq. 26 and Eq. 36,

$$|\Delta\mathcal{J}| = |\mathcal{J}(Q) - \mathcal{J}_M(Q)| \leq |E_1| + |E_2| + |E_3|, \tag{37}$$

where E_1, E_2 and E_3 are defined in Eq. 26. Below we bound each term.

Bounding E_1. By Eq. 35,

$$|E_1| \leq \gamma \, \mathbb{E}_{\rho_\pi} \left[\sum_{s'} \left| P(s'|s,a) - \hat{P}(s'|s,a) \right| \cdot \left| \hat{V}_H(s') \right| \right]. \tag{38}$$

Under Assumption 3.3,

$$|\hat{V}_H(s)| = \left| \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t \alpha H(\pi(\cdot|s_t)) \,\middle|\, s_0 = s \right] \right| \leq \sum_{t=0}^{\infty} \gamma^t \alpha |H(\pi(\cdot|s_t))| \leq \frac{\alpha C_H}{1-\gamma}, \tag{39}$$

which yields

$$|E_1| \leq \frac{\gamma \alpha C_H}{1-\gamma} \, \mathbb{E}_{\rho_\pi} \left[\| P(\cdot|s,a) - \hat{P}(\cdot|s,a) \|_1 \right]. \tag{40}$$

Bounding E_2.

$$|E_2| = \mathbb{E}_{\hat{\rho}_\pi}[|U(s,a)|]. \tag{41}$$

Bounding E_3. By the occupancy-weighted separable form of the regularizer,

$$\begin{aligned} E_3 &= \psi_{\hat{\rho}_\pi}(\hat{\mathcal{T}}^\pi Q - U) - \psi_{\rho_\pi}(\mathcal{T}^\pi Q) \\ &= \mathbb{E}_{\hat{\rho}_\pi} \left[\zeta \left((\hat{\mathcal{T}}^\pi Q)(s,a) - U(s,a) \right) \right] - \mathbb{E}_{\rho_\pi} \left[\zeta \left((\mathcal{T}^\pi Q)(s,a) \right) \right]. \end{aligned} \tag{42}$$

Adding and subtracting $\mathbb{E}_{\rho_\pi}[\zeta((\hat{\mathcal{T}}^\pi Q)(s,a) - U(s,a))]$, we decompose E_3 into an occupancy-shift part and a function-difference part:

$$\begin{aligned} E_3 = &\underbrace{\mathbb{E}_{\hat{\rho}_\pi}\left[\zeta((\hat{\mathcal{T}}^\pi Q)(s,a) - U(s,a))\right] - \mathbb{E}_{\rho_\pi}\left[\zeta((\hat{\mathcal{T}}^\pi Q)(s,a) - U(s,a))\right]}_{E_3^{(a)}} \\ &+ \underbrace{\mathbb{E}_{\rho_\pi}\left[\zeta((\hat{\mathcal{T}}^\pi Q)(s,a) - U(s,a)) - \zeta((\mathcal{T}^\pi Q)(s,a))\right]}_{E_3^{(b)}}. \end{aligned} \tag{43}$$

For $E_3^{(a)}$, since ζ is bounded by C_ψ on the reward range, Hölder's inequality gives

$$|E_3^{(a)}| \le C_\psi \, \|\hat{\rho}_\pi - \rho_\pi\|_1. \tag{44}$$

The occupancy mismatch $\|\hat{\rho}_\pi - \rho_\pi\|_1$ is in turn controlled by the simulation lemma. Following the recursion $P_{t+1}^\pi(s') = \sum_{s,a} \pi(a|s) P_t^\pi(s) P(s'|s,a)$ and the analogous one for $\hat{P}_{t+1}^\pi$, adding and subtracting $\pi(a|s) P_t^\pi(s) \hat{P}(s'|s,a)$ and applying the triangle inequality yields

$$\|P_{t+1}^\pi - \hat{P}_{t+1}^\pi\|_1 \le \|P_t^\pi - \hat{P}_t^\pi\|_1 + \mathbb{E}_{s\sim P_t^\pi,\, a\sim\pi}\left[\|P(\cdot|s,a) - \hat{P}(\cdot|s,a)\|_1\right]. \tag{45}$$

By induction on t with $P_0^\pi = \hat{P}_0^\pi = p_0$,

$$\|P_t^\pi - \hat{P}_t^\pi\|_1 \le \sum_{k=0}^{t-1} \mathbb{E}_{s\sim P_k^\pi,\, a\sim\pi}\left[\|P(\cdot|s,a) - \hat{P}(\cdot|s,a)\|_1\right]. \tag{46}$$

