ExID: Offline RL with Intuitive Expert Insights in Limited-Data Settings

Anonymous Author(s)
Affiliation
Address
email

Abstract

With the ability to learn from static datasets, Offline Reinforcement Learning (RL) emerges as a compelling avenue for real-world applications. However, state-of-the-art offline RL algorithms perform sub-optimally when confronted with limited data confined to specific regions within the state space. The performance degradation is attributed to the inability of offline RL algorithms to learn appropriate actions for rare or unseen observations. This paper proposes a novel domain knowledge-based regularization technique and adaptively refines the initial domain knowledge to considerably boost performance in limited data with partially omitted states. The key insight is that the regularization term mitigates erroneous actions for sparse samples and unobserved states covered by domain knowledge. Empirical evaluations on standard discrete environment datasets demonstrate a substantial average performance increase compared to ensemble of domain knowledge and existing offline RL algorithms operating on limited data.

1 Introduction

Offline RL [9, 1], also referred to as batch RL, is a learning approach that focuses on extracting knowledge solely from static datasets. This class of algorithms has a wider range of applications being particularly appealing to real-world data sets from business [46], healthcare [25], and robotics [35]. However, offline RL poses unique challenges, including over-fitting and the need for generalization to data not present in the dataset. To surpass the behavior policy, offline RL algorithms need to query Q values of actions not in the dataset, causing extrapolation errors [21]. Most offline RL algorithms address this problem by enforcing constraints that ensure that the learned policy does not deviate too far away from the data set's state action distribution [13, 11] or is conservative towards Out-of-Distribution (OOD) actions [21, 20]. However, such approaches are designed on coherent batches [13], which do not account for OOD states.

In many domains, such as business and healthcare, available data is scarce and often confined to expert behaviors within a limited state space. *For example, a sales recommendation system, where historic data may not contain details about many active users and operator gives coupon of higher value to attract sales.* Learning on such limited data sets can curtail the generalization capabilities of state-of-the-art (SOTA) offline RL algorithms, resulting in sub-optimal performance [23]. We illustrate this limitation via Fig 1. In Fig 1a) the state action space of a simple Mountain Car environment [27] is plotted for an expert dataset [32] and a partial dataset with first 10% samples from the entire dataset. Fig 1b) shows the average reward obtained over these data sets and the average difference between the Q value of action taken by the under-performing Conservative Q Learning (CQL) [21] agent and the action in the full expert dataset for unseen states. It can be observed that the performance of the offline RL agent considerably drops. This is attributed to the critic overestimating the Q value of non-optimal actions for states that do not occur in the dataset while training.

Submitted to 38th Conference on Neural Information Processing Systems (NeurIPS 2024). Do not distribute.

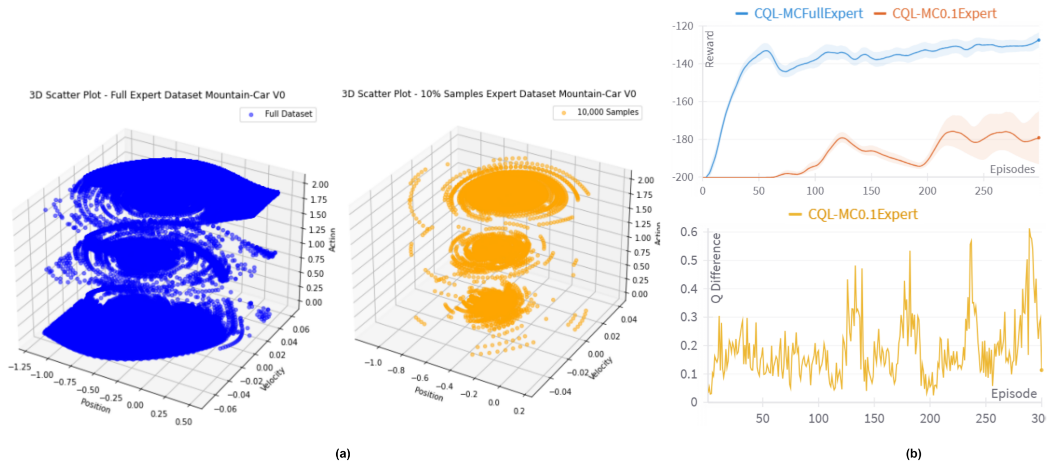


Figure 1: a) Full expert, Mountain Car dataset, and reduced dataset with first 10% samples showing distribution of state (position, velocity) and action b) CQL agent converging to a sub-optimal policy for reduced dataset exhibiting high Q values for actions different from actions in the expert dataset for unseen states.

In numerous real-world applications, expert insights regarding the general behavior of a policy are often accessible [33]. *For example, sales operators often distribute lower discount coupons to active users to maximize profit*. While these insights may not be optimal, they serve as valuable guidelines for understanding the overall behavior of the policy. A rich literature in knowledge distillation [18] has shown that teacher networks trained on domain knowledge can transfer knowledge to another network unaware of it. This work aims to leverage a teacher network mimicking simple decision tree-based domain knowledge to help offline RL generalize in limited data settings.

The paper makes the following novel contributions:

- We introduce an algorithm dubbed **ExID**, leveraging intuitive human obtainable expert insights. The domain expertise is incorporated into a teacher policy, which improves offline RL in limited-data settings through regularization.
- The teacher based on expected performance improvement of the offline policy during training, improving the teacher network beyond initial heuristics.
- We demonstrate the effectiveness of our methodology on *real sales promotion dataset*, several discrete OpenAI gym and Minigrid environments with standard offline RL data sets and show that ExID significantly exceeds the performance when faced with limited data.

2 Related Work

This work improves offline RL learning on batches sampled from static datasets using domain expertise. One of the major concerns in offline RL is the erroneous extrapolation of OOD actions [13]. Two techniques have been studied in the literature to prevent such errors. 1) Constraining the policy to be close to the behavior policy 2) Penalizing overly optimistic Q values [24]. We discuss a few relevant algorithms following these principles. In Batch-Constrained deep Q-learning (BCQ) [13] candidate actions sampled from an adversarial generative model are considered, aiming to balance proximity to the batch while enhancing action diversity. Algorithms like Random Ensemble Mixture Model (REM) [2], Ensemble-Diversified Actor-Critic (EDAC) [3] and Uncertainty Weighted Actor-Critic (UWAC) [42] penalize the Q value according to uncertainty by either using Q ensemble networks or directly weighting the loss with uncertainty. CQL [21] enforces regularization on Q-functions by incorporating a term that reduces Q-values for OOD actions while increasing Q-values for actions within the expected distribution. However, these algorithms do not handle OOD actions for states not in the static dataset and can have errors induced by changes in transition probability.
Integration of domain knowledge in offline RL, though an important avenue, has not yet been

extensively explored. Domain knowledge incorporation has improved online RL with tight regret bounds [33, 4]. In offline RL, bootstrapping via blending heuristics computed using Monte-Carlo returns with rewards has shown to outperform SOTA algorithms by 9% [15]. Recent works improve offline RL by incorporating a safety expert [40] and preference query [44], contrary to our work which improves imperfect domain knowledge. The closest to our work is Domain Knowledge guided Q learning (DKQ) [46] where domain knowledge is represented in terms of action importance and the Q value is weighted according to importance. However, obtaining action importance in practical scenarios is nontrivial.

3 Preliminaries

A DRL setting is represented by a Markov Decision Process (MDP) formalized as $(S, A, T, r, \rho_0, \gamma)$. Here, S denotes the state space, A signifies the action space, $T(s'|s,a)$ represents the transition probability distribution, $r : S \times A \to \mathbb{R}$ is the reward function, ρ_0 represents the initial state distribution, and $\gamma \in (0, 1]$ is the discount factor. The primary objective of any DRL algorithm is to identify an optimal policy $\pi(a|s)$ that maximizes $\mathbb{E}_{s_t,a_t}[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)]$ where, $s_0 \sim d_0(.), a_t \sim \pi(.|s_t)$, and $s' \sim T(.|s_t, a_t)$. Deep Q networks (DQNs) [26] learn this objective by minimizing the Bellman residual $(Q_\theta(s,a) - B^{\pi_\theta} Q_\theta(s,a))^2$ where $B^{\pi_\theta} Q_\theta(s,a) = \mathbb{E}_{s' \sim T}[r(s,a) + \gamma \mathbb{E}_{a' \sim \pi_\theta(.|s')}[Q_{\theta'}(s',a')]]$. The policy π_θ chooses actions that maximize the Q value $\max_{a' \in A} Q_\theta(s',a')$. However, in offline RL where transitions are sampled from a pre-collected dataset $\mathcal{B}$, the chosen action a' may exhibit a bias towards OOD actions with inaccurately high Q-values. To handle the erroneous propagation from OOD actions, CQL [22] learns conservative Q values by penalizing OOD actions. The CQL loss for discrete action space is given by

$$\mathcal{L}_{cql}(\theta) = \min_Q \alpha \, \mathbb{E}_{s \sim \mathcal{B}}[log \sum_a exp(Q_\theta(s,a)) - \\ \mathbb{E}_{a \sim \mathcal{B}|s}[Q_\theta(s,a)]] + \frac{1}{2} \mathbb{E}_{s,a,s' \sim \mathcal{B}}[Q_\theta - Q_{\theta'}]^2 \tag{1}$$

Eq. 1 encourages the policy to be close to the actions seen in the dataset. However, CQL works on the assumption of coherent batches, i.e., if $(s, a, s') \in \mathcal{B}$, then $s' \in \mathcal{B}$. There is no provision for handling OOD actions for $s \notin \mathcal{B}$, which can lead to policy failure when data is limited. In the next sections, we present ExID, a domain knowledge-based approach to improve performance in data-scarce scenarios.

4 Problem Setting and Methodology

In our problem setting, the RL agent learns the policy on a limited dataset with rare and unseen demonstrations. We define the characteristics of this dataset as follows:

Definition 4.1. A reduced buffer $\mathcal{B}_r$ is a proper subset of the full dataset $\mathcal{B}$ i.e., $\mathcal{B}_r \subset \mathcal{B}$ satisfying the following conditions:

- Some states in $\mathcal{B}$ are not present in $\mathcal{B}_r$, i.e., $\exists s' \in \mathcal{B} \wedge \forall (s,a,s') : (s,a,s') \notin \mathcal{B}_r$
- The number of samples $N(s,a,s')$ for some transitions in $\mathcal{B}$ are less in $\mathcal{B}_r$ i.e, $\exists (s,a,s') \in \mathcal{B} : N(s,a,s')_{\mathcal{B}_r} < N(s,a,s')_{\mathcal{B}}$

We observe, *performing $Q-Learning$ by sampling from a limited buffer $\mathcal{B}_r$ may not converge to an optimal policy for the MDP $M_{\mathcal{B}}$ representing the full buffer.* This can be shown as a special case of (Theorem 1,[13]) as $p_{\mathcal{B}}(s'|s,a) \neq p_{\mathcal{B}_r}(s'|s,a)$ and no Q updates for $(s,a) \notin \mathcal{B}_r$ leading to sub-optimal policy. Please refer to the App. B for analysis and example.

