Learning from Sparse-Reward Offline Datasets via Preference-based Policy Optimization

Wenjie Qiu
Rutgers University
wq37@cs.rutgers.edu

Guofeng Cui
Rutgers University
gc669@cs.rutgers.edu

Shicheng Liu
Pennsylvania State University
sfl5539@psu.edu

Yuanlin Duan
Rutgers University
yuanlin.duan@rutgers.edu

He Zhu
Rutgers University
hz375@cs.rutgers.edu

Abstract

Offline reinforcement learning (RL) holds the promise of training effective policies from static datasets without the need for costly online interactions. However, offline RL faces key limitations, most notably the challenge of generalizing to unseen or infrequently encountered state-action pairs. When a value function is learned from limited data in sparse-reward environments, it can become overly optimistic about parts of the space that are poorly represented, leading to unreliable value estimates and degraded policy quality. To address these challenges, we introduce a novel approach based on contrastive preference learning that bypasses direct value function estimation. Our method trains policies by contrasting successful demonstrations with failure behaviors present in the dataset, as well as synthetic behaviors generated outside the support of the dataset distribution. This contrastive formulation mitigates overestimation bias and improves robustness in offline learning. Empirical results on challenging sparse-reward offline RL benchmarks show that our method substantially outperforms existing state-of-the-art baselines in both learning efficiency and final performance.

1 Introduction

Offline reinforcement learning (RL) aims to learn high-quality decision policies purely from static datasets, without requiring additional environment interactions. This paradigm offers a compelling route for deploying RL in real-world domains where data collection is costly, risky, or constrained—such as robotics, healthcare, or recommendation systems. However, offline RL remains fundamentally challenging due to the distributional mismatch between the policy being learned and the limited data it learns from. A core issue lies in the extrapolation error that arises when learned value functions are queried on state-action pairs not well represented in the dataset. This is particularly problematic in sparse-reward settings [1], where the dataset may lack sufficient reward-bearing trajectories or behavioral diversity. As a result, value-based methods can become overly optimistic in poorly covered regions of the state-action space, leading to unstable or suboptimal policies.

To address this, prior work has largely focused on three classes of solutions. First, pessimism-based approaches mitigate overestimation by explicitly penalizing uncertain or unsupported regions in the learned value function. Techniques such as conservative Q-learning [2] or uncertainty-aware backups enforce value suppression on out-of-distribution actions. However, these methods often rely on assumptions about the behavior policy and require careful calibration of the degree of pessimism, which becomes increasingly difficult in high-dimensional or sparse settings. Second, regularization-based methods constrain policy updates to remain close to the behavior policy by adding policy

Preprint. Under review.

divergence penalties [3, 4, 5]. While effective in well-covered datasets, these methods can be brittle when tuning the regularization strength and may fail to explore beyond suboptimal behaviors. Third, importance sampling-based techniques, including DICE-style distribution correction [6, 7], attempt to re-weight observed rewards based on estimated marginal state-action densities. Although theoretically sound and behavior-agnostic, these methods are sensitive to support mismatches and can suffer from high variance or instability, especially when the data are limited or reward signals are sparse.

In this paper, we propose a fundamentally different approach that avoids direct value function estimation altogether. We introduce, **PREFORL** (PREFerence-based Optimization for Offline RL), a contrastive preference learning framework that learns policies by directly comparing successful and unsuccessful behaviors in sparse-reward offline datasets. Crucially, we extend the contrastive signal beyond failure behaviors present in the dataset to include synthetic behaviors generated outside the dataset's support. By contrasting both types of behaviors against successful demonstrations, our method trains policies to imitate not just what succeeds, but to actively avoid what likely fails or lies outside the data support. This formulation enables us to sidestep the estimation pitfalls of value-based methods while directly combating overestimation. Our empirical evaluation on challenging sparse-reward offline RL benchmarks shows that this contrastive approach leads to more stable learning and substantially outperforms existing state-of-the-art offline RL baselines.

2 Problem Formulation and Motivations

We formulate the reinforcement learning problem in the context of a Markov Decision Process (MDP) $M = \langle \mathcal{S}, \mathcal{A}, T, r, \gamma, \rho_0 \rangle$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $T : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ is the transition probability function $T(s' \mid s, a)$, $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward function, $\gamma \in (0, 1)$ is the discount factor, and $\rho_0 : \mathcal{S} \to [0, 1]$ is the initial state distribution. A policy $\pi : \mathcal{S} \times \mathcal{A} \to [0, 1]$ maps each state to a distribution over actions. Let $\tau = \{s_0, a_0, s_1, a_1, \ldots\}$ denote a trajectory sampled by interacting with the MDP under policy π, i.e., $s_0 \sim \rho_0$, $a_t \sim \pi(\cdot \mid s_t)$, $s_{t+1} \sim T(\cdot \mid s_t, a_t)$. Then, the discounted state-action distribution induced by π is defined as $d^\pi(s, a) = (1 - \gamma) \sum_{t=0}^{\infty} \gamma^t \mathbb{E}_{\tau \sim \pi} [\mathbb{1}[s_t = s, a_t = a]]$. The goal is to learn a policy $\pi_\theta(a|s)$ that maximizes the expected discounted return: $\mathbb{E}_{s_0, a_0, s1, \ldots \sim d^{\pi_\theta}} [\Sigma_0^\infty \gamma^t r(s_t, a_t)]$. In Offline RL, the agent does not have access to the environment M, and instead must learn a policy solely from a static dataset: $\mathcal{D} = \{(s_i, a_i, r_i, s'_i)\}_{i=1}^N$ collected from some (possibly unknown) behavior policy π_β. The empirical state-action distribution of the dataset is denoted $d^{\mathcal{D}}(s, a)$, which approximates $d^{\pi_\beta}(s, a)$. We consider the challenging setting of *sparse reward offline RL*, where informative reward signals are infrequent and the dataset $\mathcal{D}$ predominantly consists of transitions with zero or low rewards, making it difficult to identify and generalize from successful behaviors. We define $d^*(s)$ as the optimal state marginal, which can be viewed as a state distribution of successful trajectories in the dataset $\mathcal{D}$.

The Advantage Preference Model. In Direct Preference Optimization (DPO) [8], a Bradley-Terry (BT) [9] model is built on top of the hidden reward model r_E given by expert users to capture the preferences of pairs of answers $(y_1, y_2) \sim \pi_\theta(y|x)$. While DPO and other reinforcement learning from human feedback (RLHF) algorithms [10] have shown strong performance for large language models (LLMs)—which can be framed as contextual bandit problems—they are not directly suited for general RL tasks where trajectory-level preferences are crucial for solving long-horizon problems. To that end, we define a trajectory of length n as $\tau = (s_0, a_0, \ldots, s_{n-1}, a_{n-1})$, and introduce the notion of a length-k representative segment, denoted by $\sigma = \Sigma(\tau, k) = (\hat{s}_0, \hat{a}_0, \ldots, \hat{s}_{k-1}, \hat{a}_{k-1})$, which approximates the overall quality and semantics of its original trajectory $\tau = \mathcal{T}(\sigma)$. Each $(\hat{s}_t, \hat{a}_t)$ in the segment is sampled from τ, with the constraint that their original indices $\mathbb{I}_\tau(t)$ are strictly increasing to preserve temporal order. We denote a segment-level preferences as $\sigma^+ > \sigma^-$, which we assume it reflects overall preference for their corresponding full trajectories, i.e., $\mathcal{T}(\sigma^+) > \mathcal{T}(\sigma^-)$. Recent work such as Knox et al. [11] estimates such preferences by comparing partial discounted returns $\sum_t^k \gamma^t r(s_t, a_t)$ for trajectory segments. However, in settings with sparse or highly imbalanced rewards, this return-based signal may be too weak or misleading to support reliable comparisons. To mitigate this, we instead adopt an advantage-based preference model in Contrastive Preference Learning (CPL) [12], which focuses on distinguishing successful behaviors not just based on returns, but through their relative quality under advantage estimation:

$$P_{A^*}[\tau^+ > \tau^-] = P_{A^*}[\sigma^+ > \sigma^-] = \frac{\exp \sum_{\sigma^+} \gamma^t A^*(\hat{s}_t^+, \hat{a}_t^+)}{\exp \sum_{\sigma^+} \gamma^t A^*(\hat{s}_t^+, \hat{a}_t^+) + \exp \sum_{\sigma^-} \gamma^t A^*(\hat{s}_t^-, \hat{a}_t^-)}, \quad (1)$$

where A^* denotes the optimal advantage function, and $\tau^+ = \mathcal{T}(\sigma^+)$ and $\tau^- = \mathcal{T}(\sigma^-)$ are two complete trajectories. We use the shorthand "+" and "-" to denote the preferred / less preferred representative segments.