Combining with $\|\hat{\rho}_\pi - \rho_\pi\|_1 \le (1-\gamma)\sum_{t=0}^\infty \gamma^t \|P_t^\pi - \hat{P}_t^\pi\|_1$ and exchanging the order of summation,

$$\begin{aligned} \|\hat{\rho}_\pi - \rho_\pi\|_1 &\le (1-\gamma)\sum_{t=0}^\infty \gamma^t \sum_{k=0}^{t-1} \mathbb{E}_{P_k^\pi,\pi}\left[\|P - \hat{P}\|_1\right] \\ &= (1-\gamma)\sum_{k=0}^\infty \mathbb{E}_{P_k^\pi,\pi}\left[\|P - \hat{P}\|_1\right] \sum_{t=k+1}^\infty \gamma^t \\ &= \gamma \sum_{k=0}^\infty \gamma^k \mathbb{E}_{P_k^\pi,\pi}\left[\|P - \hat{P}\|_1\right] = \frac{\gamma}{1-\gamma}\,\mathbb{E}_{\rho_\pi}\left[\|P(\cdot|s,a) - \hat{P}(\cdot|s,a)\|_1\right]. \end{aligned} \tag{47}$$

Combining Eq. 44 and Eq. 47,

$$|E_3^{(a)}| \le \frac{\gamma C_\psi}{1-\gamma}\,\mathbb{E}_{\rho_\pi}\left[\|P(\cdot|s,a) - \hat{P}(\cdot|s,a)\|_1\right]. \tag{48}$$

For $E_3^{(b)}$, by the L_ψ-Lipschitz property of ζ on the reward range and the triangle inequality,

$$\begin{aligned} |E_3^{(b)}| &\le L_\psi\, \mathbb{E}_{\rho_\pi}\left[\left|\hat{\mathcal{T}}^\pi Q - U - \mathcal{T}^\pi Q\right|\right] \\ &= L_\psi\, \mathbb{E}_{\rho_\pi}\left[\left|\gamma \sum_{s'} (P(s'|s,a) - \hat{P}(s'|s,a)) V^\pi(s') - U(s,a)\right|\right] \\ &\le \gamma L_\psi C_V\, \mathbb{E}_{\rho_\pi}\left[\|P(\cdot|s,a) - \hat{P}(\cdot|s,a)\|_1\right] + L_\psi\, \mathbb{E}_{\rho_\pi}[|U(s,a)|], \end{aligned} \tag{49}$$

where the second equality uses $\hat{\mathcal{T}}^\pi Q - \mathcal{T}^\pi Q = \gamma \sum_{s'} (P - \hat{P}) V^\pi(s')$ and the last inequality uses $|V^\pi| \le C_V$.

Combining all terms. Summing Eq. 40, Eq. 41, Eq. 48 and Eq. 49,

$$\begin{aligned} |\Delta\mathcal{J}| \le &\left[\frac{\gamma(\alpha C_H + C_\psi)}{1-\gamma} + \gamma L_\psi C_V\right] \mathbb{E}_{\rho_\pi}\left[\|P(\cdot|s,a) - \hat{P}(\cdot|s,a)\|_1\right] \\ &+ \mathbb{E}_{\hat{\rho}_\pi}[|U(s,a)|] + L_\psi\, \mathbb{E}_{\rho_\pi}[|U(s,a)|]. \end{aligned} \tag{50}$$

The proof is completed. □

C.4 Proof of Lemma 3.5

Proof. Define the per-step squared score-estimation error at noise level k as

$$\varepsilon_k^2(s,a) := \mathbb{E}_{s'_k \sim p(\cdot|s,a,k)}\Big[\|\varsigma_\phi(s'_k, s, a, k) - \nabla \log p(s'_k|s,a,k)\|_2^2\Big]. \tag{51}$$

Expanding the expectation over the discrete uniform distribution $k \sim \mathrm{Unif}\{0, 1, \ldots, K\}$, we obtain $U_\phi(s,a) = \frac{1}{K+1}\sum_{k=0}^{K} \lambda(k)\, \varepsilon_k^2(s,a)$. Assuming $\lambda(k) \geq \lambda_{\min} > 0$, the worst-step score error $\varepsilon_{\text{score}}(s,a) := \max_k \varepsilon_k(s,a)$ satisfies

$$\varepsilon_{\text{score}}(s,a) \;\leq\; \sqrt{\tfrac{K+1}{\lambda_{\min}} \cdot U_\phi(s,a)}. \tag{52}$$