We also assume a set of common sense rules in the form of domain knowledge is available. Domain knowledge $\mathcal{D}$ is defined as hierarchical decision nodes capturing $S \to A$ as represented by Eq. 2. Each decision node T_{η_i} is represented by a constraint ϕ_{η_i} and Boolean indicator μ_{η_i} function selects the branch to be traversed based on ϕ_{η_i}.

$$Action = \begin{cases} a_{\eta_i} & \text{if } leaf \\ \mu_{\eta_i} T_{\eta_i \swarrow}(s) + (1 - \mu_{\eta_i}) T_{\eta_i \searrow}(s) & \text{o/w} \end{cases}$$
$$\mu_{\eta_i}(s) = \begin{cases} 1 & \text{if } s \models \phi_{\eta_i} \\ 0 & \text{o/w} \end{cases} \tag{2}$$

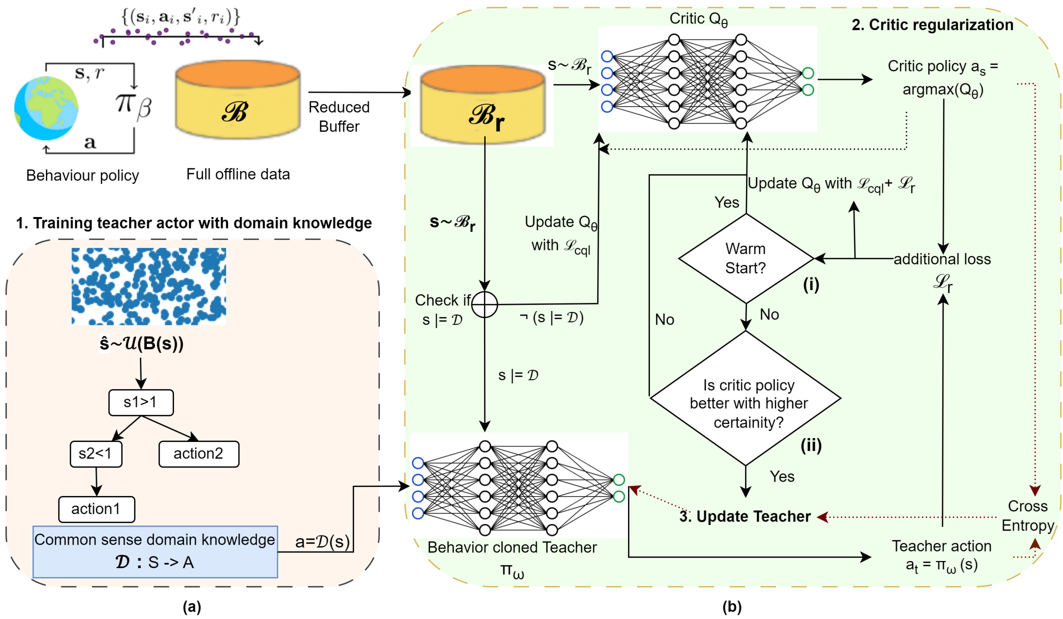


Figure 2: Overview of the proposed methodology (a) Training a teacher policy network with domain knowledge and synthetic data (b) Updating the offline RL critic network with teacher network

We assume that $\mathcal{D}$ gives heuristically reasonable actions for $s \models D$ and $S_{\mathcal{D}} \cap S_{\mathcal{B}_r} \neq \emptyset$ where $S_{\mathcal{D}}, S_{\mathcal{B}_r}$ are the state coverage of $\mathcal{D}$ and $\mathcal{B}_r$.

Training Teacher: An overview of our methodology is depicted in Fig 2. We first construct a trainable actor network π_t^ω parameterized by ω from $\mathcal{D}$, Fig 2 step 1. For training π_t^ω synthetic data $\hat{S}$ is generated by sampling states from a uniform random distribution over state boundaries $B(s)$, $\hat{S} = \mathcal{U}(B(S))$. Note that this does not represent the true state distribution and may have state combinations that will never occur. We train π_t^ω using behavior cloning where state $\hat{s} \sim \hat{S}$ is checked with root decision node in Eq. 2. A random action is chosen if $\hat{s}$ does not satisfy decision node T_{η_0} or leaf action is absent. If $\hat{s}$ satisfies a T_{η_i}, T_{η_i} is traversed and action a_{η_i} is returned from the leaf node. This is illustrated in Fig 2 (a). We term the pre-trained actor network π_t^ω as the teacher policy.

Regularizing Critic: We now introduce Algo 1 (App C) to train an offline RL agent on $\mathcal{B}_r$. Algo 1 takes $\mathcal{B}_r$ and pretrained π_t^ω as input. The algorithm uses two hyper-parameters, warm start parameter k and mixing parameter λ. A critic network Q_s^θ with Monte-Carlo (MC) dropout and target network $Q_s^{\theta'}$ are initialized. ExID is divided into two phases. In the first phase, we aim to warm start the critic network Q_s^θ with actions from π_t^ω as shown in Fig 2b(i). However, this must be done selectively as the teacher's policy is random around the states that do not satisfy domain knowledge. In each iteration, we first check the states sampled from a mini-batch of $\mathcal{B}_r$ with $\mathcal{D}$. For the states which satisfy $\mathcal{D}$ we compute the teacher action $\pi_t^\omega(s)$ and critic's action $\text{argmax}_a(Q_s^\theta(s,a))$ and collect it in lists a_t, a_s, Algo 1 lines 4-10. Our main objective is to keep actions chosen by the critic network for $s \models \mathcal{D}$ close to the teacher's policy. To achieve this, we introduce a regularization term:

$$\mathcal{L}_r(\theta) = \underbrace{\mathbb{E}_{s \sim \mathcal{B}_r \wedge s \models \mathcal{D}}}_{\text{states matching domain rule}} \underbrace{[Q_s^\theta(s, a_s) - Q_s^\theta(s, a_t)]^2}_{\text{Q regularizer}} \tag{3}$$

Eq 3 incentivizes the critic to increase Q values for actions from π_t^ω and decreases Q values for other actions when $\text{argmax}_a(Q_s^\theta(s,a)) \neq \pi_t^\omega(s)$ for states that satisfy domain knowledge. Note that Eq 3 will only be 0 when $\text{argmax}_a(Q_s^\theta(s,a)) = \pi_t^\omega(s)$ for $s \models \mathcal{D}$. It is also set to 0 for $s \not\models \mathcal{D}$. However, since π_t^ω mimicking heuristic rules is sub-optimal, it is also important to incorporate learning from the data. The final loss is a combination of Eq. 1 and Eq. 3 with a mixing parameter $\lambda \in [0, 1]$:

$$\mathcal{L}(\theta) = \mathcal{L}_{cql}(\theta) + \lambda \mathbb{E}_{s\sim\mathcal{B}_r \wedge s\models\mathcal{D}}[Q_s^\theta(s, a_s) - Q_s^\theta(s, a_t)]^2 \tag{4}$$

The choice of λ and the warm start parameter k depends on the quality of $\mathcal{D}$. In the case of perfect domain knowledge, λ would be set to 1, and setting λ to 0 would lead to the vanilla CQL loss. Mixing both the losses allows the critic to learn both from the data in $\mathcal{B}_r$ and knowledge in $\mathcal{D}$.

Updating Teacher: Given a reasonable warm start, the critic is expected to give higher Q values for optimal actions for $s \in \mathcal{D} \cap \mathcal{B}_r$ as it learns from data. We aim to leverage this knowledge to enhance the initial teacher policy π_t^ω trained on heuristic domain knowledge. For $s \sim \mathcal{B}$ and $s \models \mathcal{D}$, we calculate the average Q values over critic actions and teacher actions and check which one is higher in Algo 1 line 11 which refers to Cond. 6. For brevity $\mathbb{E}_{s\sim\mathcal{B}_r\wedge s\models\mathcal{D}}$ is written as $\mathbb{E}$. If $\mathbb{E}(Q_s^\theta(s, a_s)) > \mathbb{E}(Q_s^\theta(s, a_t))$ it denotes the critic expects a better return on an average over its own policy than the teacher's policy. Hence, we can use the critic's policy to update π_t^ω, making it better over $\mathcal{D}$. However, only checking the critic's value can be erroneous as the critic can have high values for OOD actions. We check the average uncertainty of the predicted Q values to prevent the teacher from getting updated by OOD actions. Uncertainty has been shown to be a good metric for OOD action detection by [42, 3]. A well-established methodology to capture uncertainty is predictive variance, which takes inspiration from Bayesian formulation for the critic function and aims to maximize $p(\theta|X, Y) = p(Y|X, \theta)p(\theta)/p(Y|X)$ where $X = (s, a)$ and Y represents the true Q value of the states. However, $p(Y|X)$ is generally intractable and is approximated using Monte Carlo (MC) dropout, which involves including dropout before every layer of the critic network and using it during inference [14]. Following [42], we measure the uncertainty of prediction using Eq 5.

$$Var^T[Q(s, a)] \approx \frac{1}{T}\sum_{t=1}^{T}[Q(s, a) - \bar{Q}(s, a)]^2 \tag{5}$$

Eq 5 estimates the variance of Q value $Q(s, a)$ for an action a using T forward passes on the $Q_s^\theta(s, a)$ with dropout where $\bar{Q}(s, a)$ represents the predictive mean. We check the average uncertainty of the Q value for action chosen by the critic and teacher policy over the states that match domain knowledge in a batch. The teacher network is updated using the critic's action only when the policy expects a higher average Q return on its action and the average uncertainty of taking this action is lower than the teacher action. $\mathbb{E}(Var^T Q_s^\theta(s_r, a_s)) < \mathbb{E}(Var^T Q_s^\theta(s_r, a_t))$ indicates the actions were learned from the expert data in the buffer and are not OOD samples. The condition is summarized in cond. 6:

$$\begin{gathered}\mathbb{E}(Q_s^\theta(s_r, a_s)) > \mathbb{E}(Q_s^\theta(s_r, a_t)) \wedge \\ \mathbb{E}(Var^T Q_s^\theta(s_r, a_s)) < \mathbb{E}(Var^T Q_s^\theta(s_r, a_t))\end{gathered} \tag{6}$$

We update the teacher with cross-entropy described in Eq 7:

$$\mathcal{L}(\omega) = -\sum_{s\models D}(\pi_t^\omega(s) log(\pi_s(s))) \tag{7}$$

where, $\pi_s(s, a) = \frac{e^{Q(s,a)}}{\sum_{a'} Q(s,a')}$. When the critic's policy is better than the teacher's policy, $\mathcal{L}_r(\theta)$ is set to 0 Algo 1 Lines 11 to 13. Finally, the critic network is updated using calculated loss $\mathcal{L}(\theta)$ Algo 1 Lines 17-18. We theoretically analyse the implications of using ExID in propositions 4.2 and 4.3.

Proposition 4.2. *Denote $\hat{\pi}$ as the policy learned by ExID, π_u as any offline RL policy learned on $\mathcal{B}_r$ and optimal Q function as Q^* and V function as V^*. Then it holds that*

$$\eta(\hat{\pi}) - \eta(\pi_u) \geq \mathbb{E}_{s\sim O|\pi_u}[V^*(s) - Q^*(s, \pi_u(s))] - \bar{\rho}_{\hat{\pi}}\alpha$$

Where $\alpha = \mathbb{E}_{s\sim O}[V^*(s) - Q^*(s, \hat{\pi}(s))]$, $\bar{\rho}_\pi(s) = [\frac{1}{|S_{\hat{\pi}}|(1-\gamma)}, \frac{1}{1-\gamma}]$ ($| S_{\hat{\pi}} |$ is the number of different states observed by $\hat{\pi}$) and $O \notin \mathcal{B}_r$. Here α denotes the quality of regularized action for $s \notin \mathcal{B}_r$. Hence, updating π_t^ω is important as high divergence of action from the optimal can lead to performance degradation. In offline RL, the extrapolation error for non optimal action is usually high for states not observed in dataset (as illustrated in 1b), regularization can lead performance improvement when π_t^ω is reasonable. Furthermore, in ExID coarse actions from π_t^ω are updated driving them closer to the optimal actions, improving the performance lower bound. Additionally π_t^ω increases $| S_{\hat{\pi}} |$ making $\bar{\rho}_\pi \ll 1$ in practice further improving the performance lower bound. *Proof is deferred to App. A.*

Proposition 4.3. *ExID reduces generalization error if* $Q^*(s, \pi_t^\omega(s)) > Q^*(s, \pi_u(s))$ *for* $s \in \mathcal{D} \cap \mathcal{B}_r$.

Proof is deferred to App. A. In the next section, we discuss our empirical evaluations.