Contrastive Preference Learning. CPL [12] eliminates the hidden optimal advantage function A^* in the advantage-based preference model in the context of maximum entropy RL [13, 14, 15]. The derivation is straightforward, as Ziebart [13] provides a critical insight, i.e., the optimal advantage function A^* and optimal policy $\pi^*(a|s)$ has a direct relationship:

$$A^*(s,a) = \alpha \log \pi^*(a|s). \tag{2}$$

assuming that the optimal advantage function is normalized $\int e^{A^*(s,a)/\alpha}\, da = 1$. This means that instead of learning an implicit optimal advantage function, CPL can leverage the preference model to acquire the optimal policy directly. Given an offline preference dataset $\mathcal{D}_{\text{pref}}$, the learning objective is to minimize the following loss function while increasing the likelihood of actions in the datasets.

$$\mathcal{L}_{\text{CPL}}(\pi_\theta, \mathcal{D}_{\text{pref}}) = \mathbb{E}_{(\varsigma^+,\varsigma^-)\sim\mathcal{D}_{\text{pref}}}[-\log \frac{\exp \sum_{\varsigma^+} \gamma^t \alpha \log \pi_\theta(s_t^+, a_t^+)}{\exp \sum_{\varsigma^+} \gamma^t \alpha \log \pi_\theta(s_t^+, a_t^+) + \exp \lambda \sum_{\varsigma^-} \gamma^t \alpha \log \pi_\theta(s_t^-, a_t^-)}], \tag{3}$$

where $\lambda \in (0, 1]$ denotes the asymmetric "bias" regularizer [16] that down-weights the negative segments. Intuitively, this conservative regularizer let loss the function produce lower loss when the policy has higher likelihood on actions in $\mathcal{D}_{\text{pref}}$, thus encourage π_θ to be in-distribution.

Practical Considerations. In Equation 3, ς^+ and ς^- denote the positive and negative segments sampled from offline preference datasets $\mathcal{D}_{\text{pref}}$. However, in practice, explicit preference labels in $\mathcal{D}_{\text{pref}}$ may be **unavailable**. We may acquire ς^+ and ς^- which are *consecutive* segments with identical lengths that cropped from different trajectories. In order to construct CPL loss, we must build preference labels explicitly based on discounted return if dense rewards are available in the dataset, or let human experts involve [17]. Yet, the reward function can be sparse in some RL environments, which poses challenges in constructing reliable preference labels. Hejna et al. [12] present a work-around solution to demonstrate the feasibility and effectiveness of CPL algorithm in the MetaWorld [18] benchmark. They first train an oracle policy using online RL algorithms, e.g., PPO [19], and then use the trained critic function to evaluate the discounted return of sampled segments to generate binary preference labels. Experiments show that CPL is effective in post-training of an RL policy in an offline manner.

3 Preference-based Policy Optimization

In Section 2, we discussed the practical considerations of building contrastive preference loss. In this section, we bridge the gap between proof-of-concept to a practical offline RL algorithm called **PREFORL** for sparse-reward offline datasets.

3.1 Trajectory Degradation

We propose a contrastive preference learning framework to learn effective policies through direct comparison between successful and unsuccessful behaviors. The key idea is to incorporate synthetically generated suboptimal trajectories out of the support of the dataset. By training policies to prefer successful demonstrations over both observed and synthetic failure cases, the framework encourages imitation of high-quality behavior while simultaneously improving robustness against failure modes and distributional drift.

When presented with a successful trajectory $\tau = \tau^+$ in a sparse-reward offline dataset $\mathcal{D}$ and the request to build its corresponding less preferred trajectory τ^-, a naive approach is to decompose the problem from trajectory level to state-action level. To this end, we define $q(s,a)$ as the quality of a single state-action pair. Similarly, we define $\mathcal{Q}(\tau) = \sum q(s,a)$ as the quality of an entire trajectory. Note that this is different from state value function $V(s,a)$ or state-action value function $Q(s,a)$ which consider long-term discounted return in the future. Suppose (s,a) is a state-action pair in the given dataset, we argue a is an optimal action given state s. Without loss of generality, we assume:

$$q(s,a^*) = q(s,a) > q(s, \mathcal{F}(s,a)), \tag{4}$$

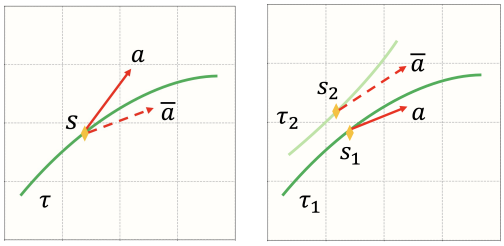


Figure 1: Two variants of $\mathcal{F}(s,a)$. In the left figure, $\mathcal{F}^{\sigma}(s,a)$ degrades the action a to $\bar{a}$ using Gaussian noise. In the right figure, $\mathcal{F}^{\gamma}(s,a)$ degrades the action a in trajectory τ_1 by finding a substitution action $\bar{a}$ correspond a neighbor state in a less preferred trajectory τ_2. Red arrows with solid lines denotes the original actions, and the red arrows with dashed lines denotes degraded actions.

where $\mathcal{F}(s,a)$ is the action degradation function that satisfy $\mathcal{F}(s,a) \neq a$. This assumption is intuitive and necessary, as any change to the optimal action a^* could inevitably degrade its quality. Consequently, when applying such action degradation on all or a subset of actions in a trajectory τ, we have:

$$\mathcal{Q}(\tau) = \sum q(s,a^*) = \mathcal{Q}(\tau^+) > \sum q(s,\mathcal{F}_P(s,a^*)) = \mathcal{Q}(\tau^-), \tag{5}$$

where $\mathcal{F}_P(s,a)$ is a probabilistic action degradation function that degrades with probability P. This inequality demonstrates the feasibility of generating suboptimal trajectories utilizing $\mathcal{F}(s,a)$, we also assume this is valid for their representative segments. In practice, we have two implementations of $\mathcal{F}$ corresponding to two variants action degradation methods. The degradation function $\mathcal{F}^{\sigma}(s,a)$ perturbs actions using Gaussian noise, deliberately pushing them outside the support of the dataset (in Equation 6). The $\mathcal{F}^{\gamma}(s,a)$ function first performs a nearest-neighbor search from state s over the dataset $\mathcal{D}$, constrained by a condition `cond` that selects states from less preferred trajectories—i.e., those with low sparse rewards. It then retrieves the corresponding action at the matched state as the degraded action (as defined in Equation 7).

$$\mathcal{F}^{\sigma}(s,a) = \text{clip}(a+N, a_{min}, a_{max}), N \overset{d}{\sim} \mathcal{N}(0,\sigma^2) \tag{6}$$

$$\mathcal{F}^{\gamma}(s,a) = \texttt{RetrieveAction}(\mathcal{D}, \texttt{NeighborSearch}(\mathcal{D}, s, \texttt{cond})) \tag{7}$$

PREFORL Loss Function. Given contrastive segments, similar to CPL, the PREFORL loss is:

$$\mathcal{L}_{\text{PREFORL}}(\pi_\theta, \mathcal{D}) = \mathbb{E}_{(\sigma^+,\sigma^-)\sim\mathcal{D}}[-\log \frac{\exp \sum_{\sigma^+} \gamma^t \alpha \log \pi_\theta(\hat{s}_t^+, \hat{a}_t^+)}{\exp \sum_{\sigma^+} \gamma^t \alpha \log \pi_\theta(\hat{s}_t^+, \hat{a}_t^+) + \exp \lambda \sum_{\sigma^-} \gamma^t \alpha \log \pi_\theta(\hat{s}_t^-, \hat{a}_t^-)}]. \tag{8}$$

3.2 Algorithm Overview

We present the overview of PREFORL in Algorithm 1. We use $\mathcal{D}^+$ as the set of successful trajectories in a sparse-reward offline dataset $\mathcal{D}$. Given initial policy π_θ, in each iteration, we sample multiple preferred representative segments σ^+ from $\mathcal{D}^+$, and build their corresponding less preferred degraded segments σ^-. At the end of each iteration, we optimize policy π_θ using PREFORL loss function shows in Equation 8. Note that PREFORL is an offline algorithm that does not requires online interaction with the environment.

