LEARNING WHAT TO DO AND WHAT NOT TO DO: OFFLINE IMITATION FROM EXPERT AND UNDESIRABLE DEMONSTRATIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Offline imitation learning typically learns from expert and unlabeled demonstrations, yet often overlooks the valuable signal in explicitly undesirable behaviors. In this work, we study offline imitation learning from contrasting behaviors, where the dataset contains both expert and undesirable demonstrations. We propose a novel formulation that optimizes a difference of KL divergences over the state-action visitation distributions of expert and undesirable (or bad) data. Although the resulting objective is a DC (Difference-of-Convex) program, we prove that it becomes *convex* when expert demonstrations outweigh undesirable demonstrations, enabling a practical and stable non-adversarial training objective. Our method avoids adversarial training and handles both positive and negative demonstrations in a unified framework. Extensive experiments on standard offline imitation learning benchmarks demonstrate that our approach consistently outperforms state-of-the-art baselines.

1 INTRODUCTION

Imitation learning (Garg et al., 2021; Kim et al., 2021; Li et al., 2023; Hoang et al., 2024a; Xu et al., 2022) offers a compelling alternative to Reinforcement Learning (RL) (Sutton & Barto, 2018; Puterman, 2014; Mnih et al., 2015) by enabling agents to learn directly from expert demonstrations without the need for explicit reward signals. This paradigm has been successfully applied in various domains, even with limited expert data, and is particularly effective in capturing complex human behaviors and preferences.

Imitation learning typically assumes access to high-quality expert demonstrations, which can be expensive and difficult to obtain (Ross et al., 2011; Torabi et al., 2018; Zhu et al., 2020). In practice, datasets often contain a mixture of expert and sub-optimal demonstrations. Recent advances in imitation learning have begun to address this more realistic setting, aiming to develop algorithms that can leverage informative signals from both expert and non-expert data (Brown et al., 2019; Myers et al., 2022; Hoang et al., 2024a).

In the offline setting, imitation learning methods typically assume the presence of a labeled expert dataset and an unlabeled dataset of mixed quality (which can contain expert, non-expert, and bad data), and further assume that the unlabeled demonstrations are not drastically different from expert behavior. This allows for framing the learning problem as mimicking both expert and unlabeled trajectories—albeit with different weights (Kim et al., 2021; 2022; Xu et al., 2022). However, in practice, unlabeled data may contain poor or undesirable demonstrations that the agent should explicitly avoid. For example, in autonomous driving, undesirable demonstrations may include unsafe lane changes or traffic violations, which should not be imitated under any circumstances. Another example can be found in healthcare applications, where undesirable demonstrations may correspond to incorrect diagnosis or unsafe treatment plans that could harm patients if imitated.

Unfortunately, existing imitation learning approaches are ill-equipped to deal with scenarios where both expert and undesirable demonstrations coexist within the dataset (Wu et al., 2019; Zhang et al., 2021; Hoang et al., 2024a). It is important to note that learning by mimicking expert or mildly sup-optimal demonstrations is often tractable, as the corresponding objective—typically framed as divergence minimization—is convex (Kim et al., 2021; 2022). However, incorporating objectives that explicitly avoid bad (or undesirable) demonstrations can introduce non-convexities, making the

optimization significantly more challenging. In this paper, we propose a unified framework that addresses these challenges, aiming to bridge this gap in the current imitation learning literature.

Specifically, we focus on the setting of *offline imitation learning* (no interaction with the environment) where the dataset contains both *expert* and *undesirable* demonstrations[1]. We make the following key contributions:

- We formulate the learning problem with the goal of matching expert behavior while explicitly avoiding undesirable demonstrations. Although the resulting training objective is expressed as the difference between two KL divergences (and is therefore difference-convex), we prove that it becomes *convex* when the expert component outweighs the undesirable one. This convexity is critical, as it enables us to reformulate the learning problem over the state-action visitation distribution as an more tractable unconstrained optimization via Lagrangian duality. Our objective stands in contrast to most existing distribution-matching imitation learning approaches, which typically rely solely on divergence minimization and naturally yield convex objectives. By introducing a divergence maximization term to account for undesirable behavior, we demonstrate that the overall objective *remains convex and manageable.*
- We further enhance the learning objective by proposing a surrogate objective that lower-bounds the original one, offering the advantage of a non-adversarial and convex optimization problem in the Q-function space. In addition, we introduce a novel Q-weighted behavior cloning (BC) approach, supported by theoretical guarantees, for efficient policy extraction.
- Extensive experiments on standard imitation learning benchmarks show that our method consistently outperforms existing approaches, both in conventional settings where datasets contain expert and unlabeled demonstrations, and in more realistic scenarios where explicitly undesirable demonstrations are included.

2 Related Works

Imitation Learning. Imitation learning trains agents to mimic expert behavior from demonstrations, with Behavioral Cloning (BC) serving as a foundational method by maximizing the likelihood of expert actions. However, BC often suffers from distributional shift (Ross et al., 2011). Recent work addresses this issue by leveraging the strong generalization capabilities of generative models (Zhao et al., 2023; Chi et al., 2023). Inspired by GANs (Goodfellow et al., 2014), methods like GAIL (Ho & Ermon, 2016) and AIRL (Fu et al., 2018) use a discriminator to align the learner's policy with the expert's, while SQIL (Reddy et al., 2019) simplifies reward assignment by distinguishing expert and non-expert behaviors. Although effective, these approaches typically require online interaction, which may be impractical in many real-world scenarios.

To address this, offline methods such as AlgaeDICE (Nachum et al., 2019) and ValueDICE (Kostrikov et al., 2020) employ Stationary Distribution Correction Estimation (DICE), though they often encounter stability issues. Building on ValueDICE, O-NAIL (Arenz & Neumann, 2020) avoids adversarial training, enabling stable offline imitation. More recently, several approaches have extended the DICE framework with stronger theoretical foundations and improved empirical performance (Lee et al., 2021; Mao et al., 2024). In parallel, IQ-Learn (Garg et al., 2021) has emerged as a unified framework for both online and offline imitation learning, inspiring a range of follow-up works (Al-Hafez et al., 2023; Hoang et al., 2024c). However, all these approaches rely on the presence of many expert demonstrations, which may not always be available.

Offline imitation learning from suboptimal demonstrations: Several approaches have been developed to tackle the challenges of offline imitation learning from suboptimal data, which is common in real-world scenarios. A notable direction involves preference-based methods, where algorithms infer reward functions by leveraging ranked or pairwise-compared trajectories to guide learning (Kim et al., 2023; Kang et al., 2023; Hejna & Sadigh, 2024). Recent works, such as SPRINQL (Hoang et al., 2024a), take advantage of demonstrations that exhibit varying levels of suboptimality, enabling the learner to better generalize beyond near-optimal behaviors. Another

[1] In practice, while desirable demonstrations can be collected from expert decisions, undesirable ones can also be identified by experts (or even through fine-tuned LLMs (Mu & Others, 2024))

important line of research explores the use of unlabeled demonstrations in conjunction with a limited number of expert trajectories. Techniques like DemoDICE (Kim et al., 2021), SMODICE (Ma et al., 2022), and ReCOIL (Sikchi et al., 2024) apply Distribution Correction Estimation (DICE) (Sunehag et al., 2017; Lee et al., 2021; Mao et al., 2024) to re-weight trajectories and align the state or state-action distributions with those of the expert. In parallel, classifier-based methods, such as DWBC (Xu et al., 2022), ISW-BC (Li et al., 2023), and ILID (Yue et al., 2024), use discriminators to distinguish expert-like behaviors within mixed-quality data and assign them greater importance. Collectively, these strategies aim to enhance policy robustness and performance in offline settings where high-quality expert data is scarce or expensive to obtain. However, all of these approaches are primarily focused on imitating and are unable to avoid undesirable or bad demonstrations, which is crucial in domains such as self driving where there are many unsafe behaviors that would need to be avoided. There is prior work that focuses on learning explicitly from undesirable demonstrations (Jang et al., 2024; Hoang et al., 2024b), but these approaches cannot handle scenarios where both expert and undesirable datasets are available.

In this paper, we aim to optimize on the principle of "*Imitate the Good and Avoid the Bad*", which has recently gained attention in reference and safe RL (Abdolmaleki et al., 2025; Hoang et al., 2024a; Gong et al., 2025) and large language model training (Lu et al., 2025). We extend this idea to the offline imitation setting by proposing a novel and efficient method that learns from expert demonstrations while avoiding undesirable ones. To our knowledge, this is the first offline imitation learning approach to efficiently learn policies by jointly utilizing both expert and undesirable demonstrations.

3 PRELIMINARIES

Markov Decision Process (MDP). We consider a MDP defined by the following tuple $\mathcal{M} = \langle S, A, r, P, \gamma, s_0 \rangle$, where S denotes the set of states, s_0 represents the initial state set, A is the set of actions, $r : S \times A \to \mathbb{R}$ defines the reward function for each state-action pair, and $P : S \times A \to S$ is the transition function, i.e., $P(s'|s, a)$ is the probability of reaching state $s' \in S$ when action $a \in A$ is made at state $s \in S$, and γ is the discount factor. In reinforcement learning (RL), the aim is to find a policy that maximizes the expected long-term accumulated reward: $\max_\pi \left\{\mathbb{E}_{(s,a)\sim d^\pi}[r(s, a)]\right\}$, where d^π is the occupancy measure (or state-action visitation distribution) of policy π: $d^\pi(s, a) = (1 - \gamma)\pi(a|s) \sum_{t=1}^{\infty} \gamma^t P(s_t = s|\pi)$.