5 Empirical Evaluations

We investigate the following through our empirical evaluations: *1. Does ExID perform better than combining $\mathcal{D}$ and offline RL algos on different environments with datasets exhibiting rare and OOD states Sec 5.2? 2. Does ExID generalize to OOD states covered by $\mathcal{D}$ Sec 5.4? 3. What is the effect of varying k, λ and updating π_t^ω Sec 5.5? 4. How does performance vary with the quality of $\mathcal{D}$ Sec 5.6?*

5.1 Experimental Setting

We evaluate our methodology on open-AI gym [5], MiniGrid [6] and *real sales promotion (SP) [30]* offline data sets. All our data sets are generated using standard methodologies defined in [32, 31] *except SP which is generated by human operators.* All experiments have been conducted on a Ubuntu 22.04.2 LTS system with 1 NVIDIA K80 GPU, 4 CPUs, and 61GiB RAM. App. F notes the hyperparameter values and network architectures.

Dataset: We experiment on three types of data sets. *Expert Data-set* [10, 16, 22] generated using an optimal policy without any exploration with high trajectory quality but low state action coverage. *Replay Data-set* [2, 13] generated from a policy while training it online, exhibiting a mixture of multiple behavioral policies with high trajectory quality and state action coverage. *Noisy Data-set* [12, 13, 22, 16] generated using an optimal policy that also selects random actions with ϵ greedy strategy where $\epsilon = 0.2$ having low trajectory quality and high state action coverage. Additionally we also experiment on human generated dataset for sales promotion task.

Baselines: We do comparative studies on 10 baselines for OpenAI gym datasets. The first baseline simply checks the conditions of $\mathcal{D}$ and applies corresponding actions in execution. The performance of this baseline shows that $\mathcal{D}$ is imperfect and does not achieve the optimal reward. CQL SE is from [40] where the expert is replaced by $\mathcal{D}$. The other baselines are an ensemble of $\mathcal{D}$ and eight algorithms popular in the Offline RL literature for discrete environments. These algorithms include Behavior Cloning (BC) [29], Behaviour Value Estimation (BVE) [16], Quantile Regression DQN (QRDQN) [7], REM, MCE, BCQ, CQL and Critic Regularized Regression Q-Learning (CRR) [41]. *For a fair comparison, we use actions from domain knowledge for states not in the buffer and actions from the trained policy for other states to obtain the final reward.* Hence, each algorithm is renamed with the suffix D in Table 5.1.

Limiting Data: To create limited-data settings for benchmark datasets, we first extract a small percentage of samples from the full dataset and remove some of the samples based on state conditions. This is done to ensure the reduced buffer satisfies the conditions defined in Def 4.1. We describe the specific conditions of removal in the next section. Further insights and the state visualizations for selected reduced datasets are in App H. **Note : no data reduction has been performed on SP dataset to demonstrate a real dataset exhibits characteristics of reduced buffer**.

5.2 Performance across Different Datasets

Our results for OpenAI gym environments are summarised in Table 5.1 and Minigrid in Table 3 (App D). We observe the performance of offline RL algorithms degrades substantially when part of the data is not seen and trajectory ratios change. For these cases with only 10% partial data, ExID surpasses the performance by at least 27% in the presence of reasonable domain knowledge. The proposed method performs strongest on the replay dataset where the contribution of $L_r(\theta)$ is significant due to state coverage, and the teacher learns from high-quality trajectories. Environment details are described in the App. D. All domain knowledge trees are shown in the App. D Fig 10. We describe limiting data conditions and domain knowledge specific to the environment as follows:

Mountain Car Environment: [27] We use simple, intuitive domain knowledge in this environment shown in the App. D Fig 10 (c), which represents taking a left action when the car is at the bottom of the valley with low velocity to gain momentum; otherwise, taking the right action to drive the car up. Fig 6 (c) shows the state action pairs this rule generates on states sampled from a uniform random distribution over the state boundaries. It can be observed that the states of $\mathcal{D}$ cover part of the missing

Table 1: Average reward [↑] obtained during online evaluation over 3 seeds on openAI gym envs

Env Data	Data Type	$\mathcal{D}$	QRDQN D	REM D	BVE D	CRR D	MCE D	BC D	BCQ D	CQL D	CQL SE	CQL (Full)	**ExID (ours)**
Mountain Car	Expert	-159.9 ± 52.28	-168.2 ± 33.71	-147.7 ± 21.54	-175.36 ± 25.16	-157.2 ± 39.09	-152 ± 37.41	-181.38 ± 28.60	-172.9 ± 27.5	-167.49 ± 12.3	-161.33 ± 18.57	-128.63 ± 10.94	-125.5 ± 2.60
	Replay		-137.14 ± 39.27	-136.26 ± 40.15	-152.0 ± 35.06	-137.23 ± 42.79	-139.91 ± 40.01	-137.26 ± 43.04	-136.29 ± 36.15	-140.38 ± 33.58	-150.67 ± 16.68	-135.4 ± 3.74	-105.79 ± 11.38
	Noisy		-141.61 ± 33.04	-134.99 ± 32.60	-173.95 ± 39.60	-178.99 ± 23.58	-168.69 ± 38.78	-140.0 ± 28.5	-144.52 ± 43.04	-179.8 ± 29.99	-126.96 ± 17.84	-107.06 ± 12.73	-109.9 ± 13.45
Cart Pole	Expert	57.0 ± 5.35	33.23 ± 3.17	41.31 ± 8.76	16.16 ± 9.41	15.24 ± 5.62	16.1 ± 4.4	225.76 ± 74.39	165.36 ± 15.01	121.8 ± 14.0	155.78 ± 26.47	364.1 ± 22.15	307.18 ± 137.72
	Replay		149.09 ± 14.05	180.70 ± 62.79	11.1 ± 2.13	11.24 ± 2.71	9.16 ± 0.25	144.43 ± 2.41	144.76 ± 6.04	131.97 ± 23.23	113.37 ± 5.88	250.02 ± 55.02	340.26 ± 30.58
	Noisy		161 ± 6.40	15.33 ± 0.58	11.53 ± 3.77	13.68 ± 7.49	10.66 ± 2.04	68.4 ± 14.67	63.53 ± 14.08	92.6 ± 22.05	92.6 ± 22.05	93.72 ± 37.79	228.61 ± 38.64
Lunar Lander	Expert	52.48 ± 26.51	5.14 ± 25.10	-184.84 ± 26.45	-681.67 ± 34.86	8.79 ± 25.38	19.71 ± 10.52	38.40 ± 23.21	-45.99 ± 30.47	65.43 ± 71.37	53.22 ± 78.85	167.74 ± 29.4	161.34 ± 17.10
	Replay		-444.20 ± 12.20	-556.81 ± 21.39	-572 ± 27.93	-131.21 ± 31.97	-115.23 ± 18.16	136.63 ± 12.40	111.47 ± 54.67	61.83 ± 45.57	87.70 ± 18.20	187.72 ± 25.62	156.03 ± 56.67
	Noisy		-4.81 ± 97.28	21.41 ± 14.71	28.65 ± 12.26	-158.27 ± 7.71	-50.47 ± 15.78	98.62 ± 28.01	101.59 ± 30.83	5.01 ± 128.63	40.35 ± 65.72	111 ± 52.32	163.57 ± 49.24

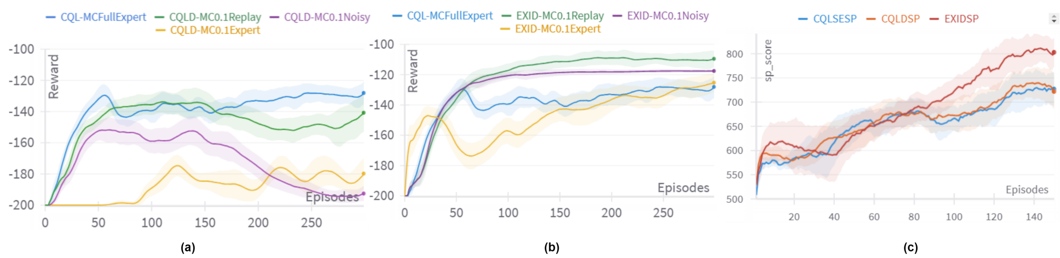


Figure 3: Performance of (a) CQL and (b) EXID on all datasets for Mountain Car during online evaluation (c) Evaluation curves for the sales promotion dataset

data in Fig 1 (a). For limiting datasets, we remove states with position > -0.8. The performance of CQLD and ExID are shown in Fig 3 (a),(b) where ExID surpasses CQLD for all three datasets.

Cart-pole Environment: For this environment, we use domain knowledge from [33], which aims to move in the direction opposite to the lean of the pole, keeping the cart close enough to the center. If the cart is close to an edge, the domain knowledge attempts to account for the cart's velocity and recenter the cart. The full tree is given in the App. D Fig 10 (a). We remove states with cart velocity > -1.5 to create the reduced buffer.

Lunar-Lander Environment: We borrow the decision nodes from [34] and get actions from a sub-optimal policy trained online with an average reward of 52.48. The full set of decision nodes is shown in the App. D Fig 10 (b). $\mathcal{D}$ focuses on keeping the lander balanced when the lander is above ground. When the lander is near the surface, $\mathcal{D}$ focuses on keeping the y velocity lower. To create the reduced datasets, we remove data of lander angle < -0.04.

Mini-Grid Environments: For our experiments, we choose two environments: Random Dynamic Obstacles 6X6 and LavaGapS 7X7. We use intuitive domain knowledge which avoids crashing into obstacles in front, left, or right of agent ref. App. D Fig 10 (d), (e). We remove states with obstacles on the right for creating limited data settings. Due to limitation of space we report the results of the best-performing algorithms on the replay dataset in Table 3 (App D).

5.3 Case study on real human generated Sales Promotion (SP) dataset

SP dataset and environment [30] simulates a real-world sales promotion platform. The number of coupons and the discount the user received will affect his behavior. A higher discount will promote the sales, but the cost will also increase. The goal for the platform operator is to maximize the total profit. The horizon of the dataset is 50 days for the training and 30 days for the test. Domain knowledge ([30], App A] : Active users can be given more coupons with lower discount to maximize profit. We model this as $order_{number} > 60 \wedge Avg_{fee} > 0.8 \implies [5, 0.95]$ where action 1 is number of coupons range [0,5] and action 2 is coupon value (discount value = (1-coupon value)) range [0.6,0.95]. The dataset exhibits the properties in Def 4.1 as first 50 days of sales does not contain many active users as reported in the coverage column of Tab 2 depicting scarcity. The domain rule is imperfect as coupon value and number depend on multiple factors such as user purchase history and behavior. As illustrated in the table 2 and Fig 3 (c) the intuitive domain rule enhances performance by 10.49% in the real dataset.