Theoretical Justification. Define the state marginals of d^D, d^π, and d^* as $d^D(s)$, $d^\pi(s)$, and $d^*(s)$, respectively. The following bound on the performance gap between the learned and optimal policies is established based on the above assumption in [6]:

$$|V^\pi(\rho_0) - V^{\pi^*}(\rho_0)| \leq \frac{2R_{\max}}{1-\gamma} D_{\text{TV}}(d^*(s) \,\|\, d^D(s)) + \frac{2R_{\max}}{1-\gamma} \mathbb{E}_{d^*(s)} \left[D_{\text{TV}}(\pi(\cdot|s) \,\|\, \pi^*(\cdot|s)) \right],$$

where $R_{\max} = \max_{s,a} \|r(s,a)\|$ is the maximum reward. This shows that we can minimize $D_{\text{TV}}(\pi(\cdot|s) \,\|\, \pi^*(\cdot|s))$ to optimize the learned policy π. Let $P_{A^*}(\sigma_k^+, \sigma_k^-) = \text{Bern}(\frac{e^{A^*(\sigma_k^+)}}{e^{A^*(\sigma_k^+)} + e^{A^*(\sigma_k^-)}})$

Algorithm 1 Preference-based Optimization for Offline RL (**PREFORL**)

Require: Policy parameters θ, offline dataset of trajectories $\mathcal{D} = \{\ldots, \tau_i = (\ldots, s_t^i, a_t^i, \ldots), \ldots\}$, l-length representative segment sampling function $\Sigma(\tau, l)$, probabilistic action degradation function $\mathcal{F}_P(s, a)$, representative segment length k, temperature α, contrastive bias λ, discount factor γ.
Ensure: Policy $\pi_\theta(s)$
for $j = 0, 1, \ldots, N-1$ **do**
  $\mathcal{D}_j^+ = \{\,\}, \mathcal{D}_j^- = \{\,\}$
  for $m = 0, 1, \ldots, M-1$ **do**
    $\tau_m = (\ldots, s_t, a_t, \ldots) \overset{d}{\sim} \mathcal{D}$
    $\sigma = \Sigma(\tau_m, l) = (\ldots, \hat{s}_t, \hat{a}_t, \ldots, \hat{s}_l, \hat{a}_l)$ ▷ Build representative segment σ
    $\mathcal{D}_j^+ = \mathcal{D}_j^+ \cup \{\sigma^+ = (\ldots, \hat{s}_t, \hat{a}_t, \ldots)\} \cup \{\sigma^+\}$ ▷ Collect expert segments $(+)$
    $\mathcal{D}_j^- = \mathcal{D}_j^- \cup \{\sigma^- = (\ldots, \hat{s}_t, \mathcal{F}_P^\sigma(\hat{s}_t, \hat{a}_t), \ldots)\}$ ▷ Construct degraded segments $(-)$
    $\mathcal{D}_j^- = \mathcal{D}_j^- \cup \{\sigma^- = (\ldots, \hat{s}_t, \mathcal{F}_P^\gamma(\hat{s}_t, \hat{a}_t), \ldots)\}$
  end for
  $\theta_{j+1} = \arg\min_\theta \frac{1}{|\mathcal{D}_j^+|T} \sum_{\sigma^\pm \in \mathcal{D}_j^\pm} \sum_{t=0}^{T-1} [-\log \frac{\exp \sum_{\sigma^+} \gamma^t \alpha L^+}{\exp \sum_{\sigma^+} \gamma^t \alpha L^+ + \exp\lambda \sum_{\sigma^-} \gamma^t \alpha L^-}]$,
  where $L^\pm = \log \pi_\theta(\hat{s}_t^\pm, \hat{a}_t^\pm)$. ▷ Update the policy π_θ
end for

and $P_{\hat{A}}(\sigma_k^+, \sigma_k^-) = \text{Bern}(\frac{e^{\hat{A}(\sigma_k^+)}}{e^{\hat{A}(\sigma_k^+)} + e^{\hat{A}(\sigma_k^-)}})$, where Bern denotes Bernoulli distribution. Then the cross-entropy $\mathcal{L}_{\text{CPL}}$ loss function can be re-written in terms of the advantage functions as follows:

$$\mathcal{L}_{\text{CPL}}(\hat{A}, \mathcal{D}) = \mathbb{E}_{\sigma_k^+ \sim \mathcal{D}^+} \left[D_{KL} \left(P_{A^*}(\sigma_k^+, \sigma_k^-) \| P_{\hat{A}}(\sigma_k^+, \sigma_k^-) \right) \right]$$

We show that Algorithm 1 establishes a connection between minimizing $\mathcal{L}_{\text{CPL}}(\hat{A}, \mathcal{D})$ and minimizing the TV divergence between the learned policy π and the expert policy π^*.

Lemma 3.1. *Let* $\pi(a|s) = \frac{e^{\hat{A}(s,a)/\alpha}}{Z(s)}$ *and* $\pi^*(a|s) = \frac{e^{A^*(s,a)/\alpha}}{Z^*(s)}$*, with softmax temperature* $\alpha > 0$*. Suppose that the perturbed segments cover the full action space in each state. Then:*

$$\mathcal{L}_{\text{CPL}}(\hat{A}, \mathcal{D}) \to 0 \quad \Longrightarrow \quad \mathbb{E}_{s \sim d^*} \left[D_{\text{TV}}(\pi^*(\cdot|s) \,\|\, \pi(\cdot|s)) \right] \to 0.$$

The assumption that the perturbed segments cover the action space ensures that segment preferences sufficiently constrain all state-action pairs. Therefore, minimizing the CPL loss encourages policy imitation of the expert on the dataset's state distribution.

4 Experiments and Evaluations

We implemented our algorithm in a tool called PREFORL[1]. In this section, we evaluate PREFORL algorithm in various challenging domains including **MetaWorld** [18], **Adroit** and **Maze2D** from D4RL [20] benchmark, and **Sparse-MuJoCo** proposed in a previous offline RL work [6].

Adroit. The Adroit [1] domain is designed for controlling a 24-DoF simulated Shadow Hand robot to complete different tasks. Demonstration of human experts and scripted controllers are given to evaluate the effectiveness of different RL or non-RL learning algorithms. In D4RL [20], Adroit is re-designed for offline RL setting only. We consider four tasks, i.e., *pen, hammer, relocate* and *door* (see Figure 2). In each task, three different types of datasets are provided to evaluate the robustness of learning algorithm. Among them, two types of datasets are adopted from the original paper [1]: *human* with 25 trajectories collected from human experts, and a large amount of *expert* demonstrations sampled from a fine-tuned RL policy. Besides, each *cloned* is a mixing dataset which combines 50 percentage of expert demonstrations, and 50 percentage episodes sampled from a

[1]PREFORL will be publicly available in the future when the paper is ready to publish.

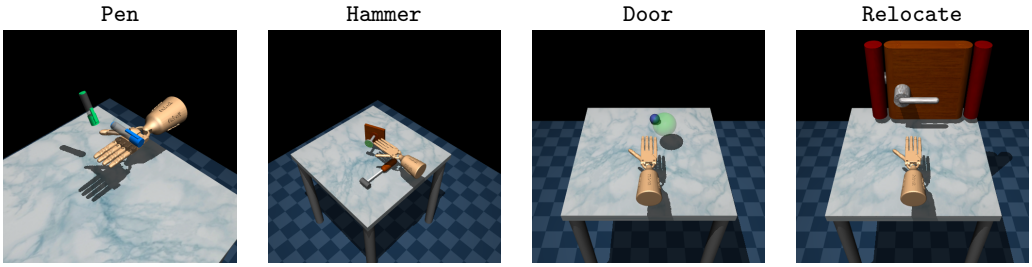


Figure 2: Hand manipulation tasks in **Adroit**.

Task	BC	CQL	IQL	TD3+BC	CDE	ReBRAC	CPL	PREFORL
pen-human	34.4	37.5	78.5	81.8	72.1	103.5	100.1	**119.1**
pen-cloned	56.9	39.2	83.4	61.4	42.1	91.8	91.2	**94.8**
pen-expert	85.1	107.0	128.0	146.0	105.0	**154.1**	130.9	144.8
door-human	0.5	9.9	3.3	-0.1	7.7	0.0	11.9	**15.5**
door-cloned	-0.1	0.4	3.1	0.1	0.1	1.1	3.6	**16.3**
door-expert	34.9	101.5	106.7	84.6	105.9	104.6	105.8	**106.2**
hammer-human	1.5	4.4	1.8	0.4	1.9	0.2	15.1	**16.5**
hammer-cloned	0.8	2.1	1.5	0.8	7.3	6.7	13.2	**28.4**
hammer-expert	125.6	86.7	128.7	117.0	126.3	**133.8**	128.3	128.6
relocate-human	0.0	0.2	0.1	-0.2	0.3	0.0	0.6	**0.9**
relocate-cloned	-0.1	-0.1	0.0	-0.1	0.2	**0.9**	0.5	**0.9**
relocate-expert	101.3	95.0	106.1	107.3	102.6	106.6	110.2	**111.2**

Table 1: Normalized scores of PREFORL against other baselines on D4RL **Adroit** tasks. BC, CQL and IQL scores were taken from [20], TD3+BC and ReBRAC scores were taken from [21], and CDE scores were taken from [6]. Our reported results are averaged over 5 random seeds, and each data point consists of 20 evaluation trajectories.

imitation policy trained on the demonstrations. We choose one imitation learning algorithm (BC) and four offline RL algorithms (CQL, IQL, TD3+BC, CDE and ReBRAC) as baselines. Table 1 denotes the normalized scores of PREFORL algorithms against other baselines on Adroit tasks. Results indicate that PREFORL algorithm demonstrates competitive performance against other baselines and outperforms previous state-of-the-art offline RL algorithm in majority of environments.