For fixed (s,a), the reverse ODE in Eq. 15 with score network $\varsigma_\phi(\cdot, s, a, \cdot)$ outputs exactly $\hat{P}_\phi(\cdot|s,a)$. Identifying the target distribution in [45] with $P(\cdot|s,a)$ and applying Theorem 2 therein,

$$\mathsf{TV}\Big(\hat{P}_\phi(\cdot|s,a),\, P(\cdot|s,a)\Big) \lesssim \sqrt{\mathsf{KL}(P(\cdot|s,a)\,\|\,\mathcal{N})}\, e^{-T} + \big(\mathcal{L}\sqrt{dh} + \mathcal{L}\mathfrak{m}_2 h\big)\sqrt{T} + \varepsilon_{\text{score}}(s,a)\sqrt{T}, \tag{53}$$

where $\mathcal{N}$ is the standard Gaussian on $\mathbb{R}^d$, T the diffusion horizon, h the step size, $\mathcal{L}$ the Lipschitz constant of the score, and $\mathfrak{m}_2$ the second moment of $P(\cdot|s,a)$. Using $\|P(\cdot|s,a) - \hat{P}(\cdot|s,a)\|_1 = 2\,\mathsf{TV}(P(\cdot|s,a), \hat{P}(\cdot|s,a))$ and substituting Eq. 52 into Eq. 53,

$$\|P(\cdot|s,a) - \hat{P}_\phi(\cdot|s,a)\|_1 \;\lesssim\; C_1 \cdot \sqrt{U_\phi(s,a)} + C_2, \tag{54}$$

with

$$C_1 := \sqrt{\tfrac{(K+1)\,T}{\lambda_{\min}}}, \quad C_2 := \sup_{(s,a)\in\mathcal{D}} \sqrt{\mathsf{KL}(P(\cdot|s,a)\,\|\,\mathcal{N})}\, e^{-T} + \big(\mathcal{L}\sqrt{dh} + \mathcal{L}\mathfrak{m}_2 h\big)\sqrt{T}. \tag{55}$$

Both C_1 and C_2 are determined by $\boldsymbol{\mu}$, σ and K, independent of ς_ϕ.

The proof is completed. □

C.5 Proof of Corollary 3.6

Proof. We first convert the $\mathbb{E}_{\hat{\rho}_\pi}[|U|]$ term in Theorem 3.4 to one expressed under ρ_π. Adding and subtracting $\mathbb{E}_{\rho_\pi}[|U(s,a)|]$, applying Hölder's inequality together with the simulation lemma (Eq. 47), and using $|U(s,a)| \leq C_U$ from Assumption 3.3,

$$\begin{aligned}\mathbb{E}_{\hat{\rho}_\pi}[|U(s,a)|] &\leq \mathbb{E}_{\rho_\pi}[|U(s,a)|] + C_U\, \|\hat{\rho}_\pi - \rho_\pi\|_1 \\ &\leq \mathbb{E}_{\rho_\pi}[|U(s,a)|] + \frac{\gamma C_U}{1-\gamma}\, \mathbb{E}_{\rho_\pi}\Big[\|P(\cdot|s,a) - \hat{P}(\cdot|s,a)\|_1\Big].\end{aligned} \tag{56}$$

Substituting Eq. 56 into Theorem 3.4,

$$\begin{aligned}|\Delta\mathcal{J}| \leq &\left[\frac{\gamma(\alpha C_H + C_\psi + C_U)}{1-\gamma} + \gamma L_\psi C_V\right] \mathbb{E}_{\rho_\pi}\Big[\|P(\cdot|s,a) - \hat{P}(\cdot|s,a)\|_1\Big] \\ &+ (1 + L_\psi)\mathbb{E}_{\rho_\pi}[|U(s,a)|].\end{aligned} \tag{57}$$

Next, taking the expectation of Eq. 54 under ρ_π and applying Jensen's inequality $\mathbb{E}[\sqrt{U_\phi}] \leq \sqrt{\mathbb{E}[U_\phi]}$,

$$\mathbb{E}_{\rho_\pi}\Big[\|P(\cdot|s,a) - \hat{P}_\phi(\cdot|s,a)\|_1\Big] \;\lesssim\; C_1\sqrt{\bar{U}_P} + C_2. \tag{58}$$

Substituting Eq. 58 into Eq. 57 and using $\mathbb{E}_{\rho_\pi}[|U(s,a)|] = -\xi\bar{U}_P$ from $U(s,a) = \xi U_\phi(s,a) \leq 0$,

$$|\Delta\mathcal{J}| \lesssim \left[\frac{\gamma(\alpha C_H + C_\psi + C_U)}{1-\gamma} + \gamma L_\psi C_V\right]\Big(C_1\sqrt{\bar{U}_P} + C_2\Big) - \xi(1 + L_\psi)\bar{U}_P. \tag{59}$$