Table 2: Results on human generated Sales Promotion dataset

Dataset	$\mathcal{D}$	coverage $\mathcal{D}$	CQL + $\mathcal{D}$	CQLSE	EXID	Performance gain
Sales Promotion	654.68 $\pm$ 20.06	20.32%	722.06 $\pm$ 71.40	727.03 $\pm$ 49.56	802.91 $\pm$ 41.69	10.49%

5.4 Generalization to OOD states and contribution of $\mathcal{L}_r(\theta)$

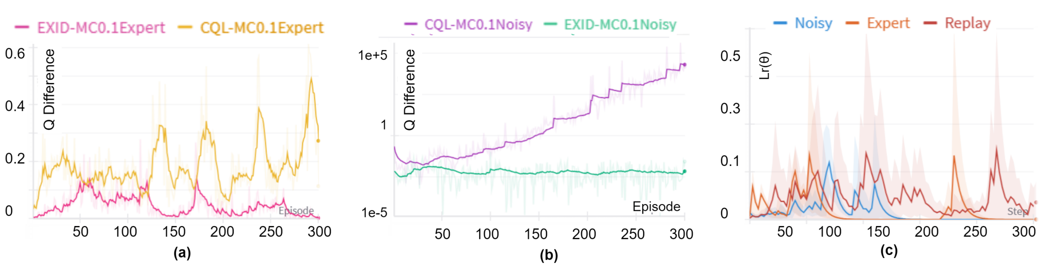


Figure 4: Q value difference between CQL and EXID for expert and policy action on states not present in the buffer for a) expert b) noisy in log scale c) contribution of $\mathcal{L}_r(\theta)$

In Fig 4 (a), (b), we plot $Q^{\theta}_s(s, a_{expert}) - Q^{\theta}_s(s, a_{\theta})$ for CQL and EXID policies for different datasets of Mountain-Car environments. Action a_{expert} is obtained from the full expert dataset where position > -0.8. We observe that the Q value for actions of CQL policy diverges from the expert policy actions with high values for the states not in the reduced buffer, whereas ExID stays close to the expert actions for the unseen states. This empirically shows generalization to OOD states not in the dataset but covered by domain knowledge. In Fig 4 (d), we plot the contribution by $\mathcal{L}_r(\theta)$ during the training and observe the contribution is higher for replay data sets with more state coverage.

5.5 Performance on varying λ, k, and ablation of π^{ω}_t

We study the effect of varying λ on the algorithm for the given domain knowledge. We empirically observe setting a high or a low λ can yield sub-optimal performance, and $\lambda = 0.5$ generally gives good performance. In Fig 5 (a), we show this effect for LunarLander. Plots for other environments are in the App. G Fig 11. For k we observe setting the warm start parameter to 0 yields a sub-optimal policy, as the critic may update π^{ω}_t without completely learning from it. The starting performance increases with an increase in k as shown in Fig 5 (b) for LunarLander. $k = 30$ works best according to empirical evaluations. Plots for other environments are in the App. G Fig 12. We show two ablations for Cart-pole in Fig 5 (c) with no teacher update after the warm start and no inclusion of $\mathcal{L}_r(\theta)$ after the warm start. The warm start in this environment is set to 30 episodes. Fig 5 c) shows without teacher updated, the sub-optimal teacher drags down the performance of the policy beyond the warm start, exhibiting the necessity of π^{ω}_t update. Also, the student converges to a sub-optimal policy if no $\mathcal{L}_r(\theta)$ is included beyond the warm start.

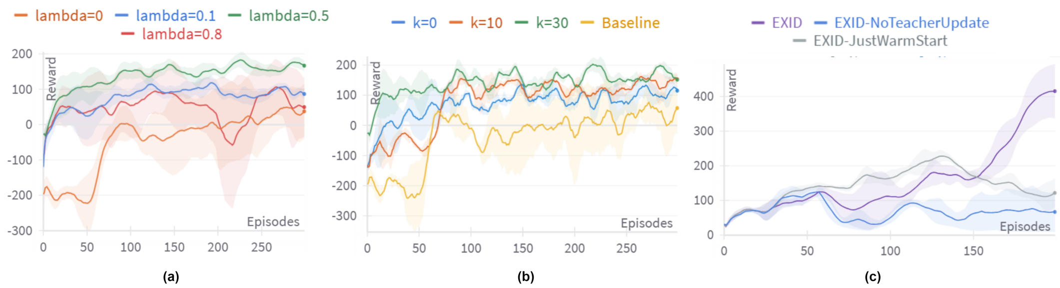


Figure 5: (a) Effect of different λ on the performance of ExID on Lunar Lander (b) Effect of different k on the performance of EXID on Lunar Lander (c) Performance of EXID with teacher update, no teacher update, and just warm start on Cart-pole.

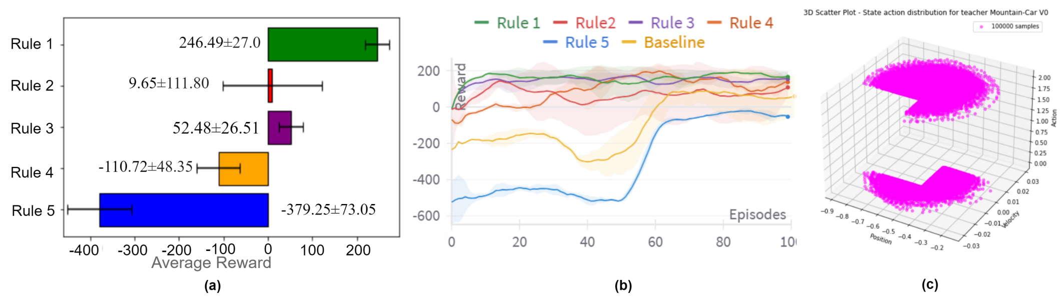


Figure 6: (a) $\mathcal{D}$ with different average rewards (b) Performance effect on Lunar-lander (c) State distribution generated for training the teacher network for mountain-car

5.6 Effect of varying $\mathcal{D}$ quality

We show the effect of choosing policies as $\mathcal{D}$ with different average rewards for Lunar-Lander expert data in Fig 6 (a) and (b). Rule 1 is optimal and has almost the same effect as Rule 3, which is the $\mathcal{D}$ used in our experiments exhibiting that updating a sub-optimal $\mathcal{D}$ can lead to equivalent performance as optimal $\mathcal{D}$. Using a rule with high uncertainty, as Rule 2, induces high uncertainty in the learned policy but performs slightly better than the baseline. Rule 4, which has a lower average reward, also causes gains on average performance with slower convergence. Finally, Rule 5, with very bad actions, affects policy performance adversely and leads to a performance lower than baseline CQL.

6 Conclusion and Limitation

In this paper, we study the effect of limited and partial data on offline RL and observe that the performance of SOTA offline RL algorithms is sub-optimal in such settings. The paper proposes a methodology to handle offline RL's performance degradation using domain insights. We incorporate a regularization loss in the CQL training using a teacher policy and refine the initial teacher policy while training. We show that incorporating reasonable domain knowledge in offline RL enhances performance, achieving a performance close to full data. However, this method is limited by the quality of the domain knowledge and the overlap between domain knowledge states and reduced buffer data. The study is also limited to discrete domains. In the future, the authors would like to improve on capturing domain knowledge into the policy network without dependence on data and extending the methodology to algorithms that handle continuous action space.

7 Broader Impact

During the trial-and-error training phase, RL agents may exhibit irrational behavior, which can be risky and costly in real-world scenarios. As a more practical alternative to online RL, offline RL utilizes pre-existing collected data to eliminate the need for real-time interactions during training. However, a drawback of offline RL is its dependence on the quality and quantity of historical data, which, when sub-optimal, could adversely affect overall performance. Therefore, through this work, we use domain knowledge to suppress erroneous actions when available data is limited. However, this inclusion may facilitate harmful behavior in the presence of biased domain knowledge. Therefore, we advocate the use of well-regulated domain knowledge obtained from experts. Beyond this, we do not foresee any ethical impact on our work.

References

[1] A survey on offline reinforcement learning: Taxonomy, review, and open problems. *IEEE Transactions on Neural Networks and Learning Systems*, 2023. ISSN 21622388. doi: 10.1109/TNNLS.2023.3250269.

[2] Rishabh Agarwal, Dale Schuurmans, and Mohammad Norouzi. An optimistic perspective on offline reinforcement learning. In *International Conference on Machine Learning*, pages 104–114. PMLR, 2020.

[3] Gaon An, Seungyong Moon, Jang-Hyun Kim, and Hyun Oh Song. Uncertainty-based offline reinforcement learning with diversified q-ensemble. *Advances in neural information processing systems*, 34:7436–7447, 2021.

[4] Peter L. Bartlett and Ambuj Tewari. Regal: A regularization based algorithm for reinforcement learning in weakly communicating mdps. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, UAI '09, page 35–42, Arlington, Virginia, USA, 2009. AUAI Press. ISBN 9780974903958.

[5] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym. *arXiv preprint arXiv:1606.01540*, 2016.

[6] Maxime Chevalier-Boisvert, Bolun Dai, Mark Towers, Rodrigo de Lazcano, Lucas Willems, Salem Lahlou, Suman Pal, Pablo Samuel Castro, and Jordan Terry. Minigrid & miniworld: Modular & customizable reinforcement learning environments for goal-oriented tasks. *CoRR*, abs/2306.13831, 2023.

[7] Will Dabney, Mark Rowland, Marc Bellemare, and Rémi Munos. Distributional reinforcement learning with quantile regression. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

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

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

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

[12] Scott Fujimoto, Edoardo Conti, Mohammad Ghavamzadeh, and Joelle Pineau. Benchmarking batch deep reinforcement learning algorithms. *arXiv preprint arXiv:1910.01708*, 2019.

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

[14] Yarin Gal and Zoubin Ghahramani. A theoretically grounded application of dropout in recurrent neural networks. *Advances in neural information processing systems*, 29, 2016.

[15] Sinong Geng, Aldo Pacchiano, Andrey Kolobov, and Ching-An Cheng. Improving offline rl by blending heuristics. *arXiv preprint arXiv:2306.00321*, 2023.

[16] Caglar Gulcehre, Sergio Gómez Colmenarejo, Ziyu Wang, Jakub Sygnowski, Thomas Paine, Konrad Zolna, Yutian Chen, Matthew Hoffman, Razvan Pascanu, and Nando de Freitas. Regularized behavior value estimation. *arXiv preprint arXiv:2103.09575*, 2021.

[17] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[18] Zhiting Hu, Xuezhe Ma, Zhengzhong Liu, Eduard Hovy, and Eric Xing. Harnessing deep neural networks with logic rules. *arXiv preprint arXiv:1603.06318*, 2016.

[19] Sham Kakade and John Langford. Approximately optimal approximate reinforcement learning. In *Proceedings of the Nineteenth International Conference on Machine Learning*, pages 267–274, 2002.

[20] Ilya Kostrikov, Rob Fergus, Jonathan Tompson, and Ofir Nachum. Offline reinforcement learning with fisher divergence critic regularization. In *International Conference on Machine Learning*, pages 5774–5783. PMLR, 2021.

[21] Aviral Kumar, Justin Fu, Matthew Soh, George Tucker, and Sergey Levine. Stabilizing off-policy q-learning via bootstrapping error reduction. *Advances in Neural Information Processing Systems*, 32, 2019.

[22] Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative q-learning for offline reinforcement learning. *Advances in Neural Information Processing Systems*, 33:1179–1191, 2020.

[23] Sergey Levine, Aviral Kumar, G. Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *ArXiv*, abs/2005.01643, 2020. URL `https://api.semanticscholar.org/CorpusID:218486979`.

[24] Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*, 2020.

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

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

[27] Andrew William Moore. Efficient memory-based learning for robot control. Technical report, University of Cambridge, Computer Laboratory, 1990.

[28] Susan A Murphy. A generalization error for q-learning. 2005.

[29] Dean A Pomerleau. Efficient training of artificial neural networks for autonomous navigation. *Neural computation*, 3(1):88–97, 1991.

[30] Rong-Jun Qin, Xingyuan Zhang, Songyi Gao, Xiong-Hui Chen, Zewen Li, Weinan Zhang, and Yang Yu. Neorl: A near real-world benchmark for offline reinforcement learning. *Advances in Neural Information Processing Systems*, 35:24753–24765, 2022.

[31] Kajetan Schweighofer, Markus Hofmarcher, Marius-Constantin Dinu, Philipp Renz, Angela Bitto-Nemling, Vihang Prakash Patil, and Sepp Hochreiter. Understanding the effects of dataset characteristics on offline reinforcement learning. In *Deep RL Workshop NeurIPS 2021*, 2021. URL `https://openreview.net/forum?id=A4EWtf-TO3Y`.