Sparse-MuJoCo. The Sparse-MuJoCo benchmark was proposed in CDE [6] and originated from MuJoCo domain in D4RL [20] benchmark. Despite all episodes are collected from inherently dense-reward based environments, the *quality* of each trajectory can be classified into *success* and *failed* categories by examining the episode return. Following the settings in CDE, the return thresholds are set to be the 75-percentile of all episode returns in each dataset. We set rewards to be 0 for all *failed* trajectories in the lower 75 percent, whereas 1 for other *success* trajectories. In evaluation, a trajectory is considered successful when the return is above the threshold and failed otherwise. On Sparse-MuJoCo, we choose BCQ, CQL, IQL, TD3+BC and CDE as baselines. These offline RL algorithms utilize different methods to optimize policies or learn value functions, and all of them leverage sparse reward information. In Table 2, PREFORL demonstrates competitive performance against other offline RL algorithms in all domains, yet only utilizing reward information *indirectly* to construct a sparse optimizing target. Details of experimental settings including dataset formulation and return thresholds can be found in Appendix B.

Maze2D. The Maze2D domain includes navigation tasks aiming to instruct a 2D agent to reach a fixed goal position. Three maze layouts are provided with increasing difficulties, i.e., *umaze*, *medium*, and *large* (see figures in Table 3). Different from above-mentioned domains, the training data distribution in Maze2D differs from its evaluation distribution, and the lengths of the trajectories in the dataset varies. Specifically, in data collection process, a starting position and a goal position are randomly sampled from valid positions in the maze, and an episode does not terminates if the

Task	BCQ	CQL	IQL	TD3+BC	CDE	ReBRAC	CPL	PREFORL
halfcheetah-medium	57.8	97.6	76.6	41.6	82.0	**100.0**	96.0	96.8
walker2d-medium	41.0	17.7	19.5	21.0	53.0	42.0	85.3	**98.0**
hopper-medium	2.0	74.0	0.0	0.0	85.5	96.0	96.0	**100.0**
halfcheetah-medium-expert	24.8	4.2	95.4	0.0	95.2	0.0	47.3	**100.0**
walker2d-medium-expert	87.0	61.6	94.6	32.2	97.0	36.0	**100.0**	**100.0**
hopper-medium-expert	20.0	0.0	94.8	22.0	97.0	21.0	0.0	**98.4**

Table 2: Success rate (in percent) of PREFORL against other baselines on **Sparase-MuJoCo** tasks. PREFORL and CPL results are averaged over 5 random seeds, and each data point consists of 50 evaluation trajectories. Results of other baselines were taken from [6, 21].

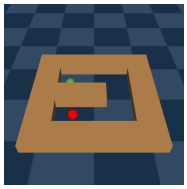
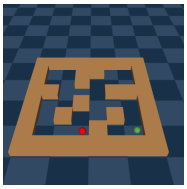
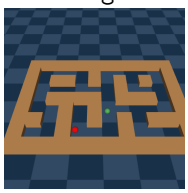
	UMaze	Medium	Large
			
ReBRAC	2.07	**0.71**	0.34
CDE	1.05	0.57	0.55
PREFORL	**2.22**	0.67	**0.63**

Table 3: Figures show navigation tasks in different mazes in **Maze2D**. In each maze, red and green balls denote start and goal positions. The table demonstrates average returns of PREFORL against other baselines on Maze2D tasks. All results are averaged over 5 random seeds, and each data point consists of 50 evaluation trajectories. Note that we use sparse rewards in Maze2D environment, where 1 denotes one successful contact to the sampled goal and 0 otherwise.

agent reach the goal. Instead, a new goal is randomly sampled and the previous successful episode would be collected as if it is an independent trajectory. In evaluation, an agent that always start from a fixed position is required to reach as many goals as possible within maximum episode steps. These goals will be substitute by a newly randomized one if reached. Table 3 demonstrates the average returns of PREFORL and other baselines (ReBRAC, CDE) on Maze2D tasks. The results shows PREFORL outperforms or ReBRAC and CDE in most environments, and can acquire high-quality policies consistently. It also shows that PREFORL performs well in both narrow (Adroit) and diverse (Maze2D) dataset distributions. We exclude the CPL baseline from this evaluation, as the dataset lacks unsuccessful trajectories required for its contrastive learning objective.

MetaWorld. The MetaWorld [18] is a open-source simulated benchmark for meta-reinforcement learning and multi-task learning. It consists of 50 diverse and challenging robotic manipulation tasks. The quantity of tasks and distinction of each environment make it suitable for evaluating online or offline reinforcement learning algorithms. Specifically, we select 16 diverse tasks from this benchmark and many of them are deemed most challenging tasks [22]. To build the offline RL dataset, we use the provided scripted controller to sample 50 expert demonstrations for each selected environment. Note that we do not record any reward signal in any trajectory. Therefore, this is a typical *learning-from-demonstration* problem which typically can only be solve by imitation learning algorithms. To evaluate the feasibility of applying PREFORL on high-dimensional environments, we set the observation space of MetaWorld environments to be an 84×84 RGB **image**. We use Behavior Cloning (BC) as the sole baseline, since standard offline RL algorithms typically fail in settings where only expert demonstrations are available and reward signals are absent. To handle image-based observations, we use a pre-trained ResNet-50 [23] as the image encoder for both PREFORL and BC, and other training details are left in Appendix B. The evaluation results are shown in Table 4. The table shows, although BC is still a strong baseline in high-dimensional goal-achieving tasks, the PREFORL algorithm outperforms it by a large margin in nearly every domains by using sparse and limited artificial preference signals.

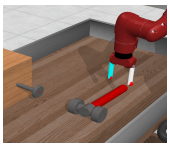
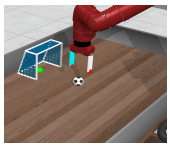
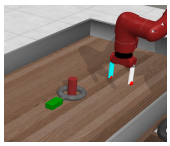
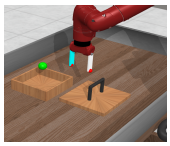
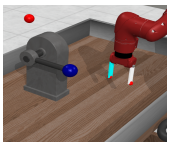
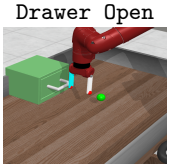
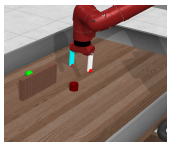
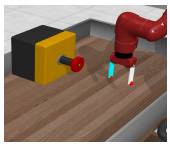
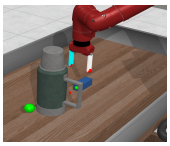
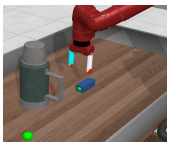
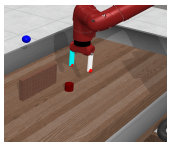
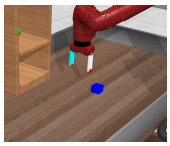
	Hammer	Peg Insert	Peg Unplug	Soccer
				
PREFORL	**98.7** $\pm$ **0.9**	**73.3** $\pm$ **8.2**	**85.3** $\pm$ **3.4**	**67.7** $\pm$ **4.1**
BC	93.3 $\pm$ 12.1	49.0 $\pm$ 5.0	67.3 $\pm$ 6.5	25.0 $\pm$ 15.6
	Window Open	Sweep	Disassemble	Box Close
				
PREFORL	**96.8** $\pm$ **2.8**	**60.7** $\pm$ **10.9**	**90.7** $\pm$ **4.1**	**89.3** $\pm$ **0.9**
BC	91.3 $\pm$ 3.9	43.3 $\pm$ 11.1	82.7 $\pm$ 3.1	77.0 $\pm$ 12.2
	Lever Pull	Drawer Open	Push Wall	Button Press
				
PREFORL	**80.7** $\pm$ **3.4**	**100.0** $\pm$ **0.0**	**84.7** $\pm$ **5.2**	**98.7** $\pm$ **1.2**
BC	57.0 $\pm$ 25.1	86.2 $\pm$ 9.8	47.7 $\pm$ 7.1	77.3 $\pm$ 5.3
	Stick Push	Stick Pull	Pick Place Wall	Soccer
				
PREFORL	**100.0** $\pm$ **0.0**	**96.6** $\pm$ **0.9**	**59.3** $\pm$ **5.2**	**48.0** $\pm$ **3.3**
BC	96.8 $\pm$ 3.0	86.2 $\pm$ 9.8	41.3 $\pm$ 3.0	25.0 $\pm$ 15.6

Table 4: Success rate (in percent) of PREFORL against BC on 16 tasks from **MetaWorld** benchmark. 50 expert demonstrations are provided for each environment. We report the average results of PREFORL over 5 random seeds, and each data point consists of 50 evaluation trajectories.