Offline Imitation Leaning. Recent imitation learning (IL) approaches have adopted a distribution-matching formulation, where the objective is to minimize the divergence between the occupancy measures (i.e., state-action visitation distributions) of the learning policy and the expert policy: $\min_{d^\pi} \left\{D_f \left(d^\pi \,\|\, d^E\right)\right\}$, where D_f denotes an f-divergence between the occupancy distributions d^π (induced by the learning policy π) and d^E (induced by the expert policy). In particular, when the Kullback–Leibler (KL) divergence is used, the learning objective becomes: $\min_{d^\pi} \ \mathbb{E}_{(s,a)\sim d^\pi}\left[\log\left(\frac{d^\pi(s,a)}{d^E(s,a)}\right)\right]$. In the space of state-action visitation distributions (d^π), the training can be formulated as a convex constrained optimization problem. To enable efficient training, Lagrangian duality is typically employed to recast the problem into an unconstrained form (Lee et al., 2021; Kim et al., 2021).

Offline IL with unlabeled data. In offline imitation learning with unlabeled data, it is typically assumed that a limited set of expert demonstrations $\mathcal{B}^E$ is available, along with a larger set of unlabeled demonstrations $\mathcal{B}^{\text{MIX}}$. Distribution-matching approaches have been widely adopted to handle this setting. Prior methods often formulate the objective as a weighted sum of divergences between the learning policy and both expert and unlabeled data: $\min_{d^\pi} \left\{D_f \left(d^\pi \,\|\, d^E\right) + \alpha D_f \left(d^\pi \,\|\, d^{\text{MIX}}\right)\right\}$, where $\alpha \geq 0$. Other approaches construct mixtures of occupancy distributions, such as $d^{\pi,\text{MIX}} = \alpha d^\pi + (1 - \alpha) d^{\text{MIX}}$ and $d^{E,\text{MIX}} = \alpha d^E + (1 - \alpha) d^{\text{MIX}}$, and minimize the divergence between $d^{\pi,\text{MIX}}$ and $d^{E,\text{MIX}}$ (Kim et al., 2021; 2022; Ma et al., 2022; Sikchi et al., 2024). In most existing approaches along this line of research, the convexity of the objective with respect to d^π has been heavily leveraged to derive tractable learning objectives. However, when a divergence *maximization* term is introduced—as in our approach—this convexity may no longer hold, rendering many existing methods inapplicable.

4 DUALCOIL: OFFLINE IMITATION LEARNING FROM CONTRASTING BEHAVIORS

We begin by introducing a novel learning objective based on the difference between two KL divergences. Leveraging the convexity of this formulation, we derive a tractable and unconstrained optimization problem. Given that the resulting objective includes exponential terms that may lead to numerical instability, we enhance this by proposing a lower-bound approximation. This approximation enables us to reformulate the learning process as a more tractable, non-adversarial Q-learning objective, which remains convex in the space of Q-functions.

4.1 DUAL KL-BASED FORMULATION

Assume that we have access to three sets of demonstrations: good dataset $\mathcal{B}^G$ contains *good* or *expert* demonstrations, bad dataset $\mathcal{B}^B$ contains *bad* or *undesirable* demonstrations that the agent should avoid, and the unlabeled dataset $\mathcal{B}^{\text{MIX}}$ is a large set of unlabeled demonstrations used to support offline training. Here, we assume that $\mathcal{D}^B$ may contain some low-reward or unsafe demonstrations that are undesirable to imitate, though not necessarily ones that must be avoided at all costs. We consider the realistic scenario where the identified datasets $\mathcal{B}^G$ and $\mathcal{B}^B$ are limited in size, while $\mathcal{B}^{\text{MIX}}$ is significantly larger—an assumption that aligns with typical settings in offline imitation learning from unlabeled demonstrations.

Let $d^\pi(s,a)$, $d^G(s,a)$, and $d^B(s,a)$ denote the state-action visitation distributions induced by the learned policy π, the good policy, and the bad policy, respectively. Following the DICE framework (Nachum et al., 2019; Kostrikov et al., 2020), we propose to optimize the following objective:

$$\min_{d^\pi} \quad f(d^\pi) = D_{\text{KL}}(d^\pi \,\|\, d^G) - \alpha\, D_{\text{KL}}(d^\pi \,\|\, d^B), \tag{1}$$

where $\alpha > 0$ is a tunable hyperparameter. The goal of this objective is twofold: (1) to minimize the divergence between the learned policy and the good policy, and (2) to *maximize* the divergence from the bad policy, thereby avoiding undesirable behavior.

This formulation differs from all existing DICE-based approaches in the literature, which primarily focus on minimizing KL divergence—even when dealing with undesirable or unsafe demonstrations. By contrast, our approach introduces a principled mechanism to explicitly repel the learned policy from undesirable behavior while still aligning it with good data.

While the presence of a KL divergence maximization term in the objective may raise concerns about the convexity of the training problem, we observe that the objective in equation 1 takes the form of a difference between two convex functions. This is, in general, not convex and can be challenging to optimize. Fortunately, we show that under a mild condition, the overall objective remains convex. Specifically, if the weight on the bad policy divergence term is smaller than that on the good policy (i.e., $\alpha < 1$), then the objective becomes convex in d^π.

Proposition 4.1. *If $\alpha \leq 1$, then the objective function $f(d^\pi) = D_{KL}(d^\pi \,\|\, d^G) - \alpha\, D_{KL}(d^\pi \,\|\, d^B)$ is convex in d^π.*

Convexity is essential in most DICE-based frameworks, as it enables the use of Lagrangian duality to construct well-behaved and tractable training objectives. Our goal is to develop a Q-learning method that recovers a policy minimizing the objective in equation 1. To this end, we formulate the problem as the following constrained optimization:

$$\begin{aligned} \min_{d,\pi} \quad & f(d,\pi) = D_{\text{KL}}(d \,\|\, d^G) - \alpha\, D_{\text{KL}}(d \,\|\, d^B) \\ \text{s.t.} \quad & d(s,a) = (1-\gamma)p_0(s)\pi(a \mid s) + \gamma\pi(a \mid s) \sum_{s',a'} d(s',a')T(s \mid s',a'),\ \forall s \in S, a \in A \end{aligned} \tag{2}$$

where $d(s,a)$ is the state-action visitation distribution, and T is the environment transition function. Let $\mathcal{B}^U = \mathcal{B}^G \cup \mathcal{B}^{\text{MIX}}$ denote the union dataset, and let d^U be the state-action visitation distribution derived from it. The following proposition gives an another formulation for the objective in equation 1:

Proposition 4.2. *The objective function in equation 2 can be written as: $f(d,\pi) = (1-\alpha)D_{KL}(d||d^U) - \mathbb{E}_{(s,a)\sim d}\left[\Psi(s,a)\right]$, where $\Psi(s,a) = \log \frac{d^G(s,a)}{d^U(s,a)} - \alpha \log \frac{d^B(s,a)}{d^U(s,a)}$.*

This formulation introduces a KL-based regularization centered on the reference distribution d^U, with $\Psi(s,a)$ acting as a correction term that incorporates information from the labeled good and bad demonstrations. The reformulated objective in Proposition 4.2 further confirms that the function $f(d,\pi)$ remains convex in d when $\alpha \leq 1$. Here we note that, under the same condition $\alpha \leq 1$, convexity may not hold for other f-divergences (a detailed discussion is provided in the appendix).

The reformulated objective in Proposition 4.2 takes the conventional form of maximizing a long-term surrogate reward $\Psi(s,a)$, subtracted by a KL divergence between two occupancy measures. Hence, we can follow the approach in DUARL Sikchi et al. (2024) to further reformulate it into a practical Q-learning objective (the detailed derivation is given in Appendix).

$$\max_{\pi}\min_{Q}\Big\{(1-\gamma)\,\mathbb{E}_{(s,a)\sim p_0,\pi}\left[Q(s,a)\right] + (1-\alpha)\mathbb{E}_{(s,a)\sim d^U}\left[\exp\left(\frac{\Psi(s,a)+\gamma\,\mathbb{E}_{(s',a')\sim T,\pi}[Q(s',a')]-Q(s,a)}{1-\alpha}\right)\right]\Big\} \quad (3)$$

To further enhance the efficiency of Q-learning, we adopt the well-known Maximum Entropy (MaxEnt) reinforcement learning framework by incorporating an entropy term into the training objective (Garg et al., 2021; Haarnoja et al., 2018). This leads to the following objective:

$$L(Q,\pi) = (1-\gamma)\,\mathbb{E}_{(s,a)\sim p_0,\pi}\left[Q(s,a) - \beta\log\frac{\pi(a\mid s)}{\mu^U(a|s)}\right] + (1-\alpha)\mathbb{E}_{(s,a)\sim d^U}\left[\exp\left(\frac{\Psi(s,a)+\gamma\,\mathbb{E}_{(s',a')\sim T,\pi}\left[Q(s',a')-\beta\log\frac{\pi(a'|s')}{\mu^U(a'|s')}\right]-Q(s,a)}{1-\alpha}\right)\right].$$

where $\mu^U(a|s)$ is the behavior policy representing the union dataset $\mathcal{B}^U$. We now define the soft value function and the soft Bellman operator as follows:

$$V_Q^\pi(s) = \mathbb{E}_{a\sim\pi(\cdot|s)}\left[Q(s,a) - \beta\log\frac{\pi(a\mid s)}{\mu^U(a|s)}\right], \quad \mathcal{T}^\pi[Q](s,a) = Q(s,a) - \gamma\,\mathbb{E}_{s'\sim\mathcal{T}(\cdot|s,a)}\left[V_Q^\pi(s')\right].$$

Using these definitions, the training objective can be rewritten as:

$$L(Q,\pi) = (1-\gamma)\,\mathbb{E}_{s\sim p_0}\left[V_Q^\pi(s)\right] + (1-\alpha)\mathbb{E}_{(s,a)\sim d^U}\left[\exp\left(\frac{\Psi(s,a)-\mathcal{T}^\pi[Q](s,a)}{1-\alpha}\right)\right]. \quad (4)$$

This formulation shares structural similarities with IQ-Learn, where $\mathcal{T}^\pi[Q](s,a)$ is referred to as the *inverse Bellman operator* and is often interpreted as a reward function expressed in terms of the Q-function itself.