[32] Kajetan Schweighofer, Marius-constantin Dinu, Andreas Radler, Markus Hofmarcher, Vihang Prakash Patil, Angela Bitto-Nemling, Hamid Eghbal-zadeh, and Sepp Hochreiter. A dataset perspective on offline reinforcement learning. In *Conference on Lifelong Learning Agents*, pages 470–517. PMLR, 2022.

[33] Andrew Silva and Matthew Gombolay. Encoding human domain knowledge to warm start reinforcement learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 5042–5050, 2021.

[34] Andrew Silva, Matthew Gombolay, Taylor Killian, Ivan Jimenez, and Sung-Hyun Son. Optimization methods for interpretable differentiable decision trees applied to reinforcement learning. In *International conference on artificial intelligence and statistics*, pages 1855–1865. PMLR, 2020.

[35] Samarth Sinha, Ajay Mandlekar, and Animesh Garg. S4rl: Surprisingly simple self-supervision for offline reinforcement learning in robotics. In Aleksandra Faust, David Hsu, and Gerhard Neumann, editors, *Proceedings of the 5th Conference on Robot Learning*, volume 164 of *Proceedings of Machine Learning Research*, pages 907–917. PMLR, 08–11 Nov 2022.

[36] Kihyuk Sohn, Zizhao Zhang, Chun-Liang Li, Han Zhang, Chen-Yu Lee, and Tomas Pfister. A simple semi-supervised learning framework for object detection. *arXiv preprint arXiv:2005.04757*, 2020.

[37] Jiaxi Tang and Ke Wang. Ranking distillation: Learning compact ranking models with high performance for recommender system. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 2289–2298, 2018.

[38] Raphael Tang, Yao Lu, Linqing Liu, Lili Mou, Olga Vechtomova, and Jimmy Lin. Distilling task-specific knowledge from bert into simple neural networks. *arXiv preprint arXiv:1903.12136*, 2019.

[39] Wei-Cheng Tseng, Tsun-Hsuan Johnson Wang, Yen-Chen Lin, and Phillip Isola. Offline multi-agent reinforcement learning with knowledge distillation. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 226–237. Curran Associates, Inc., 2022.

[40] Richa Verma, Durgesh Kalwar, Harshad Khadilkar, and Balaraman Ravindran. Guiding offline reinforcement learning using a safety expert. In *Proceedings of the 7th Joint International Conference on Data Science & Management of Data (11th ACM IKDD CODS and 29th COMAD)*, pages 82–90, 2024.

[41] Ziyu Wang, Alexander Novikov, Konrad Zolna, Josh S Merel, Jost Tobias Springenberg, Scott E Reed, Bobak Shahriari, Noah Siegel, Caglar Gulcehre, Nicolas Heess, et al. Critic regularized regression. *Advances in Neural Information Processing Systems*, 33:7768–7778, 2020.

[42] Yue Wu, Shuangfei Zhai, Nitish Srivastava, Joshua M. Susskind, Jian Zhang, Ruslan Salakhutdinov, and Hanlin Goh. Uncertainty weighted actor-critic for offline reinforcement learning. In *International Conference on Machine Learning*, 2021. URL `https://api.semanticscholar.org/CorpusID:234763307`.

[43] Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V Le. Self-training with noisy student improves imagenet classification. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10687–10698, 2020.

[44] Qisen Yang, Shenzhi Wang, Matthieu Gaetan Lin, Shiji Song, and Gao Huang. Boosting offline reinforcement learning with action preference query. In *International Conference on Machine Learning*, pages 39509–39523. PMLR, 2023.

[45] Li Yuan, Francis EH Tay, Guilin Li, Tao Wang, and Jiashi Feng. Revisiting knowledge distillation via label smoothing regularization. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3902–3910, 2020. doi: 10.1109/CVPR42600.2020.00396.

[46] Xiaoxuan Zhang and S Zhang Y Yu. Domain knowledge guided offline q learning. In *Second Offline Reinforcement Learning Workshop at Neurips*, volume 2021, 2021.

[47] Ying Zheng, Haoyu Chen, Qingyang Duan, Lixiang Lin, Yiyang Shao, Wei Wang, Xin Wang, and Yuedong Xu. Leveraging domain knowledge for robust deep reinforcement learning in networking. In *IEEE INFOCOM 2021 - IEEE Conference on Computer Communications*, pages 1–10, 2021. doi: 10.1109/INFOCOM42981.2021.9488863.

A Theoretical Analysis

Notations

For any deterministic policy π the performance return is formulated as $\eta(\pi) = \mathbb{E}_{\tau\sim\pi}[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)]$

For any policy π, ρ_π is the (unormalized) discounted visitation frequency given by $\rho_\pi(s) = \sum_{t=0}^{\infty} \gamma^t P(s_t = s)$ where $s_0 \sim \rho^0(s_0)$ and the trajectory $(s_0, s_1, \dots)$ is sampled from the policy π and $\rho_\pi(s) \in [0, \frac{1}{1-\gamma}]$. $\bar{\rho}_\pi(s) = sup\{\rho_\pi(s), s \in S\} \in [\frac{1}{|S_\pi|(1-\gamma)}, \frac{1}{(1-\gamma)}]$

We denote the regularized policy learned by ExID on $\mathcal{B}_r$ as $\hat{\pi}$ and the unregularized policy as π_u.

Lemmas

We introduce the following Lemma required for our theoretical analysis.

Lemma A.1. *([44]) Given two policies π_1 and π_2*

$$\eta(\pi_1) - \eta(\pi_2) = \int_{s\in S} \rho_{\pi_1}(s)(Q^*(s, \pi_1(s) - V^*(s))ds - \int_{s\in S} \rho_{\pi_2}(s)(Q^*(s, \pi_2(s) - V^*(s))ds$$

Proof. Please refer to Lemma A.1 Eq 17 in [44] □

Proposition A.2. *(4.2) Denote $\hat{\pi}$ as the policy learned by ExID, π_u as any offline RL policy learned on $\mathcal{B}_r$ and optimal Q function as Q^* and V function as V^*. Then it holds that*

$$\eta(\hat{\pi}) - \eta(\pi_u) \geq \mathbb{E}_{s\sim O|\pi_u}[V^*(s) - Q^*(s, \pi_u(s))] - \bar{\rho}_{\hat{\pi}}\alpha$$

Proof. According to [19] performance improvement between two policies if given by

$$\eta(\pi_1) = \eta(\pi_2) + \mathbb{E}_{\tau\sim\pi_1}\left[\sum_{t=0}^{\infty} \gamma^t Q_{\pi_2}(s_t, a_t) - V_{\pi_2}(s_t)\right] \tag{8}$$

Replacing π_1 by $\hat{\pi}$ and π_2 by π_u and by following Lemma A.1

$$\eta(\hat{\pi}) - \eta(\pi_u) = \int_{s\in S} \rho_{\hat{\pi}}(s)(Q^*(s, \hat{\pi}(s)) - V^*(s))ds - \int_{s\in S} \rho_{\pi_u}(s)(Q^*(s, \pi_u(s)) - V^*(s))ds \tag{9}$$

$$= \int_{s\in S} \rho_{\pi_u}(s)(V^*(s) - Q^*(s, \pi_u(s)))ds - \int_{s\in S} \rho_{\hat{\pi}}(s)(V^*(s) - Q^*(s, \hat{\pi}(s)))ds \tag{10}$$

Dividing the state space into in dataset domain states (I) and OOD states (O). The

$$\underbrace{\left[\int_{s\in I} \rho_{\pi_u}(s)(V^*(s) - Q^*(s, \pi_u(s)))ds - \int_{s\in I} \rho_{\hat{\pi}}(s)(V^*(s) - Q^*(s, \hat{\pi}(s)))ds\right]}_{a} + \tag{11}$$

$$\underbrace{\left[\int_{s\in O} \rho_{\pi_u}(s)(V^*(s) - Q^*(s, \pi_u(s)))ds - \int_{s\in O} \rho_{\hat{\pi}}(s)(V^*(s) - Q^*(s, \hat{\pi}(s)))ds\right]}_{b} \tag{12}$$

Since the regularization loss facilitates visitation to OOD states via knowledge distillation we assume $\rho_{\hat{\pi}} = \rho_{\pi_u} - \Delta_i$ for $s \in i$ and $\rho_{\hat{\pi}} = \rho_{\pi_u} + \Delta_o$ for $s \in o$ where $\Delta_i \in [0, \rho_{\pi_u(s)}]$ and $\Delta_o \in [0, \frac{1}{1-\gamma} - \rho_{\pi_u(s)}]$

$$a = \int_{s\in I} \rho_{\pi_u}(s)(V^*(s) - Q^*(s, \pi_u(s)))ds - \int_{s\in I} (\rho_{\pi_u} - \Delta_i)(s)(V^*(s) - Q^*(s, \hat{\pi}(s)))ds \tag{13}$$

$$= \int_{s\in I} \rho_{\pi_u}(s)(Q^*(s, \hat{\pi}(s)) - Q^*(s, \pi_u(s)))ds + \int_{s\in I} \Delta_i(s)(V^*(s) - Q^*(s, \hat{\pi}(s)))ds \tag{14}$$

Under assumption in distribution action can be learned from the dataset due to conservatism of offline RL $(Q^*(s,\hat{\pi}(s)) - Q^*(s,\pi_u(s))) \approx 0$, $a \geq 0$

$$b = \int_{s\in O} \rho_{\pi_u}(s)(V^*(s) - Q^*(s,\pi_u(s)))ds - \int_{s\in O} (\rho_{\pi_u} + \Delta_o)(s)(V^*(s) - Q^*(s,\hat{\pi}(s)))ds \tag{15}$$

$$\geq \int_{s\in O} \rho_{\pi_u}(s)(V^*(s) - Q^*(s,\pi_u(s)))ds - \int_{s\in O} \rho_{\hat{\pi}}(s)(V^*(s) - Q^*(s,\hat{\pi}(s)))ds \tag{16}$$

$$\geq \mathbb{E}_{s\sim O|\pi_u}[V^*(s) - Q^*(s,\pi_u(s))] - \mathbb{E}_{s\sim O|\hat{\pi}}[V^*(s) - Q^*(s,\hat{\pi}(s))] \tag{17}$$

Further loosening the lower bound

$$= \mathbb{E}_{s\sim O|\pi_u}[V^*(s) - Q^*(s,\pi_u(s))] - \bar{\rho}_{\hat{\pi}} \int_{s\in O} \frac{\rho_{\hat{\pi}}}{\bar{\rho}_{\hat{\pi}}}(V^*(s) - Q^*(s,\hat{\pi}(s)))ds \tag{18}$$

$$\geq \mathbb{E}_{s\sim O|\pi_u}[V^*(s) - Q^*(s,\pi_u(s))] - \bar{\rho}_{\hat{\pi}} \int_{s\in O} (V^*(s) - Q^*(s,\hat{\pi}(s)))ds \tag{19}$$

Combining Eq 14, 17 and 19, and denoting $\alpha = \mathbb{E}_{s\sim O}[V^*(s) - Q^*(s,\hat{\pi}(s))]$

$$\eta(\hat{\pi}) - \eta(\pi_u) \geq \mathbb{E}_{s\sim O|\pi_u}[V^*(s) - Q^*(s,\pi_u(s))] - \bar{\rho}_{\hat{\pi}}\alpha \tag{20}$$

Hence, Proposition 4.2 follows **Q.E.D**

□

Proposition A.3. *(4.3) Algo 1 reduces generalization error if $Q^*(s,\pi_t^\omega(s)) > Q^*(s,\pi(s))$ for $s \in \mathcal{D} \cap \mathcal{B}_r$, where π is vanilla offline RL policy learnt on $\mathcal{B}_r$.*

Proof. Generalization error for any policy π as defined by [28] can be written as:

$$G_\pi = V^*(s_0) - V_\pi(s_0) = -\mathbb{E}_{\tau\sim\pi}[\sum_{t=0}^{T} \gamma^t Q^*(s_t,\pi(s_t)) - V^*(s_t)] \tag{21}$$