Summary. In summary, PREFORL achieves strong performance across a range of challenging control tasks spanning both vector-based and image-based high-dimensional environments. Its effectiveness under varying settings—including sparse offline datasets and state-action demonstrations without reward—suggests that PREFORL offers a simple and generalizable framework for long-horizon offline RL problems.

5 Related Work

Preference-based Reinforcement Learning. The mainstream preference-based RL (PbRL) methods often involve learning a reward model to predict the scores of generated sequences, usually from pairwise comparisons, then use this reward model to perform reinforcement learning for policy optimization [10]. Early work of PbRL demonstrate the feasibility of policy learning from preference signals to solve lower-dimensional problems [24, 25, 26], recent works, however, are able to tackle control problems by training deep neural-network policies [27, 28, 29, 30, 31] given sufficient preference labels. Within PbRL, Reinforcement Learning from Human Feedback (RLHF) is a special and popular paradigm that align models with human intent. By eliminating the temporal structure of RL, RLHF frame auto-regressive text-generation as a contextual bandits problem, and many algorithms

[8, 10, 32, 33, 34] proven to work well in large-scale post-training of Large Language Models in general domains [35, 36, 37, 38, 39].

Offline Reinforcement Learning. Similar to offline RL, our work aims to optimize the policy solely from previously collected datasets without further interaction with the environment. This is particular useful in domains where online data collection is costly or unsafe. As a naive imitation learning algorithm, BC [40] often struggles with data distributions that differ from those encountered during training. This also reveals the key challenge in offline RL: out-of-distribution (OOD) generalization. Several offline RL methods have been proposed to address the challenge. Behavior regularization approaches, such as BCQ [41], BRAC [3], BEAR [4], IQL [42] and ReBRAC [21], restrict learned policies to stay close to the dataset's behavior distribution. Another line of work, uncertainty-aware approaches, including MOPO [43], and CQL [2], penalize actions with high uncertainty to mitigate the impact of distributional shift. Besides, Distribution Correction Estimation (DICE)-based methods like CDE [6] and OptiDICE [7] have been proposed to provide a direct behavior-agnostic estimation of stationary distributions to tackle offline RL problems. Recently, decision transformer [44] have introduced sequence modeling-based approaches, leveraging transformers to model trajectories directly. Approaches above optimize policies in various ways, yet they all focus on leveraging high-density transitions to construct learning objects. Our approach PREFORL, however, aggregates local experiences into a sparse preference signal, which implicitly encapsulates both step-wise knowledge for performing fine-grained control, and trajectory-level insight for discovering the optimal solution.

6 Discussion and Conclusion

Limitations. When applying action degradation using function $\mathcal{F}^\sigma$, variance in the Gaussian noise should be tuned accordingly. Although a sufficiently small value is well-suited in practice, knowing the action range is always preferred. When $\mathcal{F}^\gamma$ is invoked, the computation overhead is not negligible as (approximated) nearest neighbor algorithms are time-consuming.

Variants of Degradation Functions. In this paper, we present two methods of generating less preferred trajectories from given offline datasets. In $\mathcal{F}^\sigma$, we add Gaussian noises to ground-truth actions to formulate a *fake* suboptimal trajectory. While in $\mathcal{F}^\gamma$, we leverage nearest-neighbor algorithms to find neighboring actions with certain conditions to substitute the original actions. Different variants are suitable for different environment and offline datasets settings. Specifically, $\mathcal{F}^\gamma$ is useful when data distributions is narrow in the provided dataset. $\mathcal{F}^\sigma$-PREFORL is suitable in *learning from demonstrations* settings where rewards are completely missing.

In current version of PREFORL, we do not modify vector- or image-based states due to following reasons: 1) It is hard to capture the valid range of observation spaces, especially in the image-based environments; 2) Noises applied on input is a common data augmentation method so that the policy is obliged to handle. However, we can consider training adversarial examples [45] in high-dimensional spaces. We leave this as our future work.

Alternative Degradation Methods. Besides methods used in $\mathcal{F}^\sigma$ and $\mathcal{F}^\gamma$, we also considered alternative methods of generating degraded actions. Similar to BCQ [41], we can train a generative model to estimate suboptimal actions. However, training a generative model can be highly unstable and inefficient, and we are unable to control the level of degradation. Another possible solution is inspired by CDE [6]: training an imitation policy using BC to predict the suboptimal actions. This seems to be more sensible, but the performance gap between a BC policy to its corresponding offline dataset is concerning. Through extensive experiments, we found that if the quality gap between preferred and less preferred trajectories increases, the performance of learned policies drops.

Connections to post-training of LLMs. Hejna et al. [12] proved that the DPO is a special and dedicated version of CPL which frame the post-training of LLMs as a contextual bandit problem. Inspired by CPL, our PREFORL algorithm resolves its practicality concerns and proposes a simple framework to generate preference data with limited or no prior knowledge. Conceptually, we hope this can be utilized as a data augmentation method that transform data used in SFT stage into pairs of preference data that benefits the RLHF stage.

Conclusion. We presents a novel and generic preference-based policy optimization algorithm called PREFORL. This algorithm avoids directly learning a value function to mitigate overestimation bias and improves robustness in offline learning. Instead, PREFORL learns a policy by framing a contrastive learning target via contrasting successful trajectories versus failed trajectories and out-of-support behaviors. This is especially valuable if rewards are extremely sparse or completely missing in the given offline dataset, and we provide multiple ways of proposing negative pairing trajectories accordingly. Through extensive experiments, we demonstrate the effectiveness and efficiency of PREFORL on learning policies from sparse offline datasets on various challenging control problems.

References

[1] Aravind Rajeswaran, Vikash Kumar, Abhishek Gupta, Giulia Vezzani, John Schulman, Emanuel Todorov, and Sergey Levine. Learning Complex Dexterous Manipulation with Deep Reinforcement Learning and Demonstrations. In *Proceedings of Robotics: Science and Systems (RSS)*, 2018.

[2] 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. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/0d2b2061826a5df3221116a5085a6052-Paper.pdf.

[3] Yifan Wu, George Tucker, and Ofir Nachum. Behavior regularized offline reinforcement learning, 2019. URL https://arxiv.org/abs/1911.11361.

[4] Aviral Kumar, Justin Fu, George Tucker, and Sergey Levine. *Stabilizing off-policy Q-learning via bootstrapping error reduction*. Curran Associates Inc., Red Hook, NY, USA, 2019.

[5] Scott Fujimoto and Shixiang Gu. A minimalist approach to offline reinforcement learning. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=Q32U7dzWXpc.

[6] Zhepeng Cen, Zuxin Liu, Zitong Wang, Yihang Yao, Henry Lam, and Ding Zhao. Learning from sparse offline datasets via conservative density estimation. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=4WM0OogPTx.

[7] Jongmin Lee, Wonseok Jeon, Byung-Jun Lee, Joelle Pineau, and Kee-Eung Kim. Optidice: Offline policy optimization via stationary distribution correction estimation. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, 2021.

[8] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=HPuSIXJaa9.

[9] Ralph Allan Bradley and Milton E. Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952. ISSN 00063444, 14643510. URL http://www.jstor.org/stable/2334029.

[10] Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/d5e2c0adad503c91f91df240d0cd4e49-Paper.pdf.

[11] W. Bradley Knox, Stephane Hatgis-Kessell, Serena Booth, Scott Niekum, Peter Stone, and Alessandro G Allievi. Models of human preference for learning reward functions. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=hpKJkVoThY.

[12] Joey Hejna, Rafael Rafailov, Harshit Sikchi, Chelsea Finn, Scott Niekum, W. Bradley Knox, and Dorsa Sadigh. Contrastive preference learning: Learning from human feedback without reinforcement learning. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=iX1RjVQODj.

[13] Brian D. Ziebart. Modeling Purposeful Adaptive Behavior with the Principle of Maximum Causal Entropy. 7 2018. doi: 10.1184/R1/6720692.v1. URL https://kilthub.cmu.edu/articles/thesis/Modeling_Purposeful_Adaptive_Behavior_with_the_Principle_of_Maximum_Causal_Entropy/6720692.

[14] Brian D. Ziebart, Andrew Maas, J. Andrew Bagnell, and Anind K. Dey. Maximum entropy inverse reinforcement learning. In *Proceedings of the 23rd National Conference on Artificial Intelligence - Volume 3*, AAAI'08, page 1433–1438. AAAI Press, 2008. ISBN 9781577353683.

[15] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. 2017.

[16] Gaon An, Junhyeok Lee, Xingdong Zuo, Norio Kosaka, Kyung-Min Kim, and Hyun Oh Song. Direct preference-based policy optimization without reward modeling. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=FkAwlqBuy0.

[17] Changyeon Kim, Jongjin Park, Jinwoo Shin, Honglak Lee, Pieter Abbeel, and Kimin Lee. Preference transformer: Modeling human preferences using transformers for RL. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=Peot1SFDX0.