Note that, when $\alpha = 1$, according to Proposition 4.2, the training objective reduces to a standard offline RL problem with reward function $\Psi(s,a)$: $\max_d \mathbb{E}_{(s,a)\sim d}\left[\Psi(s,a)\right] = \max\mathbb{E}\left[\sum_{t=0}^{\infty}\gamma^t\Psi(s_t,a_t)\right]$.

4.2 Tractable Lower Bounded Objective

In this section, we propose an additional step to improve the stability and tractability of the learning objective introduced above. We first observe that the exponential term in equation 4 may lead to instability during training. To address this issue, we propose to approximate the exponential using a linear lower bound, which not only improves stability but also preserves a similar optimization objective.

Proposition 4.3. *Let the surrogate objective be defined as:*

$$\widetilde{L}(Q,\pi) = (1-\gamma)\,\mathbb{E}_{s\sim p_0}\left[V_Q^\pi(s)\right] - \mathbb{E}_{d^U}\left[\delta(s,a)\mathcal{T}^\pi[Q](s,a)\right] + (1-\alpha)\mathbb{E}_{d^U}\left[\delta(s,a)\right]. \quad (5)$$

where $\delta(s,a) = \exp\left(\frac{\Psi(s,a)}{1-\alpha}\right)$. *Then* $\widetilde{L}(Q,\pi)$ *is a lower bound of* $L(Q,\pi)$, *with equality when* $\mathcal{T}^\pi[Q](s,a) = 0$ *for all* (s,a).

The lower-bound approximation $\widetilde{L}(Q,\pi)$ offers several benefits. First, as a valid lower bound of $L(Q,\pi)$, maximizing $\widetilde{L}(Q,\pi)$ promotes the original objective. Second, its structure—linear in Q

and concave in π—leads to a simplified, non-adversarial training procedure (see Proposition 4.4). Finally, its optimization goals remain aligned with those of $L(Q, \pi)$, encouraging high expected soft value under the initial state distribution and consistency between the soft Bellman residual and the guidance signal $\Psi(s, a)$.

Remark. *The training objective in equation 5 generalizes the IQ-Learn objective (Garg et al., 2021) as a special case. In particular, $\widetilde{L}(Q, \pi)$ reduces exactly to the IQ-Learn objective when $\alpha = 0$ (i.e., the undesirable dataset is ignored) and $\mathcal{B}^G \equiv \mathcal{B}^U$ (i.e., the good dataset coincides with the union dataset). To see this, observe that when $\alpha = 0$ and $d^G = d^U$, the term $\Psi(s, a)$ becomes zero for all (s, a). As a result, the surrogate objective simplifies to: $\widetilde{L}(Q, \pi) = (1-\gamma)\,\mathbb{E}_{s\sim p_0}\left[V_Q^\pi(s)\right] - \mathbb{E}_{(s,a)\sim d^G}\left[\mathcal{T}^\pi[Q](s, a)\right]$, which is exactly the training objective proposed in IQ-Learn. Thus, our formulation can be viewed as a principled extension of IQ-Learn that explicitly accounts for and contrasts between good and bad behaviors.*

We now present several key properties of the training objective $\widetilde{L}(Q, \pi)$ that make it particularly convenient and tractable for use, as formalized in Proposition 4.4 below.

Proposition 4.4. *The following properties hold:*

(i) $\widetilde{L}(Q, \pi)$ is linear in Q and concave in π. As a result, the max–min optimization can be equivalently reformulated as a min–max problem: $\max_\pi \min_Q \widetilde{L}(Q, \pi) = \min_Q \max_\pi \widetilde{L}(Q, \pi)$.

(ii) The min–max problem $\min_Q \max_\pi \widetilde{L}(Q, \pi)$ reduces to the following non-adversarial problem:

$$\min_Q \left\{ \widetilde{L}(Q) = (1-\gamma)\,\mathbb{E}_{s\sim p_0}\left[V_Q(s)\right] - \mathbb{E}_{(s,a)\sim d^U}\left[\exp\left(\frac{\Psi(s,a)}{1-\alpha}\right)\mathcal{T}[Q](s,a)\right]\right\},$$

where the soft value function $V_Q(s)$ is defined as: $V_Q(s) = \beta \log\left(\sum_a \mu^U(a|s)\exp(Q(s,a)/\beta)\right)$, and the soft Bellman residual operator is given by: $\mathcal{T}[Q](s, a) = Q(s, a) - \gamma V_Q(s)$. Moreover $\widetilde{L}(Q)$ is convex in Q.

5 Practical Algorithm

Estimating Occupancy Ratios. The training objective involves several ratios between state-action visitation distributions, which are not directly observable. These quantities can be estimated by solving corresponding discriminator problems. Specifically, to estimate the ratio $\frac{d^G(s,a)}{d^U(s,a)}$, we train a binary classifier $c^G : \mathcal{S} \times \mathcal{A} \to [0, 1]$ by solving the following standard logistic regression objective:

$$\max_{c^G}\{\mathbb{E}_{(s,a)\sim d^G}[\log c^G(s, a)] + \mathbb{E}_{(s,a)\sim d^U}[\log(1 - c^G(s, a))]\}. \tag{6}$$

Let $c^{G*}(s, a)$ be optimal solution to this problem, then the ratio can be computed as: $\frac{d^G(s,a)}{d^U(s,a)} = \frac{c^{G*}(s,a)}{1-c^{G*}(s,a)}$. Similar discriminators can be trained to estimate other ratios such as $\frac{d^B(s,a)}{d^U(s,a)}$.

Implicit V-Update and Regularizers. In the surrogate objective $\widetilde{L}(Q)$, the value function V_Q is typically computed via a log-sum-exp over Q, which becomes intractable in large or continuous action spaces. To address this, we adopt Extreme Q-Learning (XQL) (Garg et al., 2023), which avoids the log-sum-exp by introducing an auxiliary optimization over V, jointly updated with Q. Specifically, V is optimized using the *Extreme-V* objective: $J(V \mid Q) = \mathbb{E}_{(s,a)\sim d^U}\left[e^{t(s,a)} - t(s, a) - 1\right]$, where $t(s, a) = \frac{Q(s,a)-V(s)}{\beta}$. The main training objective with fixed V is:

$$\widetilde{L}(Q \mid V) = (1-\gamma)\,\mathbb{E}_{s\sim p_0}\left[V(s)\right] - \mathbb{E}_{(s,a)\sim d^U}\left[\exp\left(\frac{\Psi(s,a)}{1-\alpha}\right)(Q(s, a) - \gamma\mathbb{E}_{s'}[V(s')])\right]. \tag{7}$$

The overall optimization proceeds by alternating: (i) updating Q via minimizing $\widetilde{L}(Q \mid V)$, and (ii) updating V via minimizing $J(V \mid Q)$. Both sub-problems are convex, enabling efficient and

stable training. To further enhance stability, we follow (Garg et al., 2021; 2023) and add a convex regularizer $\phi(\mathcal{T}[Q](s,a))$ to prevent reward divergence. We use the χ^2-divergence, $\phi(t) = t^2/2$, a common choice in Q-learning.

Policy Extraction. Once the Q and V functions are obtained, a common approach for expert policy extraction is to apply advantage-weighted behavior cloning (AW-BC) (Kostrikov et al., 2021; Garg et al., 2023; Hejna & Sadigh, 2024; Sikchi et al., 2024):

$$\max_{\pi} \sum_{(s,a)\sim\mathcal{B}^U} \exp\left(\frac{1}{\beta}\left(Q(s,a) - V(s)\right)\right) \log \pi(a \mid s). \tag{8}$$

A key limitation of this formulation is that the value function $V(s)$ is only an approximate estimate from the Extreme-V objective, potentially introducing noise and bias into advantage computation and degrading policy quality. To address this, we propose a Q-only alternative that avoids reliance on $V(s)$. The following proposition shows that this Q-based objective can, in theory, recover the same optimal policy as the original advantage-weighted BC formulation.