The proof is completed. □

C.6 Proof of Theorem B.1

Proof. During guided sampling, the policy score is augmented from $\nabla_a \log \pi_b(a|s)$ to $\nabla_a \log \pi_b(a|s) + \kappa \nabla_a g(s,a)$. By the standard equivalence between score-based classifier guidance and exponential tilting [46, 47], the resulting guided policy admits the closed form

$$\pi_g(a|s) = \frac{\pi_b(a|s)\exp(\kappa\, g(s,a))}{Z(s,\kappa)}, \quad Z(s,\kappa) := \mathbb{E}_{a\sim\pi_b}[\exp(\kappa\, g(s,a))]. \tag{60}$$

Let $\mathcal{G}(\kappa) := \log Z(s,\kappa)$ denote the cumulant generating function. By standard properties,

$$\mathcal{G}'(\kappa) = \mathbb{E}_{a\sim\pi_g}[g(s,a)], \qquad \mathcal{G}'(0) = \mathbb{E}_{a\sim\pi_b}[g(s,a)], \qquad \mathcal{G}''(0) = \mathrm{Var}_{a\sim\pi_b}(g(s,a)). \tag{61}$$

A Taylor expansion of $\mathcal{G}'(\kappa)$ at $\kappa = 0$ therefore gives

$$\mathbb{E}_{a\sim\pi_g}[g(s,a)] - \mathbb{E}_{a\sim\pi_b}[g(s,a)] = \kappa \cdot \mathrm{Var}_{a\sim\pi_b}(g(s,a)) + \mathcal{O}(\kappa^2). \tag{62}$$

By the Performance Difference Lemma [48],

$$J(\pi_g) - J(\pi_b) = \frac{1}{1-\gamma}\mathbb{E}_{s\sim d_{\pi_g}}\left[\mathbb{E}_{a\sim\pi_g}[Q^{\pi_b}(s,a)] - \mathbb{E}_{a\sim\pi_b}[Q^{\pi_b}(s,a)]\right]. \tag{63}$$

Decomposing $Q^{\pi_b} = g + (Q^{\pi_b} - g)$ and using $\|g - Q^{\pi_b}\|_\infty \le \varepsilon_p$,

$$\mathbb{E}_{a\sim\pi_g}[Q^{\pi_b}(s,a)] - \mathbb{E}_{a\sim\pi_b}[Q^{\pi_b}(s,a)] \ge \left(\mathbb{E}_{a\sim\pi_g}[g] - \mathbb{E}_{a\sim\pi_b}[g]\right) - 2\varepsilon_p. \tag{64}$$

Substituting Eq. 62 into Eq. 64 and plugging back into Eq. 63 yields

$$J(\pi_g) - J(\pi_b) \ge \frac{1}{1-\gamma}\mathbb{E}_{s\sim d_{\pi_g}}\left[\kappa \cdot \mathrm{Var}_{a\sim\pi_b}(g(s,a)) - \mathcal{O}(\kappa^2) - 2\varepsilon_p\right]. \tag{65}$$

The proof is completed. □

D Experimental details.

D.1 Benchmark

MuJoCo. MuJoCo gym environments are widely utilized in RL as a standard benchmark for evaluating algorithm performance, including various robotic continuous control tasks such as Hopper, HalfCheetah, Walker2d, and Ant.

- Hopper: This environment simulates a 2D single-legged robot whose objective is to achieve continuous forward hopping, where the primary challenge lies in maintaining balance during movement.
- HalfCheetah: This environment simulates a 2D cheetah robot whose objective is to make the cheetah run forward as fast as possible, where the primary challenge lies in complex coordination of full-body joints.
- Walker2d: This environment simulates a 2D bipedal robot whose objective is to walk forward as fast as possible, where the primary challenge lies in the coordinated control of two legs.
- Ant: This environment simulates a 3D quadrupedal ant robot whose objective is to move forward as fast as possible, where the primary challenge lies in the coordination of smooth movement across four legs.

The maximum episode length for all these environments is set to 1000 steps, as shown in Figure 8.

D4RL. D4RL is an open-source benchmark for Offline RL, providing standardized environments and datasets for training and evaluating algorithms. Each task in its Mujoco dataset includes five types of data.

- expert: Train a SAC algorithm online until the policy reaches the expert-level performance, then use the expert policy to collect 1 million samples of data.
- medium: Train a SAC algorithm online, stop training in the middle, and then use this partially trained policy to collect 1 million samples of data.