[18] Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan Julian, Karol Hausman, Chelsea Finn, and Sergey Levine. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In Leslie Pack Kaelbling, Danica Kragic, and Komei Sugiura, editors, *Proceedings of the Conference on Robot Learning*, volume 100 of *Proceedings of Machine Learning Research*, pages 1094–1100. PMLR, 30 Oct–01 Nov 2020. URL https://proceedings.mlr.press/v100/yu20a.html.

[19] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017. URL https://arxiv.org/abs/1707.06347.

[20] Justin Fu, Aviral Kumar, Ofir Nachum, George Tucker, and Sergey Levine. D4rl: Datasets for deep data-driven reinforcement learning, 2021. URL https://arxiv.org/abs/2004.07219.

[21] Denis Tarasov, Vladislav Kurenkov, Alexander Nikulin, and Sergey Kolesnikov. Revisiting the minimalist approach to offline reinforcement learning. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=vqGWslLeEw.

[22] Younggyo Seo, Danijar Hafner, Hao Liu, Fangchen Liu, Stephen James, Kimin Lee, and Pieter Abbeel. Masked world models for visual control. In *6th Annual Conference on Robot Learning*, 2022. URL https://openreview.net/forum?id=Bf6on28H0Jv.

[23] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015. URL https://arxiv.org/abs/1512.03385.

[24] Aaron Wilson, Alan Fern, and Prasad Tadepalli. A bayesian approach for policy learning from trajectory preference queries. In F. Pereira, C.J. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012. URL https://proceedings.neurips.cc/paper_files/paper/2012/file/16c222aa19898e5058938167c8ab6c57-Paper.pdf.

[25] Riad Akrour, Marc Schoenauer, and Michèle Sebag. April: active preference learning-based reinforcement learning. In *Proceedings of the 2012th European Conference on Machine Learning and Knowledge Discovery in Databases - Volume Part II*, ECMLPKDD'12, page 116–131, Berlin, Heidelberg, 2012. Springer-Verlag. ISBN 9783642334856.

[26] Róbert Busa-Fekete, Balázs Szörényi, Paul Weng, Weiwei Cheng, and Eyke Hüllermeier. Preference-based reinforcement learning: evolutionary direct policy search using a preference-based racing algorithm. *Mach. Learn.*, 97(3):327–351, December 2014. ISSN 0885-6125. doi: 10.1007/s10994-014-5458-8. URL https://doi.org/10.1007/s10994-014-5458-8.

[27] Dorsa Sadigh, Anca D. Dragan, S. Shankar Sastry, and Sanjit A. Seshia. Active preference-based learning of reward functions. In *Robotics: Science and Systems*, 2017. URL https://api.semanticscholar.org/CorpusID:12226563.

[28] Erdem Biyik and Dorsa Sadigh. Batch active preference-based learning of reward functions. In Aude Billard, Anca Dragan, Jan Peters, and Jun Morimoto, editors, *Proceedings of The 2nd Conference on Robot Learning*, volume 87 of *Proceedings of Machine Learning Research*, pages 519–528. PMLR, 29–31 Oct 2018. URL https://proceedings.mlr.press/v87/biyik18a.html.

[29] Borja Ibarz, Jan Leike, Tobias Pohlen, Geoffrey Irving, Shane Legg, and Dario Amodei. Reward learning from human preferences and demonstrations in atari. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper_files/paper/2018/file/8cbe9ce23f42628c98f80fa0fac8b19a-Paper.pdf.

[30] Daniel Shin, Anca Dragan, and Daniel S. Brown. Benchmarks and algorithms for offline preference-based reward learning. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. URL https://openreview.net/forum?id=TGuXXlbKsn.

[31] Donald Joseph Hejna III and Dorsa Sadigh. Few-shot preference learning for human-in-the-loop RL. In *6th Annual Conference on Robot Learning*, 2022. URL https://openreview.net/forum?id=IKC5TfXLuW0.

[32] Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. Kto: Model alignment as prospect theoretic optimization, 2024. URL https://arxiv.org/abs/2402.01306.

[33] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models, 2024. URL https://arxiv.org/abs/2402.03300.

[34] Jian Hu. Reinforce++: A simple and efficient approach for aligning large language models, 2025. URL https://arxiv.org/abs/2501.03262.

[35] Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA, 2022. Curran Associates Inc. ISBN 9781713871088.

[36] DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jiawei Wang, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan

Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Shengfeng Ye, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wanjia Zhao, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanhong Xu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025. URL `https://arxiv.org/abs/2501.12948`.

[37] DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Haowei Zhang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Li, Hui Qu, J. L. Cai, Jian Liang, Jianzhong Guo, Jiaqi Ni, Jiashi Li, Jiawei Wang, Jin Chen, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, Junxiao Song, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Lei Xu, Leyi Xia, Liang Zhao, Litong Wang, Liyue Zhang, Meng Li, Miaojun Wang, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingming Li, Ning Tian, Panpan Huang, Peiyi Wang, Peng Zhang, Qiancheng Wang, Qihao Zhu, Qinyu Chen, Qiushi Du, R. J. Chen, R. L. Jin, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, Runxin Xu, Ruoyu Zhang, Ruyi Chen, S. S. Li, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shaoqing Wu, Shengfeng Ye, Shengfeng Ye, Shirong Ma, Shiyu Wang, Shuang Zhou, Shuiping Yu, Shunfeng Zhou, Shuting Pan, T. Wang, Tao Yun, Tian Pei, Tianyu Sun, W. L. Xiao, Wangding Zeng, Wanjia Zhao, Wei An, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, X. Q. Li, Xiangyue Jin, Xianzu Wang, Xiao Bi, Xiaodong Liu, Xiaohan Wang, Xiaojin Shen, Xiaokang Chen, Xiaokang Zhang, Xiaosha Chen, Xiaotao Nie, Xiaowen Sun, Xiaoxiang Wang, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xingkai Yu, Xinnan Song, Xinxia Shan, Xinyi Zhou, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, Y. K. Li, Y. Q. Wang, Y. X. Wei, Y. X. Zhu, Yang Zhang, Yanhong Xu, Yanhong Xu, Yanping Huang, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Li, Yaohui Wang, Yi Yu, Yi Zheng, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Ying Tang, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yu Wu, Yuan Ou, Yuchen Zhu, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yukun Zha, Yunfan Xiong, Yunxian Ma, Yuting Yan, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Z. F. Wu, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhen Huang, Zhen Zhang, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhibin Gou, Zhicheng Ma, Zhigang Yan, Zhihong Shao, Zhipeng Xu, Zhiyu Wu, Zhongyu Zhang, Zhuoshu Li, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Ziyi Gao, and Zizheng Pan. Deepseek-v3 technical report, 2025. URL `https://arxiv.org/abs/2412.19437`.

[38] Kimi Team, Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun Xiao, Chenzhuang Du, Chonghua Liao, Chuning Tang, Congcong Wang, Dehao Zhang, Enming Yuan, Enzhe Lu, Fengxiang Tang, Flood Sung, Guangda Wei, Guokun Lai, Haiqing Guo, Han Zhu, Hao Ding, Hao Hu, Hao Yang, Hao Zhang, Haotian Yao, Haotian Zhao, Haoyu Lu, Haoze Li, Haozhen Yu, Hongcheng Gao, Huabin Zheng, Huan Yuan, Jia Chen, Jianhang Guo, Jianlin Su, Jianzhou Wang, Jie Zhao, Jin Zhang, Jingyuan Liu, Junjie Yan, Junyan Wu, Lidong Shi, Ling Ye, Longhui Yu, Mengnan Dong, Neo Zhang, Ningchen Ma, Qiwei Pan, Qucheng Gong, Shaowei Liu, Shengling Ma, Shupeng Wei, Sihan Cao, Siying Huang, Tao Jiang, Weihao Gao, Weimin Xiong, Weiran He, Weixiao Huang, Wenhao Wu, Wenyang He, Xianghui Wei, Xianqing Jia, Xingzhe Wu, Xinran Xu, Xinxing Zu, Xinyu Zhou, Xuehai Pan, Y. Charles, Yang Li, Yangyang Hu, Yangyang Liu, Yanru Chen, Yejie Wang, Yibo Liu, Yidao Qin, Yifeng Liu, Ying Yang, Yiping Bao, Yulun Du, Yuxin Wu, Yuzhi Wang, Zaida Zhou,

Zhaoji Wang, Zhaowei Li, Zhen Zhu, Zheng Zhang, Zhexu Wang, Zhilin Yang, Zhiqi Huang, Zihao Huang, Ziyao Xu, and Zonghan Yang. Kimi k1.5: Scaling reinforcement learning with llms, 2025. URL https://arxiv.org/abs/2501.12599.

[39] Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report, 2025. URL https://arxiv.org/abs/2412.15115.