Proposition 5.1. *The following Q-weighted behavior cloning (BC) objective yields the same optimal policy as the original advantage-weighted BC formulation in equation 8:*

$$\max_{\pi} \sum_{(s,a)\sim\mathcal{B}^U} \exp\left(\frac{1}{\beta}Q(s,a)\right) \log \pi(a \mid s). \tag{9}$$

While the Q-weighted BC objective is theoretically equivalent to the advantage-weighted BC objective in terms of the optimal policy it recovers, it provides a simpler and more practical formulation. This simplification can lead to more stable and accurate optimization in practice. Our experimental results further demonstrate that the Q-weighted formulation consistently yields significantly better training outcomes compared to the advantage-weighted BC baseline. Bringing all components together, we present our DUALCOIL (**Dual**-KL **CO**ntrastive **I**mitation **L**earning) algorithm in Algorithm 1.

Algorithm 1 DualCOIL

Require: Datasets $\mathcal{B}^G$, $\mathcal{B}^B$, $\mathcal{B}^{\text{MIX}}$; training steps N_μ, N; models: $c^G_{w_G}$, $c^B_{w_B}$, π_θ, Q_{w_q}, V_{w_v}

Assign $\mathcal{B}^U = \mathcal{B}^G \cup \mathcal{B}^{\text{MIX}}$
 # Train discriminator $c^G_{w_G}$ and $c^B_{w_B}$
for $i = 1$ to N_μ **do**
  Update (w_G, w_B) to minimize Objective 6.
end for
 # Train Q_{w_q} and V_{w_v}, and policy π_θ
for $i = 1$ to N **do**
  Update w_q to minimize $\widetilde{F}(Q_{w_q}|V_{w_v})$
  Update w_v to minimize $J(V_{w_v}|Q_{w_q})$
  Update θ via QW-BC:
  $\max_\pi \left\{\sum_{(s,a)\sim\mathcal{B}^U} e^{Q(s,a)/\beta} \log \pi(a|s)\right\}$
end for

6 EXPERIMENTS

In this section, we conduct extensive experiments to evaluate our method, focusing on the following key questions: **(Q1)** Can DualCOIL effectively leverage both labeled good and bad data to outperform existing baselines? **(Q2)** How does the size of the bad dataset $\mathcal{B}^B$ affect the performance of DualCOIL? **(Q3)** DualCOIL relies on an important parameter α to balance the objectives for good and bad data—how does this parameter affect overall performance? Appendix provides a comprehensive set of supplementary materials, including missing proofs and additional experiments that further validate the utility of our method.

6.1 EXPERIMENT SETTING

Environments and Dataset Generation. We evaluate our method in the context of learning from the good dataset $\mathcal{B}^G$ and avoid the bad dataset $\mathcal{B}^B$ with a support from an additional unlabeled dataset $\mathcal{B}^{\text{MIX}}$. Our experiments span four MuJoCo locomotion tasks: CHEETAH, ANT, HOPPER, WALKER, as well as four hand manipulation tasks from Adroit: PEN, HAMMER, DOOR, RELOCATE, and one task from FrankaKitchen: KITCHEN—all sourced from the official D4RL benchmark (Fu et al., 2020). For each MuJoCo task from D4RL, we have three types of datasets: RANDOM, MEDIUM, and EXPERT. The good dataset $\mathcal{B}^G$ is constructed using a single trajectory from the EXPERT dataset. The bad dataset

$\mathcal{B}^B$ consists of 10 trajectories selected from either the RANDOM or MEDIUM dataset. To construct the unlabeled dataset $\mathcal{B}^{\text{MIX}}$, we combine the entire RANDOM or MEDIUM dataset (i.e., the same source as $\mathcal{B}^B$) with 30 additional trajectories from the EXPERT dataset. This setup mirrors the challenging RANDOM+FEW-EXPERT and MEDIUM+FEW-EXPERT scenarios introduced in ReCOIL (Sikchi et al., 2024). These three datasets—$\mathcal{B}^G$, $\mathcal{B}^B$, and $\mathcal{B}^{\text{MIX}}$—form the foundation of our training pipeline. We use the same dataset construction strategy for Adroit and FrankaKitchen tasks, yielding 18 distinct dataset combinations. Please refer to the Appendix for detailed descriptions of all dataset combinations.

Baselines. We compare our method against several baselines. First, we evaluate two naive BC approaches: one that learns directly from the large unlabeled dataset $\mathcal{B}^{\text{MIX}}$ (BC-MIX), and one that learns solely from the good dataset $\mathcal{B}^G$ (BC-G). Next, we include comparisons with state-of-the-art methods designed to leverage both expert (or good) data $\mathcal{B}^G$ and unlabeled data $\mathcal{B}^{\text{MIX}}$, including SMODICE (Ma et al., 2022), ILID (Yue et al., 2024), and ReCOIL (Sikchi et al., 2024). We exclude DWBC (Xu et al., 2022) from this experiment since both DWBC and ILID use discriminator-based objectives, and ILID has been shown to outperform DWBC. In addition, based on our proposed objective in equation 5, we include a variant of our method that only learns from $\mathcal{B}^G$ and $\mathcal{B}^{\text{MIX}}$ (i.e., $\alpha = 0$), called as DualCOIL-G. For methods that incorporate support from bad data $\mathcal{B}^B$, we evaluate our approach against SafeDICE (Jang et al., 2024). Given the limited number of existing baselines that effectively utilize poor-quality data in offline imitation learning, we also propose a simple adaptation of DWBC, which is called as DWBC-GB to jointly learn from $\mathcal{B}^G$, $\mathcal{B}^B$, and $\mathcal{B}^{\text{MIX}}$. Detailed implementation of these baselines are provided in the Appendix.

Evaluation Metrics. We evaluate all methods using five training seeds. For each seed, we collect the results from the last 10 evaluations (each evaluation consist 10 different environment seeds), then aggregate all evaluations across seeds to compute the mean and standard deviation, which reflect the converged performance of each method. Across all experiments, we report the normalized score commonly used in D4RL tasks $\left(\text{Normalized Score} = \frac{\text{Score} - \text{Random Score}}{\text{Expert Score} - \text{Random Score}}\right)$. This normalization provides a consistent performance measure across different environments.