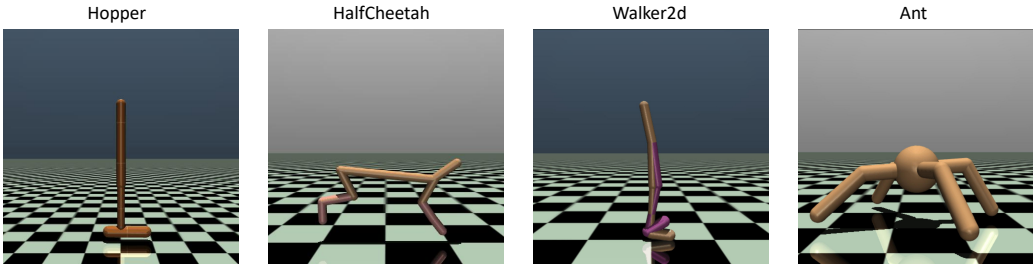


Figure 8: Four different MuJoCo environments.

- medium-expert: Mix equal amounts of data collected by expert and medium policies, approximately 2 million samples of data.
- medium-replay: Train a SAC algorithm online until the policy reaches a moderate performance level, and then collect all the samples placed in the buffer during training, approximately 1 million samples of data.
- random: Use a random initialization policy to collect 1 million samples of data.

These datasets with diverse difficulty levels are well-suited to our experimental setup. Therefore, we adopt the D4RL benchmark as our evaluation tool.

D.2 Hyperparameters settings and experimental implementations

- The diffusion dynamics is implemented with a three-layer MLP, with hidden layer dimensions of 256 and diffusion steps of 50.
- The training of our diffusion policy is built upon the open-source framework of the MaxEnt DP algorithm, available at `https://github.com/diffusionyes/MaxEntDP`.
- Our implementation of CLARE, OMLIRL, SPRINQL, IQL, and DWBC follows their official codebases, available at `https://github.com/shaunyue/clare`, `https://github.com/Cloud0723/Offline-MLIRL`, `https://github.com/hmhuy0/SPRINQL`, `https://github.com/Div99/IQ-Learn`, and `https://github.com/ryanxhr/DWBC`, respectively.
- The scale of D4RL data used in our experiments is comparable to that in existing Offline IRL baseline evaluations.
- DPBC is a simplified implementation of our algorithm.
- Our code is provided in the supplementary material.
- All experiments in this paper are conducted on the GPU of Nvidia GeForce RTX 4090 and the CPU of Intel Xeon 8368.
- The shared hyperparameters of all algorithms are summarized in Table 3.

Table 3: The shared hyperparameters of all algorithms.

Hyperparameter	OIDP	CLARE	OMLIRL	SPRINQL	IQL	DWBC	DPBC
Batch size	256	256	256	256	256	256	256
Discount factor	0.99	0.99	0.99	0.99	0.99	-	-
Soft update coefficient	0.005	0.005	0.005	0.005	0.005	-	-
Critic hidden size	256	256	256	256	256	-	-
Actor hidden size	256	256	256	256	256	256	256
Critic learning rate	3e-4	3e-4	3e-4	3e-4	3e-4	-	-
Actor learning rate	3e-4	3e-4	3e-4	3e-4	3e-4	3e-4	3e-4
Temperature learning rate	3e-4	3e-4	3e-4	-	3e-4	-	-
Optimizer	Adam	Adam	Adam	Adam	Adam	Adam	Adam
Activation	Mish	Swish	ReLU	ReLU	ReLU	ReLU	Mish

E Supplementary experiments.

E.1 Comparative evaluation on Adroit benchmark

The Adroit benchmark involves controlling a 24-DoF robotic hand. We evaluate various offline IRL algorithms on four tasks, as shown in Figure 9. Expert dataset consists of 5,000 trajectories sampled from an expert policy that solves the task. Cloned dataset is constructed by mixing human demonstration data with 2,500 trajectories generated by a behavior cloning policy trained on those demonstrations, and is treated as a non-expert dataset. Table 4 reports the performance comparison between OIDP and baseline offline IRL algorithms. On these more challenging tasks, which are also closer to real-world robotic manipulation scenarios, OIDP outperforms or remains competitive with the other baselines.