Here, $\mathbb{E}_{\tau\sim\pi}$ represents sampling trajectories with policy π. Since the state space is continuous, we can represent the expectation as an integral over the state space

$$= -\sum_{t=0}^{T} \gamma^t \int_{s\in S} P(s_t = s|\pi)(Q^*(s_t,\pi(s_t)) - V^*(s_t))ds \tag{22}$$

$$= -\int_{s\in S} \sum_{t=0}^{T} \gamma^t P(s_t = s|\pi)(Q^*(s_t,\pi(s_t)) - V^*(s_t))ds \tag{23}$$

Analysing with respect to $s \in \mathcal{D} \cap \mathcal{B}_r$ we can break the integration into two parts

$$= -\left[\int_{s\in S/D} \sum_{t=0}^{T} \gamma^t P(s_t = s|\pi)(Q^*(s_t,\pi(s_t)) - V^*(s_t))ds + \int_{s\in D} \sum_{t=0}^{T} \gamma^t P(s_t = s|\pi)(Q^*(s_t,\pi(s_t)) - V^*(s_t))\right] \tag{24}$$

$$= -\left[f(s|\pi) + \int_{s\in D} \sum_{t=0}^{T} \gamma^t P(s_t = s|\pi)(Q^*(s_t,\pi(s_t)) - V^*(s_t))\right] \tag{25}$$

For a policy $\hat{\pi}$ learnt in Algo 1 the action for $s_t = s \in \mathcal{D}$ is regularized to be close to π_t^ω which either follows domain knowledge or expert demonstrations. Hence, it is reasonable to assume $Q^*(s_t,\pi_t^\omega(s_t)) > Q^*(s_t,\pi(s_t))$. It follows

$$\int_{s\in D} \sum_{t=0}^{T} \gamma^t P(s_t = s|\hat{\pi})(Q^*(s_t,\hat{\pi}(s_t)) - V^*(s_t)) < \int_{s\in D} \sum_{t=0}^{T} \gamma^t P(s_t = s|\pi)(Q^*(s_t,\pi(s_t)) - V^*(s_t)) \tag{26}$$

Note for $s \notin \mathcal{D}$, $f(s|\hat{\pi}) \approx f(s|\pi)$. This is because the regularization term assigns max Q value to a different action for $s \in \mathcal{D}$ but $max_a(Q(s,a))$ remains same

$$\therefore -\left[f(s|\hat{\pi}) + \int_{s\in D}\sum_{t=0}^{T}\gamma^t P(s_t = s|\hat{\pi})(Q^*(s_t,\hat{\pi}(s_t)) - V^*(s_t))\right]$$
$$< -\left[f(s|\pi) + \int_{s\in D}\sum_{t=0}^{T}\gamma^t P(s_t = s|\pi)(Q^*(s_t,\pi(s_t)) - V^*(s_t))\right] \tag{27}$$

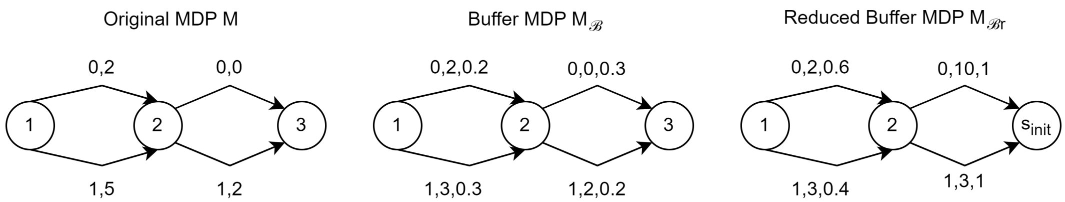
Hence, $G_{\hat{\pi}} < G_{\pi}$ Proposition 2 follows **Q.E.D**

□

B Missing Examples

Performing $Q-Learning$ by sampling from a reduced batch $\mathcal{B}_r$ may not converge to an optimal policy for the MDP $M_{\mathcal{B}}$ representing the full buffer.

Example (Theorem 1,[13]) defines MDP $M_{\mathcal{B}}$ of $\mathcal{B}$ from same state action space of the original MDP M with transition probabilities $p_{\mathcal{B}}(s'|s,a) = \frac{N(s,a,s')}{\sum_{\tilde{s}} N(s,a,\tilde{s})}$ where $N(s,a,s')$ is the number of times (s,a,s') occurs in $\mathcal{B}$ and an terminal state s_{init}. It states $p_{\mathcal{B}}(s_{init}|s,a) = 1$ when $\sum_{\tilde{s}} N(s,a,\tilde{s}) = 0$. This happens when transitions of some s' of (s,a,s') are missing from the buffer, which may occur in $\mathcal{B}_r$ when $\mathcal{B}_r \subset \mathcal{B}$. $r(s_{init},s,a)$ is initialized to $Q(s,a)$. We assume that a policy learned on reduced dataset $\mathcal{B}_r$ converges to optimal value function and disprove it using the following counterexample:



	0	1
1	7	2
2	10	3
3	0	0

Initialized q table

	0	1
1	4	6
2	0	2
3	0	0

Q table after one step backup for M$_{\mathscr{B}}$

	0	1
1	8	7
2	10	3
sinit	0	0

Q table after one step backup for M$_{\mathscr{B}r}$

Figure 7: Example MDP, sampled buffer MDP and reduced buffer with Q tables

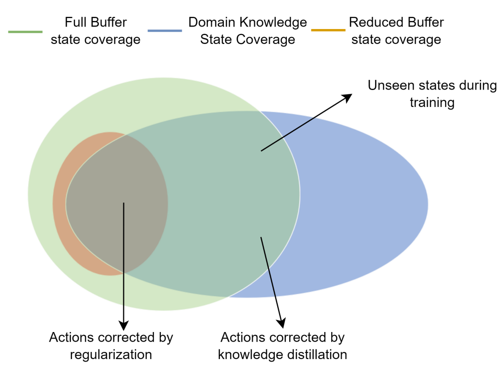


Figure 8: We hypothesize the suboptimal performance of offline RL for limited data can be addressed via domain knowledge via action regularization and knowledge distillation.

We take a simple MDP illustrated in Fig 7 with 3 states and 2 actions (0,1). The reward of each action is marked along the transition. The sampled MDP is constructed the following samples (1,0,2)-2,(1,1,2)-3, (2,0,3)-3, and (2,1,3)-2 and the reduced buffer MDP with samples (1,0,2)-2 and (1,1,2)-1. The probabilities are marked along the transition. It is easy to see that the policy learned under the reduced MDP converges to a nonoptimal policy after one step of the Q table update with $Q(s,a) = r(s,a) + p(s'|s,a) * max_{a'}(Q(s',a'))$. This happens because of transition probability shift on reducing samples $p_{\mathcal{B}}(s'|s,a) \neq p_{\mathcal{B}_r}(s'|s,a)$ and no Q updates for $(s,a) \notin \mathcal{B}_r$.

Our methodology addresses these issues as follows:

- For $s \in D \cap \mathcal{B}_r$ better actions are enforced through regularization using π_t^{ω} even when the transition probabilities are low for optimal transitions.
- Incorporating regularization distills the teacher's knowledge in the critic-enhancing generalization.

A visualization is shown in Fig 8.

C Algorithm

The pseudo code of the algorithm is described in Algo 1.

Algorithm 1 Pseudo code for EXID

Input: Reduced buffer $\mathcal{B}_r$, Initial teacher network π_t^ω, Training steps N, Warm-up steps k, Soft update τ, hyperparameters: λ, α
Initialize Critic with MC dropout and Target Critic $Q_s^\theta, Q_s^{\theta'}$
for $n \leftarrow 1$ **to** N **do**
  Sample mini-batch b of transitions $(s, a, r, s') \sim \mathcal{B}_r$ $a_t = [], a_s = [], s_r = []$
  for $s \in b$ **do**
    if $s \models \mathcal{D}$ and $\pi_t^\omega(s) \neq argmax_a(Q_s^\theta(s, a))$ **then**
      $a_t.append(\pi_t^\omega(s))$
      $a_s.append(\text{argmax}_a(Q_s^\theta(s, a)))$
      $s_r.append(s)$
    end if
  end for
  if $n > k \wedge$ Cond. 6 **then**
    Update $\pi_t^\omega(s)$ using Eq 7
    $\mathcal{L}_r(\theta) = 0$
  else
    Calculate $\mathcal{L}_r(\theta)$ using Eq 3
  end if
  Calculate $\mathcal{L}(\theta)$ using Eq 4
  Update Q_s^θ with $\mathcal{L}(\theta)$ and softy update $Q_s^{\theta'}$ and τ
end for

D Environments and Domain Knowledge Trees

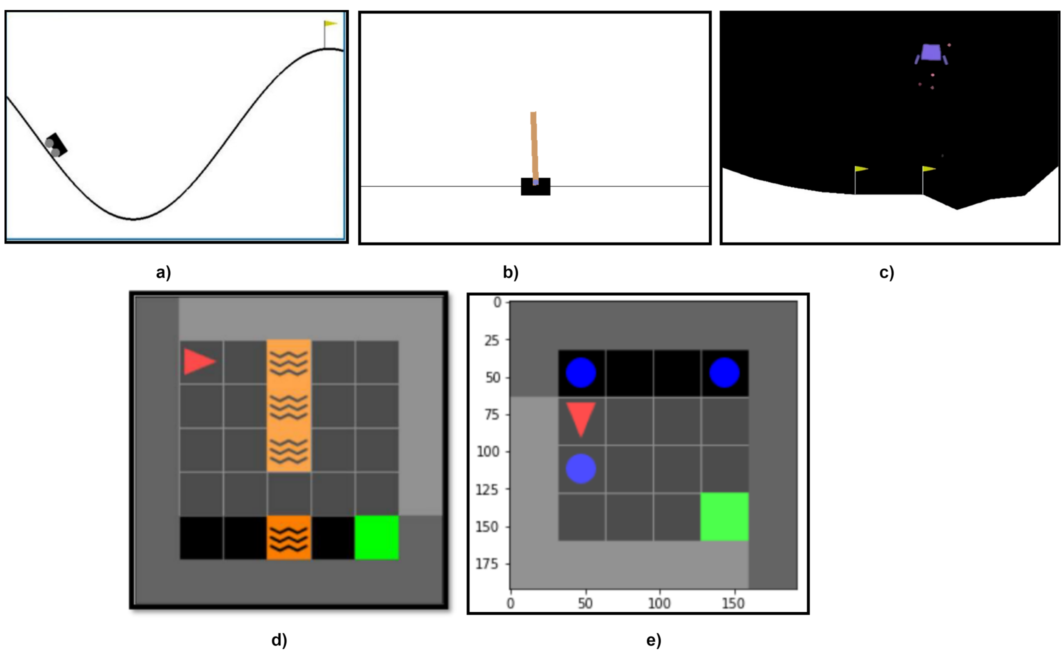


Figure 9: Graphical visualizations of environments used in the experiments. These environments are a) MountainCar-v0 b) CartPole-v1 c) LunarLander-v2 d) MiniGrid-LavaGapS7-v0 e) MiniGrid-Dynamic-Obstacles-Random-6x6-v0

The graphical visualization of each environment is depicted in Fig 9. The choice of environment in this paper depended on two factors: a) Pre-existing standard methods of generating offline RL datasets. b) Possibility of creating intuitive decision tree-based domain knowledge. All datasets have been created via [31]. We explain the environments in detail as follows:

Mountain-car Environment: This environment Fig 9 a) has two state variables, position and velocity, and three discrete actions: left push, right push, and no action [27]. The goal is to drive a car up a valley to reach the flag. This environment is challenging for offline RL because of sparse rewards, which are only obtained on reaching the flag.

Cart-pole Environment The environment Fig 9 b) has 4 states and 2 actions representing left force and right force. The objective is to balance a pole on a moving cart.