6.2 MAIN COMPARISON

Task	unlabeled $\mathcal{B}^{\text{MIX}}$			learning from $\mathcal{B}^G$ and $\mathcal{B}^{\text{MIX}}$ only				learning with $\mathcal{B}^B$			
		BC-MIX	BC-G	SMODICE	ILID	ReCOIL	DualCOIL-G	SafeDICE	DWBC-GB	DualCOIL	Expert
CHEETAH	RANDOM+EXPERT	$2.3_{\pm 0.0}$	$-0.6_{\pm 0.3}$	$4.6_{\pm 1.2}$	$21.1_{\pm 3.4}$	$2.0_{\pm 0.3}$	$84.4_{\pm 2.4}$	$-0.0_{\pm 0.0}$	$2.8_{\pm 0.5}$	**86.7**$_{\pm 2.2}$	90.6
	MEDIUM+EXPERT	$42.5_{\pm 0.2}$	$-0.6_{\pm 0.3}$	$42.4_{\pm 1.6}$	$40.3_{\pm 7.0}$	$42.5_{\pm 0.3}$	$48.6_{\pm 2.0}$	$37.7_{\pm 0.1}$	$5.6_{\pm 1.9}$	**77.6**$_{\pm 3.6}$	90.6
ANT	RANDOM+EXPERT	$30.9_{\pm 0.0}$	$-7.2_{\pm 4.6}$	$4.6_{\pm 9.7}$	$71.8_{\pm 8.7}$	$56.2_{\pm 5.0}$	$100.6_{\pm 9.9}$	$-2.6_{\pm 0.0}$	$6.5_{\pm 3.4}$	**112.7**$_{\pm 5.8}$	117.5
	MEDIUM+EXPERT	$91.2_{\pm 0.8}$	$-7.2_{\pm 4.6}$	$88.5_{\pm 4.2}$	$39.6_{\pm 11.5}$	$100.8_{\pm 4.0}$	$102.4_{\pm 3.5}$	$88.1_{\pm 0.4}$	$-4.3_{\pm 2.4}$	**107.4**$_{\pm 4.9}$	117.5
HOPPER	RANDOM+EXPERT	$4.9_{\pm 0.1}$	$17.9_{\pm 2.7}$	$56.4_{\pm 9.2}$	$81.6_{\pm 14.3}$	$81.0_{\pm 14.7}$	$79.4_{\pm 14.8}$	$41.1_{\pm 1.4}$	$40.8_{\pm 9.5}$	**93.6**$_{\pm 9.2}$	109.6
	MEDIUM+EXPERT	$52.2_{\pm 0.6}$	$17.9_{\pm 2.7}$	$53.0_{\pm 1.7}$	$87.9_{\pm 5.3}$	$46.1_{\pm 8.3}$	$70.6_{\pm 8.0}$	$55.8_{\pm 1.7}$	$21.6_{\pm 4.0}$	**103.7**$_{\pm 7.3}$	109.6
WALKER	RANDOM+EXPERT	$1.5_{\pm 0.0}$	$3.8_{\pm 1.5}$	$106.6_{\pm 0.7}$	$100.1_{\pm 4.4}$	$29.8_{\pm 14.9}$	$97.5_{\pm 10.7}$	$23.0_{\pm 0.8}$	$17.4_{\pm 7.5}$	**107.4**$_{\pm 1.7}$	107.7
	MEDIUM+EXPERT	$70.8_{\pm 0.3}$	$3.8_{\pm 1.5}$	$6.0_{\pm 2.2}$	$89.7_{\pm 10.6}$	$72.1_{\pm 5.4}$	$99.8_{\pm 6.9}$	$60.2_{\pm 1.3}$	$25.6_{\pm 7.4}$	**108.2**$_{\pm 0.4}$	107.7
PEN	CLONED+EXPERT	$56.0_{\pm 0.5}$	$8.8_{\pm 1.4}$	$10.9_{\pm 6.5}$	$1.9_{\pm 2.1}$	$79.2_{\pm 9.6}$	$66.3_{\pm 9.6}$	$19.9_{\pm 2.1}$	$9.5_{\pm 3.9}$	**96.4**$_{\pm 8.7}$	107.0
	HUMAN+EXPERT	$18.3_{\pm 0.6}$	$8.8_{\pm 1.4}$	$-2.5_{\pm 0.2}$	$5.1_{\pm 2.1}$	$99.9_{\pm 8.5}$	$95.5_{\pm 8.8}$	$21.8_{\pm 2.5}$	$6.5_{\pm 2.4}$	**101.5**$_{\pm 8.4}$	107.0
HAMMER	CLONED+EXPERT	$0.4_{\pm 0.4}$	$1.4_{\pm 0.3}$	$0.8_{\pm 0.4}$	$0.4_{\pm 0.6}$	$3.4_{\pm 2.1}$	$66.5_{\pm 11.8}$	$0.0_{\pm 0.1}$	$2.8_{\pm 2.5}$	**74.3**$_{\pm 8.0}$	119.0
	HUMAN+EXPERT	$12.8_{\pm 3.3}$	$1.4_{\pm 0.3}$	$1.9_{\pm 2.1}$	$1.2_{\pm 1.4}$	$113.2_{\pm 5.5}$	$113.2_{\pm 7.2}$	$0.6_{\pm 0.4}$	$3.4_{\pm 1.9}$	**120.0**$_{\pm 3.7}$	119.0
DOOR	CLONED+EXPERT	$0.4_{\pm 0.3}$	$-0.1_{\pm 0.0}$	$-0.1_{\pm 0.0}$	$-0.1_{\pm 0.1}$	$19.3_{\pm 7.5}$	$92.6_{\pm 5.1}$	$-0.0_{\pm 0.0}$	$-0.1_{\pm 0.0}$	**102.4**$_{\pm 1.7}$	105.3
	HUMAN+EXPERT	$4.0_{\pm 1.2}$	$-0.1_{\pm 0.0}$	$-0.1_{\pm 0.3}$	$0.2_{\pm 0.7}$	$100.3_{\pm 2.9}$	$104.7_{\pm 0.7}$	$0.9_{\pm 0.4}$	$1.1_{\pm 0.5}$	**105.0**$_{\pm 0.5}$	105.3
RELOCATE	CLONED+EXPERT	$-0.1_{\pm 0.0}$	$-0.1_{\pm 0.0}$	$0.1_{\pm 0.1}$	$-0.1_{\pm 0.0}$	$1.4_{\pm 1.1}$	$34.5_{\pm 6.2}$	$-0.1_{\pm 0.0}$	$-0.2_{\pm 0.0}$	**92.1**$_{\pm 5.0}$	100.9
	HUMAN+EXPERT	$0.0_{\pm 0.0}$	$-0.1_{\pm 0.0}$	$-0.2_{\pm 0.0}$	$-0.2_{\pm 0.1}$	$72.3_{\pm 5.6}$	$99.1_{\pm 3.1}$	$0.0_{\pm 0.0}$	$-0.1_{\pm 0.0}$	**102.6**$_{\pm 2.4}$	100.9
KITCHEN	PARTIAL+COMPLETE	$45.5_{\pm 0.8}$	$2.5_{\pm 2.2}$	$5.5_{\pm 3.7}$	$27.3_{\pm 2.4}$	$48.8_{\pm 4.0}$	$45.8_{\pm 6.6}$	$2.8_{\pm 0.5}$	$19.4_{\pm 2.1}$	**53.1**$_{\pm 5.9}$	75.0
	MIXED+COMPLETE	$42.1_{\pm 0.5}$	$2.2_{\pm 1.7}$	$3.1_{\pm 2.6}$	$13.3_{\pm 1.4}$	**50.6**$_{\pm 1.7}$	$20.3_{\pm 6.3}$	$1.5_{\pm 0.8}$	$6.7_{\pm 2.0}$	$48.9_{\pm 7.3}$	75.0
Average		26.4	2.9	21.2	32.4	56.6	78.8	19.5	9.2	**94.1**	

Table 1: Comparison with other baselines in MuJoCo, Adroit, and FrankaKitchen. The results are normalized score in mean and standard error.

To answer Question **(Q1)**, we present a comprehensive comparison between our method and existing baselines across 18 different datasets, as shown in Table 1. First, both BC-MIX and BC-G fail to achieve satisfactory performance across tasks. When learning from the good dataset $\mathcal{B}^G$ and the unlabeled dataset $\mathcal{B}^{\text{MIX}}$, methods like SMODICE and ILID perform reasonably well on the four MuJoCo locomotion tasks (CHEETAH, ANT, HOPPER, WALKER) but completely fail on the five hand manipulation tasks. In contrast, ReCOIL and our method variant (DualCOIL-G) are

able to successfully learn in both locomotion and manipulation tasks, demonstrating more robust generalization.

In the setting that incorporates additional low-quality data $\mathcal{B}^B$, SafeDICE shows similar performance to SMODICE and ILID—again failing on the manipulation tasks. Furthermore, DWBC-GB fails to learn entirely, highlighting that a naive adaptation for leveraging poor-quality data can harm the learning process. These results suggest that incorporating bad data $\mathcal{B}^B$ introduces new challenges, and that effectively utilizing such data requires a carefully designed algorithm grounded in strong theoretical principles. Overall, our method successfully leverages the bad dataset $\mathcal{B}^B$ and consistently outperforms all other baselines across both locomotion and manipulation tasks.

6.3 Effect of Number of Bad Demonstrations

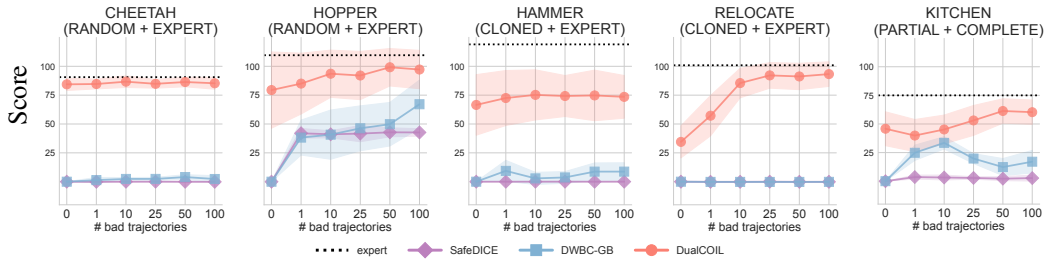

Figure 1: Effect of bad dataset size $\mathcal{B}^B$ on performance: Results, averaged over 5 seeds and reported as normalized scores, show that our method effectively leverages increasing numbers of bad trajectories, whereas baselines such as SafeDICE and DWBC-GB fail to do so.

To answer question **(Q2)**, we investigate the impact of the size of the undesirable (bad) dataset on methods designed to learn from bad data. Specifically, we gradually increase the size of the bad dataset $\mathcal{B}^B$ and evaluate how the performance of each algorithm is affected. The experimental results are presented in Figure 1. Overall, SafeDICE fails to effectively utilize the bad demonstrations, while DWBC-GB is only able to learn in the HOPPER task. In contrast, our method demonstrates strong scalability with respect to the size of the bad dataset, maintaining good performance even when provided with as few as a single bad trajectory.

6.4 Sensitivity Analysis of α

We introduce a hyperparameter $0 \leq \alpha < 1$, which controls the weighting of the bad data objective—this relates to question **(Q3)**. To evaluate the sensitivity of our method to α, we vary its value and observing the effect on final performance, as shown in Figure 2. While α does have a noticeable impact, our method remains robust across a broad range of values, with optimal performance observed within this range. The specific α values used for each task are provided in the Appendix.

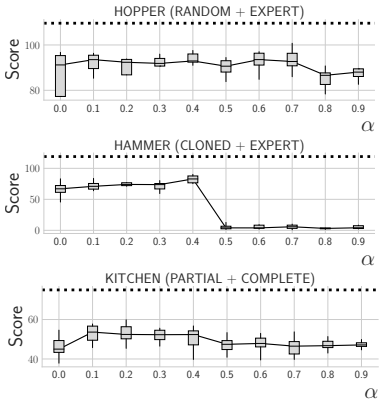

Figure 2: Sensitivity analysis on the trade-off parameter α.