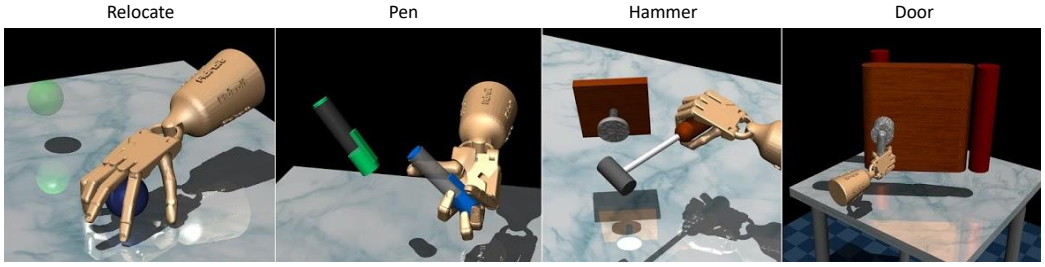


Figure 9: Adroit environments.

Table 4: Results on Adroit tasks. For each task, the experiment used 5k expert and 20k cloned tuples. All tasks are trained for 200k steps. The highest score across all algorithms is bold.

Dataset	Env	OIDP	CLARE	OMLIRL	SPRINQL	IQL
Exp. & Cloned.	Hammer	**61.5±15.1**	57.9±22.6	-1.8±0.7	19.7±23.8	-1.9±0.1
	Door	**67.9±11.6**	0.3±3.4	-2.1±0.1	17.4±17.8	-2.3±0.0
	Pen	82.9±15.7	**98.0±11.7**	1.3±1.8	52.6±13.6	-0.9±0.2
	Relocate	**22.2±8.9**	4.0±4.3	-0.3±0.1	0.6±1.2	-0.5±0.0

E.2 Reduced expert data

To analyze the sensitivity of algorithms to the amount of expert data, we reduce the expert data to 1k and present the performance comparison of different algorithms in Table 5. The results show that the proposed OIDP approach is insensitive to the number of expert demonstrations. Despite the limited expert data, OIDP still outperforms the baseline algorithms on most tasks. Notably, by comparing with Table 1, it can be observed that BC approaches are highly sensitive to expert data quantity, which aligns with our introduction.

Table 5: Results on MuJoCo tasks. For each task, the experiments are carried out with two different data combinations: 1) 1k expert and 20k medium tuples, 2) 1k expert and 20k random tuples. Except for Hopper, which is trained for 100k steps, all other tasks are trained for 200k steps. The highest score across all algorithms is bold.

Dataset	Env	OIDP	CLARE	OMLIRL	SPRINQL	IQL	DWBC	DPBC
Exp. & Med.	Hopper	**94.3±6.0**	47.6±2.6	-0.2±0.0	40.1±17.3	5.7±4.4	19.0±6.5	31.7±12.7
	HalfCheetah	**49.0±4.7**	46.0±1.5	21.8±10.6	42.0±6.0	-1.6±0.9	6.7±3.7	1.8±0.5
	Walker2d	**66.6±10.2**	54.5±11.6	3.4±4.6	58.3±22.5	0.0±0.2	42.6±17.2	40.9±7.7
Exp. & Rand.	Hopper	**71.5±8.8**	4.4±3.0	11.5±13.6	9.4±6.7	4.0±2.7	12.9±3.2	31.7±12.7
	HalfCheetah	**43.6±2.2**	2.0±0.7	-1.2±3.6	5.7±1.7	-1.2±0.5	5.9±2.4	1.8±0.5
	Walker2d	40.4±5.2	2.0±1.2	-0.1±0.0	25.6±12.1	0.1±0.1	**46.3±9.4**	40.9±7.7

E.3 Evaluation of the uncertainty estimator

To examine the sensitivity of the uncertainty estimator to data coverage (related to epistemic uncertainty) and transition stochasticity (related to aleatoric uncertainty), we construct three controlled settings using HalfCheetah data: Dense–Deterministic (DD), Dense–Stochastic (DS), and Sparse–Stochastic (SS). Table 6 reports the 20-nearest-neighbor distance (reflecting data coverage), aleatoric variance (reflecting transition stochasticity), the uncertainty proxy $\widehat{U}_\phi$ computed on rollout data using Eq. 20, the DSM residual $\widetilde{U}_\phi$ computed on real transition data using Eq. 19, and the dynamics error E on held-out data.

The results show that, at the same coverage level, increasing transition stochasticity (with aleatoric variance rising from 6.2×10^{-4} to 9.1×10^{-3}) only slightly increases $\widehat{U}_\phi$ from 0.014 to 0.017, which remains relatively low. In contrast, at the same level of transition stochasticity, reducing coverage (with the 20-NN distance increasing from 1.1 to 1.8) produces a larger increase in $\widehat{U}_\phi$, from 0.017 to 0.026. $\widetilde{U}_\phi$ exhibits the same qualitative ordering. Moreover, $\widehat{U}_\phi$ remains strongly positively correlated with the dynamics error on held-out data E. These results support using $\widehat{U}_\phi$ as an empirical uncertainty estimator that reflects model reliability while mitigating sensitivity to aleatoric uncertainty.