[40] Dean A. Pomerleau. Alvinn: An autonomous land vehicle in a neural network. In D. Touretzky, editor, *Advances in Neural Information Processing Systems*, volume 1. Morgan-Kaufmann, 1988. URL https://proceedings.neurips.cc/paper_files/paper/1988/file/812b4ba287f5ee0bc9d43bbf5bbe87fb-Paper.pdf.

[41] Scott Fujimoto, David Meger, and Doina Precup. Off-policy deep reinforcement learning without exploration. In *International Conference on Machine Learning*, pages 2052–2062, 2019.

[42] Ilya Kostrikov, Ashvin Nair, and Sergey Levine. Offline reinforcement learning with implicit q-learning. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=68n2s9ZJWF8.

[43] Tianhe Yu, Garrett Thomas, Lantao Yu, Stefano Ermon, James Y Zou, Sergey Levine, Chelsea Finn, and Tengyu Ma. Mopo: Model-based offline policy optimization. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 14129–14142. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/a322852ce0df73e204b7e67cbbef0d0a-Paper.pdf.

[44] Lili Chen, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Michael Laskin, Pieter Abbeel, Aravind Srinivas, and Igor Mordatch. Decision transformer: Reinforcement learning via sequence modeling. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=a7APmM4B9d.

[45] Bo Li, Haoke Xiao, and Lv Tang. Asam: Boosting segment anything model with adversarial tuning. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3699–3710, 2024. doi: 10.1109/CVPR52733.2024.00355.

[46] Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. The faiss library. 2024.

A Proof of Lemma 3.1

Lemma 3.1 Let $\pi(a|s) = \frac{e^{\hat{A}(s,a)/\alpha}}{Z(s)}$ and $\pi^*(a|s) = \frac{e^{A^*(s,a)/\alpha}}{Z^*(s)}$, with softmax temperature $\alpha > 0$. Suppose that the perturbed segments cover the full action space in each state. Then:

$$\mathcal{L}_{\mathrm{CPL}}(\hat{A}, \mathcal{D}) \to 0 \quad \Longrightarrow \quad \mathbb{E}_{s\sim d^*}\left[D_{\mathrm{TV}}(\pi^*(\cdot|s) \,\|\, \pi(\cdot|s))\right] \to 0.$$

Proof. Let each segment $\sigma_k = (s_0, a_0, \ldots, s_{k-1}, a_{k-1})$ have segment advantage

$$A(\sigma_k) = \sum_{t=0}^{k-1} \gamma^t A(s_t, a_t).$$

If $\mathcal{L}_{\mathrm{CPL}} \to 0$, then for all segment pairs in the dataset:

$$P_{A^*}[\sigma_k^+, \sigma_k^-] \approx P_{\hat{A}}[\sigma_k^+, \sigma_k^-].$$

This implies:

$$A^*(\sigma_k^+) - A^*(\sigma_k^-) \approx \hat{A}(\sigma_k^+) - \hat{A}(\sigma_k^-),$$

For perturbed segments σ_k^- with the same states as σ_k^+, the only difference is in the actions. By induction on k (noticed that k is uniformly sampled), we have:

$$\hat{A}(s_t, a_t) - \hat{A}(s_t, a_t') \approx A^*(s_t, a_t) - A^*(s_t, a_t') \quad \forall t.$$

As long as we generate sufficient segment perturbations that vary the actions in each state (by assumption), we collect enough constraints to pin down the function $\hat{A}(s, \cdot)$ up to additive state-dependent constants $c(s)$.

$$\hat{A}(s, a) = A^*(s, a) + c(s).$$

Thus, we have:

$$\begin{aligned}
&\mathbb{E}_{s\sim d^*}\left[D_{\mathrm{TV}}(\pi^*(\cdot|s) \,\|\, \pi(\cdot|s))\right] \\
&= \mathbb{E}_{s\sim d^*}\left[D_{\mathrm{TV}}\left(\frac{e^{A^*(s,a)/\alpha}}{\sum_{a'} e^{A^*(s,a')/\alpha}} \,\middle\|\, \frac{e^{\hat{A}(s,a)/\alpha}}{\sum_{a'} e^{\hat{A}(s,a')/\alpha}}\right)\right] \\
&\approx \mathbb{E}_{s\sim d^*}\left[D_{\mathrm{TV}}\left(\frac{e^{A^*(s,a)/\alpha}}{\sum_{a'} e^{A^*(s,a')/\alpha}} \,\middle\|\, \frac{e^{(A^*(s,a)+c(s))/\alpha}}{\sum_{a'} e^{(A^*(s,a')+c(s))/\alpha}}\right)\right] \\
&\to 0.
\end{aligned}$$

□

B Experiment Details

We use 8 × A100 80G Nvidia GPUs for experiments. In this section, we discuss more experiment details including task formulation and dataset construction, as well as training details including network architectures and hyperparameters.

B.1 Adroit

Tasks. The Adroit in D4RL [20] contains four manipulation tasks (*pen, hammer, door* and *relocate*), and three types of datasets (*expert, human* and *cloned*). In human setting, 25 human-generated high-quality trajectories are collected in each dataset. In expert datasets, a scripted controller is used to generate 5K successful trajectories for generating offline dataset. However, cloned is a special dataset that contains 5K both success and failed trajectories, as half of the episodes are collected from expert demonstrations, and the other half are sampled from an suboptimal imitation policy. The Adroit dataset is a typical narrow distribution dataset because the tasks in Adroit contains relatively fixed goals and traces. This makes it suitable for both $\mathcal{F}^\sigma$ and $\mathcal{F}^\gamma$ action degradation. In practice, we use Approximated Nearest Neighbor (ANN) search method `IndexIVFFlat` implemented in FAISS [46] to search 10 nearest neighbor states. If any state with 10 percent less reward is found, we use its corresponding action as the degraded action to perform contrastive learning.

Description	Value
Discount factor γ	1.0
Biased regularizer value α	0.1
Contrastive bias λ	0.25
Contrastive segments length l	64
Batch size	20
Number of gradient steps	15000
Learning rate	0.0003
Action degradation function $\mathcal{F}$	$\mathcal{F}^\sigma$, $\mathcal{F}^\gamma$
Gaussian noise in $\mathcal{F}^\sigma$	$\mathcal{N}(0, 0.01^2)$
Nearest neighbor search in $\mathcal{F}^\gamma$	`IndexIVFFlat`
Number of probes	10
Condition `cond` in $\mathcal{F}^\gamma$	Reward is at least 10% smaller
Policy network	MLP `(1024, 1024, 1024)`
Activation	ReLU

Table 5: Hyperparameters of PREFORL in training **Adroit** tasks.

Description	Value
Discount factor γ	1.0
Biased regularizer value α	0.1
Contrastive bias λ	0.5
Contrastive segments length l	100
Batch size	64
Number of gradient steps	15000
Learning rate	0.0003
Action degradation function $\mathcal{F}$	$\mathcal{F}^\sigma$
Gaussian noise in $\mathcal{F}^\sigma$	$\mathcal{N}(0, 0.01^2)$
Image encoder	Pre-trained `ResNet-50`[2]
Policy network	MLP `(1024, 1024, 1024)`
Activation	ReLU

Table 6: Hyperparameters of PREFORL in training **MetaWorld** tasks.

Hyperparameters. The hyperparameters listed in Table 5 are used to train Adroit policies using PREFORL algorithm.

B.2 MetaWorld

Tasks. We set tasks in MetaWorld with image-based observations; hence, we need a network structure to process the RGB image. We choose a pre-trained ResNet-50 [23] model as the image encoder for both BC and PREFORL. Unlike many previous works, we do not freeze the ResNet model during training.

Hyperparameters. The hyperparameters listed in Table 6 are used to train MetaWorld policies using PREFORL algorithm.

B.3 Maze2D

Hyperparameters. The hyperparameters listed in Table 7 are used to train Maze2D policies using PREFORL algorithm.

[2]Model is available at: https://download.pytorch.org/models/resnet50-11ad3fa6.pth

Description	Value
Discount factor γ	1.0
Biased regularizer value α	0.1
Contrastive bias λ	0.5
Contrastive segments length l	100
Batch size	64
Number of gradient steps	500
Learning rate	0.0003
Action degradation function $\mathcal{F}$	$\mathcal{F}^\sigma$
Gaussian noise in $\mathcal{F}^\sigma$	$\mathcal{N}(0, 0.01^2)$
Policy network	MLP `(1024, 1024, 1024)`
Activation	ReLU

Table 7: Hyperparameters of PREFORL in training **Maze2D** tasks.

Task	Return Threshold
halfcheetah-medium	4909.1
walker2d-medium	3697.8
hopper-medium	1621.5
halfcheetah-medium-expert	10703.4
walker2d-medium-expert	4924.8
hopper-medium-expert	3561.9

Table 8: The return thresholds for **Sparse-MuJoCo** tasks.