7 Conclusion

We introduced a new offline imitation learning framework that leverages both expert and explicitly undesirable demonstrations. By formulating the learning objective as the difference of KL divergences over visitation distributions, we capture informative contrasts between good and bad behaviors. While the resulting DC program is generally non-convex, we establish conditions under which it becomes convex—specifically, when expert data dominates—leading to a practical, stable, and non-adversarial training procedure. Our unified approach to handling both expert and undesirable demonstrations

yields superior performance across a range of offline imitation learning benchmarks, setting a new standard for learning from contrasting behaviors.

Limitations and Future Work. While our method demonstrates strong empirical performance, it is currently limited to settings where $\alpha \leq 1$. Relaxing this constraint would make the learning objective more difficult to optimize but represents a promising direction for future research. In addition, we assume access to well-labeled expert and undesirable demonstrations, which may not always be available in practical scenarios. Developing robust algorithms capable of learning effectively from noisy or weakly labeled data would thus be a valuable extension of this work. Moreover, we assume that the undesirable dataset contains undesirable demonstrations that are not necessarily catastrophic or must be avoided at all costs. Addressing more safety-critical settings would require incorporating ideas from hard-constrained RL, which we consider a promising avenue for future exploration.

Reproducibility Statement

We provide detailed hyperparameters and network architectures for each task in the Appendix. The source code has been submitted as supplementary material and will be made publicly available to ensure reproducibility and comparison. In addition, all datasets used are either publicly available or clearly described in the Appendix.

References

Abbas Abdolmaleki, Bilal Piot, Bobak Shahriari, Jost Tobias Springenberg, Tim Hertweck, Michael Bloesch, Rishabh Joshi, Thomas Lampe, Junhyuk Oh, Nicolas Heess, Jonas Buchli, and Martin Riedmiller. Learning from negative feedback, or positive feedback or both. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=4FVGowGzQb.

Firas Al-Hafez, Davide Tateo, Oleg Arenz, Guoping Zhao, and Jan Peters. Ls-iq: Implicit reward regularization for inverse reinforcement learning. In *Eleventh International Conference on Learning Representations (ICLR)*, 2023. URL https://openreview.net/pdf?id=o3Q4m8jg4BR.

Oleg Arenz and Gerhard Neumann. Non-adversarial imitation learning and its connections to adversarial methods. *arXiv preprint arXiv:2008.03525*, 2020.

Daniel Brown, Wonjoon Goo, Prabhat Nagarajan, and Scott Niekum. Extrapolating beyond suboptimal demonstrations via inverse reinforcement learning from observations. In *International conference on machine learning*, pp. 783–792. PMLR, 2019.

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

Justin Fu, Katie Luo, and Sergey Levine. Learning robust rewards with adverserial inverse reinforcement learning. In *International Conference on Learning Representations*, 2018.

Justin Fu, Aviral Kumar, Ofir Nachum, George Tucker, and Sergey Levine. D4rl: Datasets for deep data-driven reinforcement learning, 2020.

Divyansh Garg, Shuvam Chakraborty, Chris Cundy, Jiaming Song, and Stefano Ermon. Iq-learn: Inverse soft-q learning for imitation. *Advances in Neural Information Processing Systems*, 34: 4028–4039, 2021.

Divyansh Garg, Joey Hejna, Matthieu Geist, and Stefano Ermon. Extreme q-learning: Maxent rl without entropy. In *International Conference on Learning Representations (ICLR)*, 2023. URL https://arxiv.org/abs/2301.02328.

Ze Gong, Akshat Kumar, and Pradeep Varakantham. Offline safe reinforcement learning using trajectory classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 16880–16887, 2025.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

Tuomas Haarnoja, Aurick Zhou, Kristian Hartikainen, George Tucker, Sehoon Ha, Jie Tan, Vikash Kumar, Henry Zhu, Abhishek Gupta, Pieter Abbeel, et al. Soft actor-critic algorithms and applications. *arXiv preprint arXiv:1812.05905*, 2018.

Joey Hejna and Dorsa Sadigh. Inverse preference learning: Preference-based rl without a reward function. *Advances in Neural Information Processing Systems*, 36, 2024.

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

Huy Hoang, Tien Mai, and Pradeep Varakantham. Imitate the good and avoid the bad: An incremental approach to safe reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 12439–12447, 2024a.

Huy Hoang, Tien Mai, and Pradeep Varakantham. Uniq: Offline inverse q-learning for avoiding undesirable demonstrations. *arXiv preprint arXiv:2410.08307*, 2024b.

Huy Hoang, Tien Anh Mai, and Pradeep Varakantham. SPRINQL: Sub-optimal demonstrations driven offline imitation learning. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024c. URL https://openreview.net/forum?id=uDD44NROOt.

Youngsoo Jang, Geon-Hyeong Kim, Jongmin Lee, Sungryull Sohn, Byoungjip Kim, Honglak Lee, and Moontae Lee. Safedice: offline safe imitation learning with non-preferred demonstrations. *Advances in Neural Information Processing Systems*, 36, 2024.

Yachen Kang, Diyuan Shi, Jinxin Liu, Li He, and Donglin Wang. Beyond reward: Offline preference-guided policy optimization. In *International Conference on Machine Learning*, pp. 15753–15768. PMLR, 2023.

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.

Geon-Hyeong Kim, Seokin Seo, Jongmin Lee, Wonseok Jeon, HyeongJoo Hwang, Hongseok Yang, and Kee-Eung Kim. Demodice: Offline imitation learning with supplementary imperfect demonstrations. In *International Conference on Learning Representations*, 2021.

Geon-Hyeong Kim, Jongmin Lee, Youngsoo Jang, Hongseok Yang, and Kee-Eung Kim. Lobsdice: Offline learning from observation via stationary distribution correction estimation. *Advances in Neural Information Processing Systems*, 35:8252–8264, 2022.

Ilya Kostrikov, Ofir Nachum, and Jonathan Tompson. Imitation learning via off-policy distribution matching. In *International Conference on Learning Representations*, 2020.

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

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.

Jongmin Lee, Wonseok Jeon, Byungjun Lee, Joelle Pineau, and Kee-Eung Kim. Optidice: Offline policy optimization via stationary distribution correction estimation. In *International Conference on Machine Learning*, pp. 6120–6130. PMLR, 2021.

Ziniu Li, Tian Xu, Zeyu Qin, Yang Yu, and Zhi-Quan Luo. Imitation learning from imperfection: Theoretical justifications and algorithms. In *Advances in Neural Information Processing Systems 37*, 2023.

Yuxiao Lu, Arunesh Sinha, and Pradeep Varakantham. Semantic loss guided data efficient supervised fine tuning for safe responses in LLMs. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=kO0DgO07hW.

Yecheng Ma, Andrew Shen, Dinesh Jayaraman, and Osbert Bastani. Versatile offline imitation from observations and examples via regularized state-occupancy matching. In *International Conference on Machine Learning*, pp. 14639–14663. PMLR, 2022.

Liyuan Mao, Haoran Xu, Weinan Zhang, and Xianyuan Zhan. ODICE: Revealing the mystery of distribution correction estimation via orthogonal-gradient update. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=L8UNn7Llt4.

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.

Tong Mu and Others. Rule based rewards for language model safety. In *NeurIPS 2024*, 2024. URL https://arxiv.org/abs/2411.01111. ArXiv preprint arXiv:2411.01111.

Vivek Myers, Erdem Biyik, Nima Anari, and Dorsa Sadigh. Learning multimodal rewards from rankings. In *Conference on robot learning*, pp. 342–352. PMLR, 2022.

Ofir Nachum, Bo Dai, Ilya Kostrikov, Yinlam Chow, Lihong Li, and Dale Schuurmans. Algaedice: Policy gradient from arbitrary experience. *arXiv preprint arXiv:1912.02074*, 2019.

Martin L Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

Siddharth Reddy, Anca D Dragan, and Sergey Levine. Sqil: Imitation learning via reinforcement learning with sparse rewards. *arXiv preprint arXiv:1905.11108*, 2019.

Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 627–635. JMLR Workshop and Conference Proceedings, 2011.

Harshit Sikchi, Qinqing Zheng, Amy Zhang, and Scott Niekum. Dual rl: Unification and new methods for reinforcement and imitation learning. In *Proceedings of the 12th International Conference on Learning Representations (ICLR)*, 2024. URL https://openreview.net/forum?id=xt9Bu66rqv.

Peter Sunehag, Guy Lever, Audrunas Gruslys, Wojciech Marian Czarnecki, Vinicius Zambaldi, Max Jaderberg, Marc Lanctot, Nicolas Sonnerat, Joel Z Leibo, Karl Tuyls, et al. Value-decomposition networks for cooperative multi-agent learning. *arXiv preprint arXiv:1706.05296*, 2017.

Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. The MIT Press, second edition, 2018. URL http://incompleteideas.net/book/the-book-2nd.html.

Faraz Torabi, Garrett Warnell, and Peter Stone. Behavioral cloning from observation. *arXiv preprint arXiv:1805.01954*, 2018.

Yueh-Hua Wu, Nontawat Charoenphakdee, Han Bao, Voot Tangkaratt, and Masashi Sugiyama. Imitation learning from imperfect demonstration. In *International Conference on Machine Learning*, pp. 6818–6827. PMLR, 2019.

Haoran Xu, Xianyuan Zhan, Honglei Yin, and Huiling Qin. Discriminator-weighted offline imitation learning from suboptimal demonstrations. In *Proceedings of the 39th International Conference on Machine Learning*, pp. 24725–24742, 2022.