Table 6: Controlled evaluation of the uncertainty estimator in HalfCheetah.

Setting	20-NN distance	Aleatoric variance	$\widehat{U}_\phi$	$\widetilde{U}_\phi$	E	Spearman $(\widehat{U}_\phi, E)$
DD	1.1 ± 0.01	$(6.2 \pm 1.4) \times 10^{-4}$	0.014 ± 0.0025	0.023 ± 0.0011	0.17 ± 0.04	0.78 ± 0.02
DS	1.1 ± 0.01	$(9.1 \pm 2.1) \times 10^{-3}$	0.017 ± 0.0020	0.027 ± 0.0010	0.19 ± 0.06	0.75 ± 0.03
SS	1.8 ± 0.01	$(9.1 \pm 2.1) \times 10^{-3}$	0.026 ± 0.0062	0.042 ± 0.0043	0.30 ± 0.08	0.72 ± 0.02

E.4 Hyperparameter analysis and ablation

In this subsection, we conduct ablation studies on the hyperparameters and analyze the effect of different settings on performance.

- **Reward regularization coefficient** β. The purpose of reward regularization is to incorporate prior or constraints to make the reward function more reasonable, preventing overfitting, and ensuring learning stability. The performance of different reward regularization coefficients is shown in the Figure 10. Since the Hopper environment is relatively simple, it is not sensitive to this parameter. For other environments, a small coefficient provides weak constraints on reward, often leading to instability, while a large coefficient over-smooths reward, reducing discrimination capability between expert and non-expert behaviors. Therefore, after weighing the performance trade-offs, we uniformly set it to 0.5.
- **Penalty coefficient** $-\xi$. In model-based methods, the penalty term mitigates the risk of extrapolation errors and induces a conservative Q estimation. The performance of different penalty coefficients is shown in Figure 11. Due to the relatively simple environmental dynamics of Hopper, which is easily represented by models, its performance remains insensitive to changes in penalty coefficient. As the task dimension increases, more complex environments require a larger penalty, such as Ant. However, the larger coefficient is not always better. According to Corollary 3.6, an excessively large penalty coefficient causes additional bias. The settings of this parameter in different environments are shown in Table 8.
- **Target entropy** $\bar{H}$**.** The target entropy is set to satisfy Assumption 3.3 while regulating the policy's uncertainty level. We search for it for different tasks and the performance is shown in Figure 12. The results indicate that in offline settings, increasing entropy to encourage exploration does not bring additional performance improvement. Instead, according to Theorem 3.4, it amplifies model mismatch errors, leading to a larger objective gap.
- **Planning horizon** W**.** The planning horizon length W determines the strength of diffusion guidance. Table 7 demonstrates that a smaller W provides limited long-horizon guidance,

while a larger W introduces additional computational cost and accumulated bias. Therefore, W should be chosen as a reasonable trade-off. In our experiments, we set $W = 5$ by default.

Table 7: Results of different planning horizon settings on MuJoCo tasks.

Dataset	Env	W=0	W=2	W=5	W=7	W=10
Exp. & Medrep.	Hopper	93.9±8.4	96.0±6.7	99.2±4.6	**99.4±4.2**	95.0±7.7
	HalfCheetah	60.6±6.2	61.8±5.9	66.4±6.5	64.4±6.9	**67.5±5.4**

Table 8: The hyperparameters of each benchmark.

Hyperparameter	Hopper	HalfCheetah	Walker2d	Ant
Reward regularization coefficient	0.5	0.5	0.5	0.5
Penalty coefficient	1	1	1	5
Target entropy	0.2	0.2	0.2	2

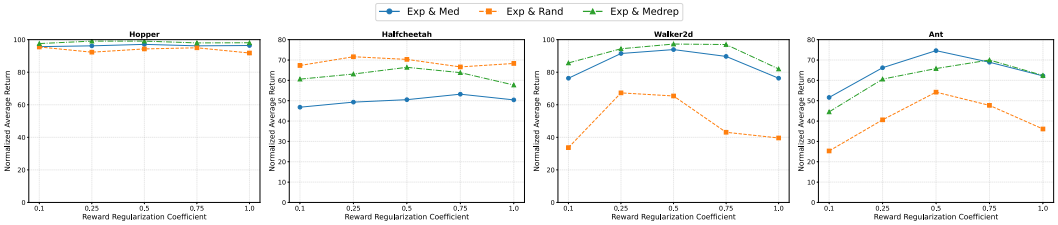


Figure 10: Results of different reward regularization coefficients on MuJoCo tasks.