Lunar-Lander Environment: The task is to land a lunar rover between two flags Fig 9 c) by observing 8 states and applying one of 4 actions.

Minigrid Environments: Mini-grid [6] is an environment suite containing 2D grid-worlds with goal oriented tasks. As explained in the main text, we experiment using MiniGrid-LavaGapS7-v0 and MiniGrid-Dynamic-Obstacles-Random-6x6-v0 from this environment suite is shown in Fig 9 d) and e). In MiniGrid-LavaGapS7-v0, the agent has to avoid Lava and pass through the gap to reach the goal. Dynamic obstacles are similar; however, the agent can start at a random position and has to avoid dynamically moving balls to reach the goal. The environment has image observation with 3 channels (OBJECT_ID, COLOR_ID, STATE). Following [31] experiments, we flatten the image to an array of 98 observations and restrict action space to three actions: Turn left, Turn Right, and Move forward. The results of minigrid environment are reported in Table 3. Since this environment uses a semantic map from image observation, we collect states from a fixed policy with random actions to generate the teacher's state distribution. CQL on the full dataset achieves the average reward of 0.92 ± 0.1 for DynamicObstacles and 0.53 ± 0.01 for LavaGapS.

The domain knowledge trees for all the environments are shown in Fig 10. The cart pole domain knowledge tree Fig 10 a) is taken from [33] (Fig 7). The Lunar Lander decision nodes Fig 10 b) have been taken from [34] (Fig4). For the mini-grid environments, we construct intuitive decision trees shown in Fig 10 d) and Fig 10 e). Positions 52, 40, and 68 represent positions front, right, and left of the agent. Value 0.2 represents a wall, 0.9 represents Lava, and 0.6 represents a ball. We check positions 52, 40, and 68 for these obstacles and choose the recommended actions as domain knowledge.

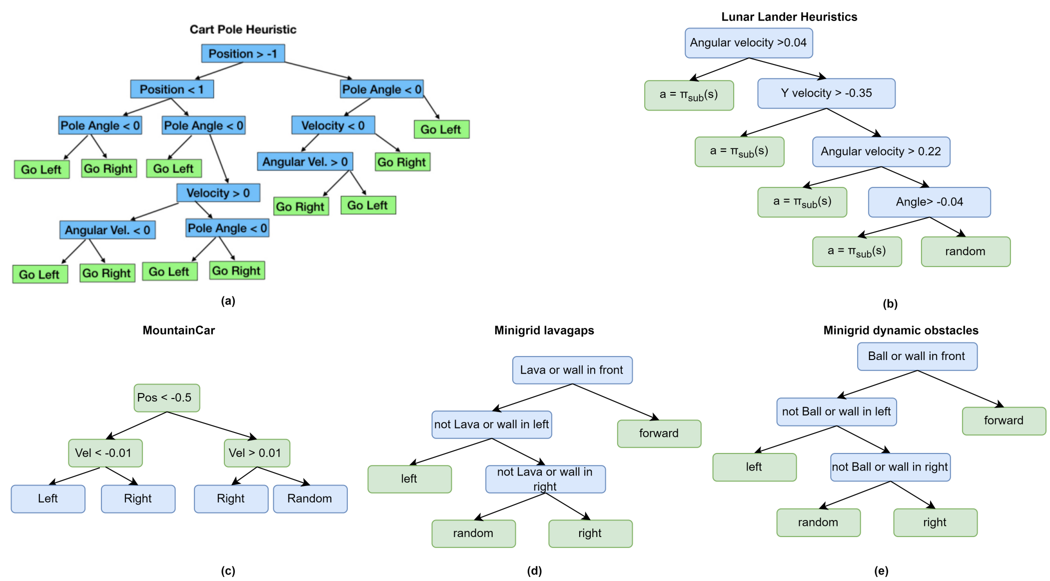


Figure 10: Domain knowledge trees for a) CartPole-v1 b) LunarLander-v2 c) MountainCar-v0 d) MiniGrid-LavaGapS7-v0 e) MiniGrid-Dynamic-Obstacles-Random-6x6-v0 environments

E Related Work: Knowledge Distillation

Knowledge distillation is a well-embraced technique of incorporating additional information in neural networks and has been applied to various fields like computer vision [43, 36], natural language processing [8, 38], and recommendation systems [37]. [17] introduced the concept of distilling knowledge from a complex, pre-trained model (teacher) into a smaller model (student). In recent years, researchers have explored the integration of rule-based regularization techniques within the context of knowledge distillation. Rule regularization introduces additional constraints based on predefined rules, guiding the learning process of the student model

Table 3: Average reward [↑] obtained during online evaluation over 3 seeds on Minigrid environments

ENVIRONMENT	$\mathcal{D}$	BC D	BCQ D	CQL D	**ExID**
MINIGRID DYNAMIC RANDOM6X6	0.50 ± 0.08	0.59 ± 0.07	0.24 ± 0.22	0.14 ± 0.1	0.79 ± 0.07
MINIGRID LAVAGAPS 7X7	0.27 ± 0.09	0.29 ± 0.11	0.26 ± 0.1	0.28 ± 0.12	0.46 ± 0.13

[18, 45]. These techniques have shown to reduce overfitting and enhance generalization [38]. Knowledge distillation is also prevalent in the field of RL [47] and offline RL [39]. Contrary to prevalent teacher-student knowledge distillation techniques, our work does not enforce parameter sharing among the networks. Through experiments, we demonstrate that a simple regularization loss and expected performance-based updates can improve generalization to unobserved states covered by domain knowledge. There are also no constraints on keeping the same network structure for the teacher, paving ways for capturing the domain knowledge into more structured networks such as Differentiable Decision Trees (DDTs).

F Network Architecture and Hyper-parameters

We follow the network architecture and hyper-parameters proposed by [31] for all our networks, including the baseline networks. The teacher BC network π_ω^t and Critic network $Q_s^\theta(s,a)$ consists of 3 linear layers, each having a hidden size of 256 neurons. The number of input and output neurons depends on the environment's state and action size. All layers except the last are SELU activation functions; the final layer uses linear activation. π_ω^t uses a softmax activation function in the last layer for producing action probabilities. A learning rate of 0.0001 with batch size 32 and $\alpha = 0.1$ is used for all environments. MC dropout probability of 0.5 and number of stochastic passes T=10 have been used for the critic network. The uncertainty check is performed every 15 episodes after the warm start to avoid computational overhead. The hyper-parameters specific to our algorithm for OpenAI gym are reported in Table F. The hyper-parameters specific to our algorithm for Minigrid environments are reported in Table 5.

Table 4: Hyperparameters for openAI gym environments

HYPERPARAM	MOUNTAINCAR			CARTPOLE			LUNAR-LANDER		
DATA TYPE	EXPERT	REPLAY	NOISY	EXPERT	REPLAY	NOISY	EXPERT	REPLAY	NOISY
λ	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
k	30	30	30	30	30	30	30	30	30
π_ω^t LR	$1e^5$	$1e^5$	$1e^5$	$1e^2$	$1e^2$	$1e^2$	$1e^4$	$1e^4$	$1e^4$
TRAINING STEPS	42000	36000	36000	30000	17000	17000	18000	18000	18000

Table 5: Hyper-parameters for Mini-grid environments for replay dataset

Environment	DynamicObstRandom6x6-v0	LavaGapS7v0
λ	0.1	0.1
k	30	30
π_ω^t lr	$1e^4$	$1e^4$
training steps	5000	10000

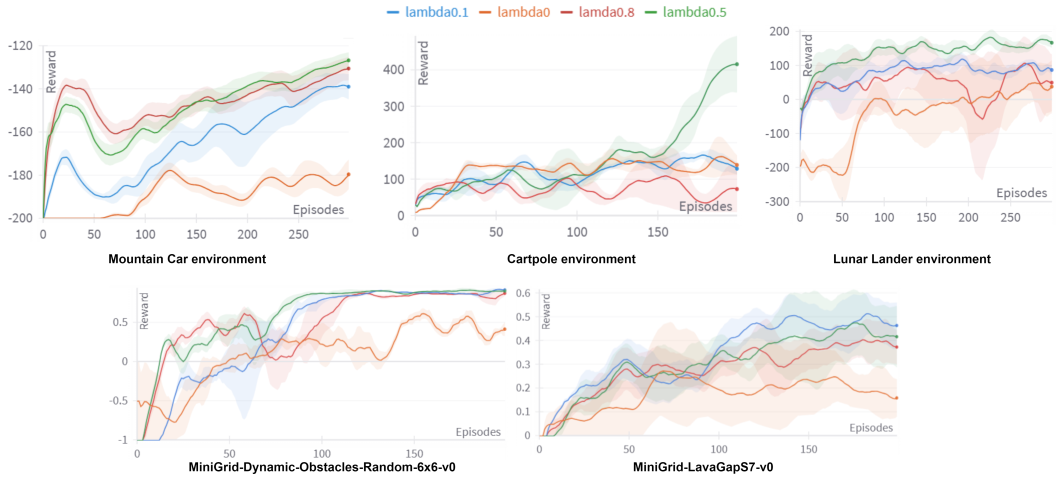


Figure 11: Effect of λ on the performance of ExID for different environments expert datasets.

G Effect of k and λ and Evaluation Plots

We empirically evaluate the effect of λ In Fig 11 and k in Fig 12. We believe these parameters depend on the quality of $\mathcal{D}$. For the given $\mathcal{D}$ in the environments we empirically observe, $\lambda = 0.5$ generally performs well, except for Minigrid environments where $\lambda = 0.1$ works better. Increasing the warm start parameter k generally increases the initial performance of the policy, allowing it to learn from the teacher. Meanwhile, no warm start adversely affects policy performance as the critic may erroneously update the teacher. From empirical evaluation, we observe that $k = 30$ gives a reasonable start to the policy. All the evaluation plots are shown in Fig 13, where it can be observed that ExID performs better than baseline CQL.

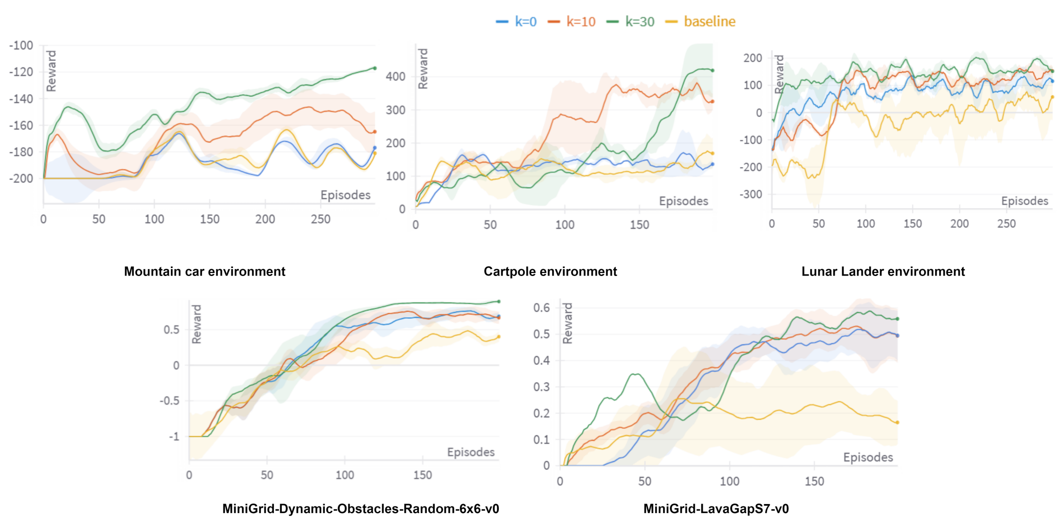


Figure 12: Effect of k on the performance of ExID for different environments expert datasets.

H Data reduction design and data distribution visualization of reduced dataset

In this section, we discuss the intuition behind our data-limiting choices. We also visually represent selected reduced datasets for the OpenAI gym environments.