Sheng Yue, Jiani Liu, Xingyuan Hua, Ju Ren, Sen Lin, Junshan Zhang, and Yaoxue Zhang. How to leverage diverse demonstrations in offline imitation learning. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=oOlooUu2Sb.

Songyuan Zhang, Zhangjie Cao, Dorsa Sadigh, and Yanan Sui. Confidence-aware imitation learning from demonstrations with varying optimality. *Advances in Neural Information Processing Systems*, 34:12340–12350, 2021.

Tony Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. *Robotics: Science and Systems XIX*, 2023.

Henry Zhu, Justin Yu, Abhishek Gupta, Dhruv Shah, Kristian Hartikainen, Avi Singh, Vikash Kumar, and Sergey Levine. The ingredients of real world robotic reinforcement learning. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=rJe2syrtvS.

APPENDIX

This appendix includes the following materials:

Missing Proofs: Proofs omitted from the main paper are provided in Appendix A and some additional discussions B.

Experimental Details: We describe the following aspects in detail:

- Full pseudocode (Appendix C.1)
- Dataset construction (Appendix C.2)
- Baseline implementations (Appendix C.3)
- Hyperparameter selections (Appendix C.4)
- Computational resources (Appendix C.5)

Additional Experiments: We further present supplementary results:

- Effect of the size of the bad dataset (Appendix D.1)
- Effect of the number of expert demonstrations in the good dataset $\mathcal{B}^G$ (Appendix D.2)
- Discussion: How many bad trajectories in $\mathcal{B}^B$ are sufficient to replace a good trajectory in $\mathcal{B}^G$ for DualCOIL? (Appendix D.3)
- Comparison of advantage-weighted BC and Q-weighted BC for policy extraction (Appendix D.4)
- Performance across varying quality levels of the unlabeled dataset $\mathcal{B}^{\text{MIX}}$ (Appendix D.5)
- Comparison with adapted offline reinforcement learning methods (Appendix D.8)
- Discussion: distribution-matching vs. preference-based approaches (Appendix D.9)
- Additional comparison with "avoid-bad-only" baselines (Appendix D.10)
- Ablations and experiments with $\alpha > 1$ (Appendix D.11)
- Comparison between $L(Q, \pi)$ and its surrogate $\widetilde{L}(Q, \pi)$ (Appendix D.12)
- Sensitivity analysis of β (Appendix D.13)

Stress Tests: We conduct additional stress-test experiments by:

- Increasing the proportion of bad data in the unlabeled dataset $\mathcal{B}^{\text{MIX}}$ to very high levels (Appendix D.6)
- Generating more bad data to enlarge the bad dataset $\mathcal{B}^B$ (Appendix D.7)

CONTENTS

A MISSING PROOFS

Proposition equation 4.1: *If $\alpha \leq 1$, then the objective function $f(d^\pi) = D_{KL}(d^\pi \,\|\, d^G) - \alpha\, D_{KL}(d^\pi \,\|\, d^B)$ is convex in d^π.*

Proof. We write the objective function as:

$$\begin{aligned} f(d^\pi) &= \sum_{(s,a)\sim d^\pi} \log \frac{d^\pi(s,a)}{d^G(s,a)} - \alpha \sum_{(s,a)\sim d^\pi} \log \frac{d^\pi(s,a)}{d^B(s,a)} \\ &= \sum_{s,a} (1-\alpha) d^\pi(s,a) \log d^\pi(s,a) + d^\pi(s,a)(\alpha d^B(s,a) - d^G(s,a)) \end{aligned} \tag{10}$$

We can see that the first term is convex in d^π since $\alpha \leq 1$ and $d^\pi(s,a)\log d^\pi(s,a)$ is convex in d^π. Moreover, the second term is linear in d^π. This implies that $f(d^\pi)$ is convex in π if $\alpha \leq 1$, as desired.

□

Proposition 4.2: *The objective function in equation 2 can be written as: $f(d,\pi) = (1-\alpha) D_{KL}(d||d^U) - \mathbb{E}_{(s,a)\sim d}\left[\Psi(s,a)\right]$, where $\Psi(s,a) = \log \frac{d^G(s,a)}{d^U(s,a)} - \alpha \log \frac{d^B(s,a)}{d^U(s,a)}$.*

Proof. We can expand the objective function as:

$$f(d,\pi) = \mathbb{E}_{(s,a)\sim d}\left[\log \frac{d(s,a)}{d^G(s,a)}\right] - \alpha\, \mathbb{E}_{(s,a)\sim d}\left[\log \frac{d(s,a)}{d^B(s,a)}\right].$$

We can rewrite the objective using d^U as an intermediate distribution:

$$\begin{aligned} f(d,\pi) &= \mathbb{E}_{(s,a)\sim d}\left[\log \frac{d(s,a)}{d^G(s,a)}\right] - \alpha\, \mathbb{E}_{(s,a)\sim d}\left[\log \frac{d(s,a)}{d^B(s,a)}\right] \\ &= \mathbb{E}_{(s,a)\sim d}\left[\log \frac{d(s,a)}{d^U(s,a)} + \log \frac{d^U(s,a)}{d^G(s,a)}\right] - \alpha\, \mathbb{E}_{(s,a)\sim d}\left[\log \frac{d(s,a)}{d^U(s,a)} + \log \frac{d^U(s,a)}{d^B(s,a)}\right] \\ &= (1-\alpha)\, \mathbb{E}_{(s,a)\sim d}\left[\log \frac{d(s,a)}{d^U(s,a)}\right] - \mathbb{E}_{(s,a)\sim d}\left[\Psi(s,a)\right], \\ &= (1-\alpha) D_{KL}(d||d^U) - \mathbb{E}_{(s,a)\sim d}\left[\Psi(s,a)\right] \end{aligned}$$

where $\Psi(s,a) = \log \frac{d^G(s,a)}{d^U(s,a)} - \alpha \log \frac{d^B(s,a)}{d^U(s,a)}$. □

Proposition 4.3: *Let the surrogate objective be defined as:*

$$\widetilde{L}(Q,\pi) = (1-\gamma)\, \mathbb{E}_{s\sim p_0}\left[V_Q^\pi(s)\right] - \mathbb{E}_{d^U}\left[\delta(s,a)\mathcal{T}^\pi[Q](s,a)\right] + (1-\alpha)\mathbb{E}_{d^U}\left[\delta(s,a)\right]. \tag{11}$$

where $\delta(s,a) = \exp\left(\frac{\Psi(s,a)}{1-\alpha}\right)$. Then $\widetilde{L}(Q,\pi)$ is a lower bound of $L(Q,\pi)$, with equality when $\mathcal{T}^\pi[Q](s,a) = 0$ for all (s,a).

Proof. We first write $L(Q,\pi)$ as:

$$\begin{aligned} L(Q,\pi) &= (1-\gamma)\, \mathbb{E}_{s\sim p_0}\left[V_Q^\pi(s)\right] \\ &\quad + (1-\alpha)\mathbb{E}_{(s,a)\sim d^U}\left[\exp\left(\frac{\Psi(s,a) - \mathcal{T}^\pi[Q](s,a)}{1-\alpha}\right)\right] \\ &= (1-\gamma)\, \mathbb{E}_{s\sim p_0}\left[V_Q^\pi(s)\right] \\ &\quad + (1-\alpha)\mathbb{E}_{(s,a)\sim d^U}\left[\exp\left(\frac{\Psi(s,a)}{1-\alpha}\right)\exp\left(\frac{-\mathcal{T}^\pi[Q](s,a)}{1-\alpha}\right)\right] \\ &= (1-\gamma)\, \mathbb{E}_{s\sim p_0}\left[V_Q^\pi(s)\right] \\ &\quad + (1-\alpha)\mathbb{E}_{(s,a)\sim d^U}\left[\delta(s,a)\exp\left(\frac{-\mathcal{T}^\pi[Q](s,a)}{1-\alpha}\right)\right], \end{aligned}$$

where we define $\delta(s,a) := \exp\left(\frac{\Psi(s,a)}{1-\alpha}\right)$.

Now, we use the inequality $e^t \geq t+1$ (which follows from the convexity of e^t and is tight at $t=0$), to obtain:

$$\exp\left(\frac{-\mathcal{T}^\pi[Q](s,a)}{1-\alpha}\right) \geq -\frac{\mathcal{T}^\pi[Q](s,a)}{1-\alpha} + 1.$$

Substituting this into the expression for $L(Q,\pi)$, we get:

$$L(Q,\pi) \geq (1-\gamma)\,\mathbb{E}_{s\sim p_0}\left[V_Q^\pi(s)\right] + (1-\alpha)\mathbb{E}_{(s,a)\sim d^U}\left[\delta(s,a)\left(-\frac{\mathcal{T}^\pi[Q](s,a)}{1-\alpha}+1\right)\right] =: \widetilde{L}(Q,\pi).$$

Equality holds in the inequality $e^t \geq t+1$ when $t=0$, which corresponds to $\mathcal{T}^\pi[Q](s,a)=0$. That is, the equality $L(Q,\pi)=\widetilde{L}(Q,\pi)$ holds when the rewards represented by the Q-function are zero everywhere. This completes the proof. □

Proposition 4.4: *The following properties hold:*

(i) $\widetilde{L}(Q,\pi)$ *is linear in* Q *and concave in* π*. As a result, the max–min optimization can be equivalently reformulated as a min–max problem:* $\max_\pi \min_Q \widetilde{L}(Q,\pi) = \min_Q \max_\pi \widetilde{L}(Q,\pi)$.