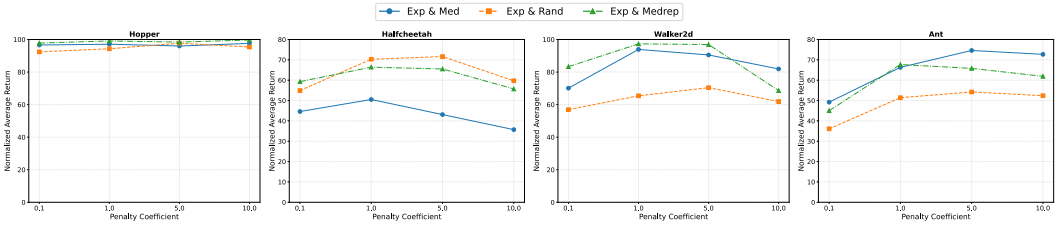


Figure 11: Results of different penalty coefficients on MuJoCo tasks.

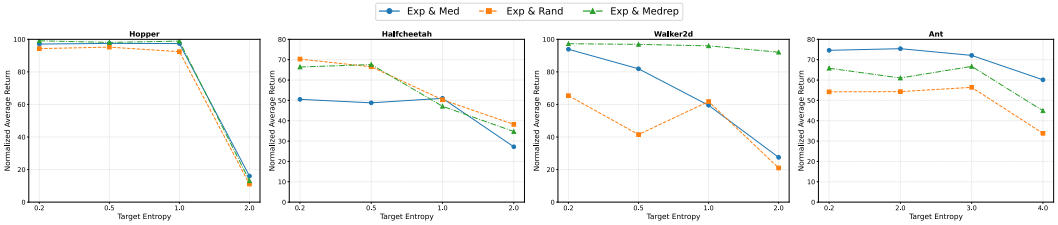


Figure 12: Results of different target entropy settings on MuJoCo tasks.

E.5 Computational cost

To quantify training and inference costs, we evaluate all methods on HalfCheetah under the same experimental setup and hardware. The results are shown in Table 9.

With a parallel implementation using JAX and GPU execution, OIDP maintains an acceptable training cost relative to the baselines. Although trajectory guidance increases inference latency, this module is optional and can be enabled according to task requirements to balance performance and efficiency.

Table 9: The comparison of computational cost on HalfCheetah.

Metric	OIDP	CLARE	OMLIRL	SPRINQL	IQL	DWBC
Model training time (h)	0.25	0.12	4.39	–	–	–
Agent training time (h)	3.02	1.06	1.53	2.00	1.14	0.70
Peak GPU memory (MiB)	3849	1066	1216	1110	876	874
Per-step inference latency (ms)	1.60 ($W = 0$) 105.34 ($W = 5$)	0.83	0.72	0.59	0.57	0.81

F Broader impacts.

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none of which we feel must be specifically highlighted here.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The main claims made in the abstract and introduction accurately reflect the paper's contributions and scope.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: The paper discusses the limitations of the work performed by the authors. Please see Sec. 5.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

Answer: [Yes]

Justification: For each theoretical result, the paper provides the full set of assumptions and a complete (and correct) proof. Please see App. C.

Guidelines:

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

4. **Experimental result reproducibility**

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

Answer: [Yes]

Justification: The paper fully discloses all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not). Please see Sec. 4 and App. D.

Guidelines:

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

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

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

   Answer: [Yes]

   Justification: The paper provides open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results. Please see the supplementary material we have added.

   Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: The paper specifies all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results. Please see Sec. 4 and App. E.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: The paper reports error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments. Please see Sec. 4.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

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

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: The paper provides sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments. Please see App. D.2.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: The research conducted in the paper conforms, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: The paper discusses both potential positive societal impacts and negative societal impacts of the work performed. Please see App. F.

Guidelines:

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

11. **Safeguards**

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

    Answer: [N/A]

    Justification: Our work does not release data or models, so the paper has no such risks.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: The creators or original owners of assets (e.g., code, data, models), used in the paper, are properly credited and are the license and terms of use explicitly mentioned and properly respected. Please see the supplementary material we have added.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.

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

13. **New assets**

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

    Answer: [N/A]

    Justification: The paper does not release new assets.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

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

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

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

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

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

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

    Answer: [N/A]

    Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

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