Figure 13: Evaluation plots of CQL and EXID algorithms for Cartpole, Lunar-Lander, and Minigrid environments using different data types and seeds reported in the main paper Table 5.1.

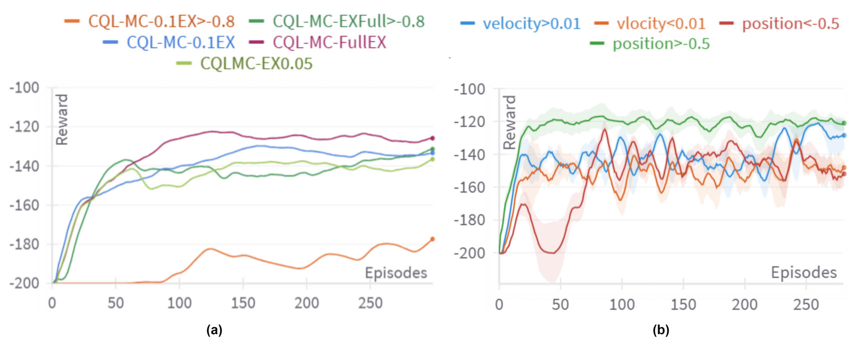


Figure 14: (a) The effect of data reduction and removal on baseline CQL visualized on Mountain Car Environment (b) Performance of ExID on removing different parts of the data based on nodes of Fig 10 (c) from Mountain Car expert dataset

Reducing transitions from the dataset: For all datasets, 10% of the data samples were extracted from the full dataset. This experimental design choice is based on the observation shown in Fig 14 (a). Performance degrades on reducing samples to 0.1% of the dataset and reduces further on reducing samples to 0.05% of the dataset. However, this drop is not substantial. The performance also reduces on removing part of the dataset from the full dataset with states > -0.8. However, the worst performance is observed when both samples are reduced and data is omitted, attributing to accumulated errors from probability ratio shift contributing to an increase in generalization error. Our methodology aims to address this gap in performance.

Removing part of the state space: Due to the simplicity of the Mountain-Car environment, we analyze the Mountain-Car expert dataset to show the effect of removing data matching state conditions of the different nodes in the decision tree in Fig 10 (c). The performance for each condition is summarised in Table 6. The most informative node in the tree is position > -0.5; removing states matching this condition causes a performance drop in the algorithm as the domain knowledge regularization does not contribute significant information to the policy. Similarly, removing data with velocity < 0.01 causes a performance drop. However, both performances are higher than the baseline CQL trained on reduced data. Based on this observation, we choose state removal conditions that preserve states matching part of the information in the tree such that the regularization term contributes substantially to the policy. Fig 15 shows the data distribution plot of 10% samples extracted from mountain car replay and noisy data with states > -0.8 removed. Fig 16 shows visualizations for 10% samples extracted from expert data with velocity > -1.5 removed. Fig 17 shows visualizations for 10% samples extracted from expert data with lander angle < -0.04 removed.

Table 6: Performance of ExID on removing different parts of the data based on nodes of Fig 10 (c) from Mountain Car expert dataset

Position>-0.5	Position<-0.5	Velocity>0.01	Velocity<0.01
-121.89 $\pm$ 7.69	-151 $\pm$ 13.6	-128.48 $\pm$ 11.84	-147.80 $\pm$ 5.01

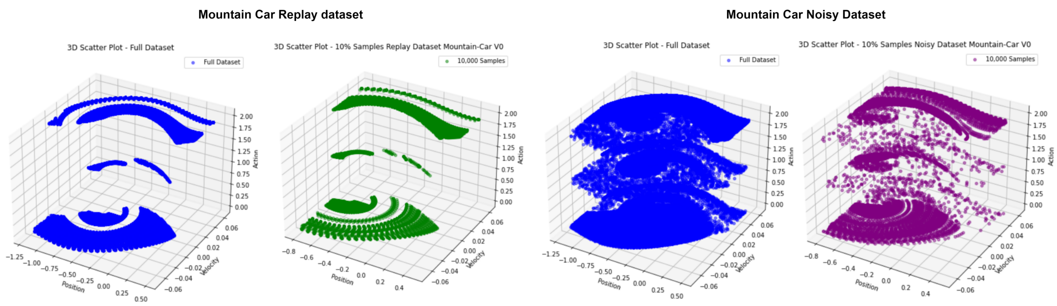


Figure 15: Data distribution of reduced dataset compared to the full dataset for mountain replay and noisy data

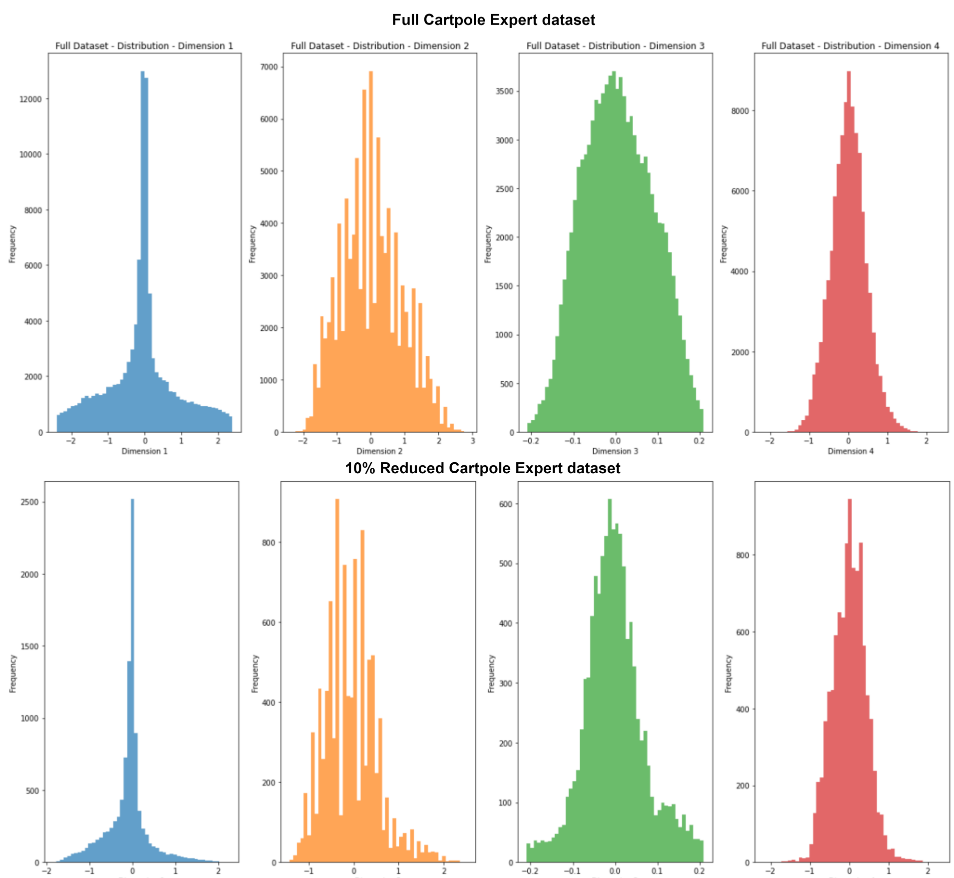


Figure 16: Data distribution of reduced cart pole expert dataset compared to the full dataset

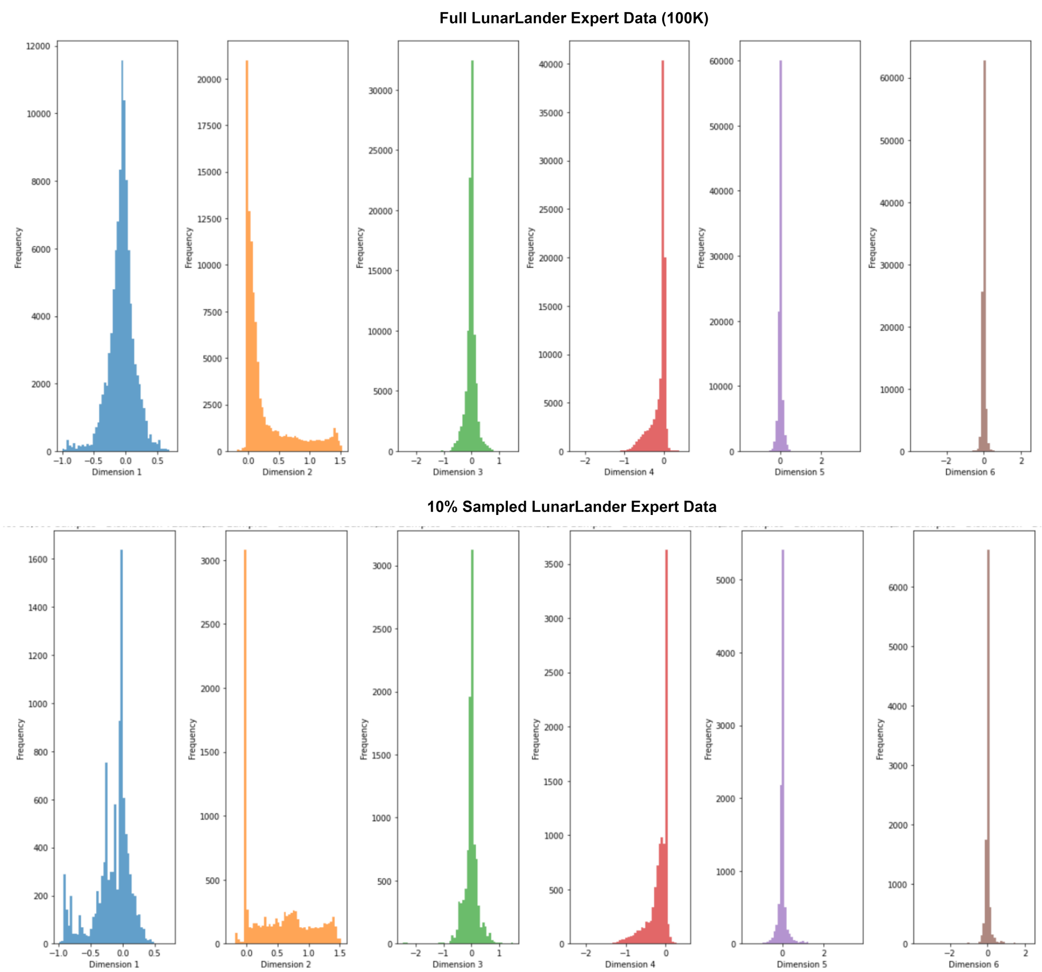


Figure 17: Data distribution of reduced LunarLander expert dataset compared to the full dataset

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 claims made in the paper have been experimented on different settings for validity and generalization. Please refer to sec 5.2.

   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: The paper acknowledges the dependency on reasonable domain knowledge and coverage please refer to sec 6

   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: Please refer to App A in the supplement material for theoretical analysis and proofs.

   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: Yes all hyper-parameters and experimental setting have been clearly listed in the paper. Please refer to App F and sec 5.1.

   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 code is provided with the submission in a zip file with Readme for instructions.

   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: The paper uses open source code to create the dataset and lists the modifications in details in the main paper and supplement material. Please refer to App F and sec 5.1.

   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: All experiments have been run on 3 random seeds and the error bounds have been reported in Table 5.1, Table 2 and Table 3.

   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: Please refer to the Experimental setup section in the main paper sec 5.1.

   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: The authors have reviewed the code of ethics and the paper adheres to it.

   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: [Yes]

    Justification: Please refer to the section broader impacts 7.

    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: The algorithm proposed in this paper does not not pose any such risk of misuse.

    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: All codes and datasets used in this paper are under MIT licence and the original owners have been cited.

    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: No new assets have been introduced in this paper.

    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: The paper did not require any crowdsourcing or human subject for experimentation.

    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: The paper did not require any crowdsourcing or human subject for experimentation.

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.