B.4 Sparse-MuJoCo

Tasks. We adopt "-v2" tasks in D4RL for Sparse-MuJoCo domains. We convert dense rewards in the offline dataset into sparse rewards as stated in the main text, and return thresholds for each environment are listed in Table 8. Note that we perform binary judgment in evaluation, i.e., a trajectory is considered successful if, and only if, its return exceeds the corresponding threshold.

Hyperparameters. The hyperparameters listed in Table 9 are used to train Sparse-MuJoCo policies using PREFORL algorithm.

Description	Value
Discount factor γ	1.0
Biased regularizer value α	0.1
Contrastive bias λ	0.5
Contrastive segments length l	100
Batch size	64
Number of gradient steps	15000
Learning rate	0.0003
Action degradation function $\mathcal{F}$	$\mathcal{F}^\sigma$
Gaussian noise in $\mathcal{F}^\sigma$	$\mathcal{N}(0, 0.01^2)$
Policy network	MLP `(1024, 1024, 1024)`
Activation	ReLU

Table 9: Hyperparameters of PREFORL in training **Sparse-MuJoCo** tasks.

C Impact of Noise Level

In limitation, we discussed that the noise level of Gaussian noise should be tuned according to environments. To acquire a suitable degradation level utilized in PREFORL for each environments, we conduct several hyperparameter investigation on MetaWorld and Adroit tasks (in Figure 3). In all experiments, we assume we have access to the ground truth action range of each environment. This is not strictly necessary but always preferred, as approximated action ranges given by offline datasets are imprecise and conservative. The conservative estimation may lead to insufficient exploration, especially when datasets distributions are narrow.

We select two environments in MetaWorld: `Peg-Unplug` and `Peg-Insert`, and six different levels of noises. Results demonstrate that when the noise level is set to be a reasonably small number, i.e., around 1% to 2%, the success rates for both environments achieve their highest level. If the noise level is too small (0.5%), the distinction between original actions to degraded actions may be too subtle for PREFORL loss to differentiate. This leads to insufficient exploration and optimization. Nevertheless, setting aggressive noise levels (e.g., 5% and 8%) may also do harm to the PREFORL algorithm. This is reasonable and intuitive, as applying such aggressive noises may break the semantics of the trajectories and yield catastrophic forgetting. Note that the core idea of PREFORL is to contrast optimal trajectories versus suboptimal but not poorly-behaved ones.

We select all four expert datasets in Adroit: *Pen-Expert, Door-Expert, Hammer-Expert* and *Relocate-Expert*, and evaluate the effect of four different noise levels accordingly. In Figure 3, the success rates for Adroit tasks keep unchanged across different levels of Gaussian noises. This trend indicates that if a task can be *solved* by PREFORL, the noise levels have little impact on performances.

In summary, finding a reasonably small level of noise is critical to the performance of PREFORL algorithm. Small numbers may lead to conservative exploration and optimization, yet large numbers yield aggressive exploration and may lead to suboptimal policies.

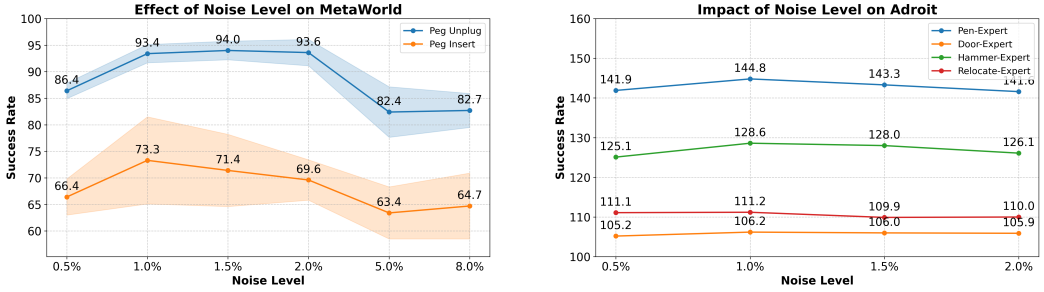


Figure 3: The left figure denotes the effect of noise level on two MetaWorld tasks. The right figure denotes the effect of noise level on Adroit tasks with expert dataset. Results are averaged over 5 run seeds, and each data point is collect by 50 evaluation trajectories.

D Training Curves

The training curves of full dataset experiments are shown in Figure 4. From the figure we can observe that the PREFORL converges quickly. The training curves are also stable, especially in the expert datasets training. Note that PREFORL utilizes a very sparse contrastive learning optimizing target, so that we can expect the number of gradient steps is ten times smaller than other offline RL methods [6, 21].

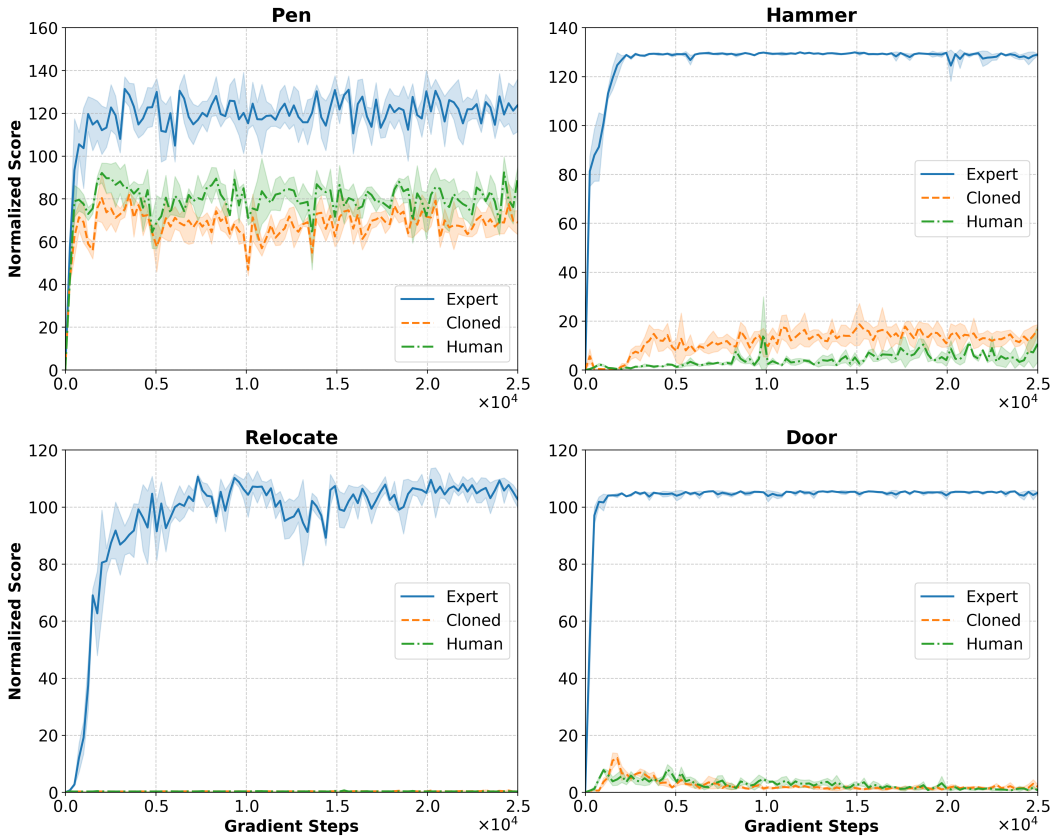


Figure 4: The training curves of PREFORL. The x-axis denotes number of gradient steps and the y-axis denotes the normalized scores. We use 5 random seeds for each environment, and each data point consists of 50 evaluation trajectories. The shadow regions denote the standard deviation of mean values across different seeds.

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 contribution of this paper is clearly stated and concluded.

   Guidelines:

   - The answer NA 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 NA 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: We discussed limitations in the final section.

   Guidelines:

   - The answer NA 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: We include proofs and analysis.

Guidelines:

- The answer NA 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: We include reproducible results and plan to release the code.

   Guidelines:

   - The answer NA 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 datasets are public and we plan to release the code.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.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://nips.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, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We include experimental settings in both main text and appendix.

   Guidelines:

   - The answer NA 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: We provide experiment results that support our claims.

   Guidelines:

   - The answer NA 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: We include computation resources in the paper.

   Guidelines:

   - The answer NA 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: We follow the code of conduct.

   Guidelines:

   - The answer NA 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: [NA]

    Justification: This work performs experiments in virtual reinforcement learning environments and poses no societal impacts.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA 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., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: This paper poses no such risks.

    Guidelines:

    - The answer NA 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: We cited them properly.

    Guidelines:

    - The answer NA 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: [NA]

    Justification: We plan to release our code in the future.

    Guidelines:

    - The answer NA 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: [NA]

    Justification: No human crowdsourcing involves.

    Guidelines:

    - The answer NA 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: [NA]

    Justification: No such potential risks.

    Guidelines:

    - The answer NA 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 rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

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

Guidelines:

- The answer NA 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 (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.