(ii) The min–max problem $\min_Q \max_\pi \widetilde{L}(Q,\pi)$ *reduces to the following non-adversarial problem:*

$$\min_Q \left\{\widetilde{L}(Q) = (1-\gamma)\,\mathbb{E}_{s\sim p_0}\left[V_Q(s)\right] - \mathbb{E}_{(s,a)\sim d^U}\left[\exp\left(\frac{\Psi(s,a)}{1-\alpha}\right)\mathcal{T}[Q](s,a)\right]\right\},$$

where the soft value function $V_Q(s)$ *is defined as:* $V_Q(s) = \beta\log\left(\sum_a \mu^U(a|s)\exp(Q(s,a)/\beta)\right)$*, and the soft Bellman residual operator is given by:* $\mathcal{T}[Q](s,a) = Q(s,a) - \gamma V_Q(s)$*. Moreover* $\widetilde{L}(Q)$ *is convex in* Q*.*

Proof. We first write $\widetilde{L}(Q,\pi)$ as:

$$\begin{aligned}\widetilde{L}(Q,\pi) = (1-\gamma)\,\mathbb{E}_{s\sim p_0}\left[V_Q^\pi(s)\right] - \mathbb{E}_{(s,a)\sim d^U}\left[\delta(s,a)\left(Q(s,a) - \gamma\mathbb{E}_{s'}\left[V_Q^\pi(s')\right]\right)\right] \\ + (1-\alpha)\mathbb{E}_{(s,a)\sim d^U}\left[\delta(s,a)\right],\end{aligned}$$

where we recall that

$$V_Q^\pi(s) = \mathbb{E}_{a\sim\pi(\cdot|s)}\left[Q(s,a) - \beta\log\frac{\pi(a\mid s)}{\mu^U(a\mid s)}\right].$$

Thus, we can observe that $\widetilde{L}(Q,\pi)$ is linear in Q.

Moreover, the function $V_Q^\pi(s)$ is concave in π, since it is composed of the expectation over a linear function of π (through $Q(s,a)$) and the negative entropy-regularized KL-divergence term, which is convex in π and thus its negative is concave. That is,

$$V_Q^\pi(s) = \mathbb{E}_{a\sim\pi(\cdot|s)}\left[Q(s,a) - \beta\log\frac{\pi(a\mid s)}{\mu^U(a\mid s)}\right]$$

is concave in π.

Furthermore, since $\delta(s,a)>0$, the coefficients associated with $V_Q^\pi(s)$ in $\widetilde{L}(Q,\pi)$ are non-negative. This implies that the entire function $\widetilde{L}(Q,\pi)$ is concave in π.

Now, since $\widetilde{L}(Q,\pi)$ is concave in π and linear in Q, we can apply the minimax theorem to swap the order of the max and min:

$$\max_\pi \min_Q \widetilde{L}(Q,\pi) = \min_Q \max_\pi \widetilde{L}(Q,\pi).$$

This holds because the function $\widetilde{L}(Q,\pi)$ satisfies the standard conditions of the minimax theorem: it is concave in π, convex (in fact, linear) in Q, and the optimization domains are convex.

Next, observe that in $\widetilde{L}(Q, \pi)$, the variable π only appears through the term $V_Q^\pi(s)$, and all coefficients multiplying $V_Q^\pi(s)$ are non-negative. Therefore, maximizing $\widetilde{L}(Q, \pi)$ over π is equivalent to maximizing $V_Q^\pi(s)$ for each state s independently. That is,

$$\max_\pi \widetilde{L}(Q, \pi) \equiv \max_\pi \sum_s c(s) V_Q^\pi(s),$$

for some non-negative coefficients $c(s) \geq 0$, which implies it suffices to solve $\max_\pi V_Q^\pi(s)$ pointwise.

Recall the definition:

$$V_Q^\pi(s) = \mathbb{E}_{a \sim \pi(\cdot|s)} \left[Q(s, a) - \beta \log \frac{\pi(a \mid s)}{\mu^U(a \mid s)} \right].$$

The inner maximization over $\pi(\cdot \mid s)$ is a standard entropy-regularized problem, and the optimal policy has the closed-form solution:

$$\pi^*(a \mid s) = \frac{\mu^U(a \mid s) \exp\left(\frac{Q(s,a)}{\beta}\right)}{\sum_{a'} \mu^U(a' \mid s) \exp\left(\frac{Q(s,a')}{\beta}\right)}.$$

This is a weighted softmax over $Q(s, a)$ values, using the baseline distribution $\mu^U(a \mid s)$ as the reference. Substituting this back into $V_Q^\pi(s)$ yields the closed-form maximized value:

$$\max_\pi V_Q^\pi(s) = \beta \log \left(\sum_a \mu^U(a \mid s) \exp\left(\frac{Q(s, a)}{\beta}\right) \right).$$

Thus:

$$\min_Q \max_\pi \widetilde{L}(Q, \pi) = \min_Q \widetilde{L}(Q)$$

where

$$\widetilde{L}(Q) = (1 - \gamma)\, \mathbb{E}_{s \sim p_0} \left[V_Q(s) \right] - \mathbb{E}_{(s,a) \sim d^U} \left[\exp\left(\frac{\Psi(s, a)}{1 - \alpha}\right) (Q(s, a) - \gamma\, \mathbb{E}_{s'}[V_Q(s')]) \right],$$

and

$$V_Q(s) = \beta \log \sum_a \mu^U(a \mid s) \exp\left(\frac{Q(s, a)}{\beta}\right).$$

We can now see that $\widetilde{L}(Q)$ is convex in Q, due to the following reasons:

- The function $Q(s, a) \mapsto \log \sum_a \mu^U(a \mid s) \exp\left(\frac{Q(s,a)}{\beta}\right)$ is a softmax (log-sum-exp), which is convex.
- $V_Q(s)$, being a composition of a convex function with an affine transformation, is convex in Q.
- Expectations over convex functions (e.g., $\mathbb{E}_{s \sim p_0}[V_Q(s)]$, $\mathbb{E}_{s'}[V_Q(s')]$) preserve convexity.
- The remaining terms in $\widetilde{L}(Q)$, such as $Q(s, a)$, appear linearly and thus preserve convexity.

Hence, the overall objective $\widetilde{L}(Q)$ is convex in Q, which completes the proof.

□

Proposition 5.1 *The following Q-weighted behavior cloning (BC) objective yields the same optimal policy as the original advantage-weighted BC formulation in equation 8:*

$$\max_\pi \sum_{(s,a) \sim \mathcal{B}^U} \exp\left(\frac{1}{\beta} Q(s, a)\right) \log \pi(a \mid s). \tag{12}$$

Proof. The Q-weighted BC objective can be written as:

$$\max_{\pi} \sum_{(s,a)} \mu^U(s,a) \exp\left(\frac{1}{\beta} Q(s,a)\right) \log \pi(a \mid s).$$

This represents a weighted maximum likelihood objective, where the weights are shaped by the exponential of the Q-values. For each state s, the optimal solution $\pi^*(a \mid s)$ is given by:

$$\pi^*(a \mid s) = \frac{\mu^U(s,a) \exp\left(\frac{1}{\beta} Q(s,a)\right)}{\sum_{a'} \mu^U(s,a') \exp\left(\frac{1}{\beta} Q(s,a')\right)}.$$

Moreover, we recall that:

$$V^Q(s) = \beta \log\left(\sum_{a'} \mu^U(s,a') \exp\left(\frac{1}{\beta} Q(s,a')\right)\right),$$

which allows us to express the optimal policy in terms of the advantage $Q(s,a) - V^Q(s)$ as:

$$\pi^*(a \mid s) = \mu^U(s,a) \exp\left(\frac{1}{\beta}(Q(s,a) - V^Q(s))\right).$$

This is precisely the optimal policy corresponding to the advantage-weighted BC objective defined in equation 8. This completes the proof.

□

B Additional Discussions

B.1 Derivation of the Q-learning Objective

We provide a detailed derivation of the Q-learning objective presented in Section 4.1 of the main paper, starting from the primal objective in Proposition 4.2. The optimization problem can be written as:

$$\begin{aligned} &\min_{d,\pi} (1-\alpha) D_{\mathrm{KL}}(d \,||\, d_U) - \mathbb{E}_{(s,a)\sim d}[\Psi(s,a)] \\ &\text{s.t.} \quad d(s,a) = (1-\gamma) p_0(s) \pi(a|s) + \gamma \pi(a|s) \sum_{s',a'} T(s|s',a') d(s',a'), \end{aligned} \tag{13}$$

where $\Psi(s,a) = \log \frac{d_G(s,a)}{d_U(s,a)} - \alpha \log \frac{d_B(s,a)}{d_U(s,a)}$. This objective is of the same form as the primal regularized RL problem analyzed in Sikchi et al. (2024), with reward $r(s,a) = \Psi(s,a)$, reference distribution $d_{\text{ref}} = d_U$, and regularization weight $(1-\alpha)$. Following the derivation in DualRL, we introduce a Lagrange multiplier $Q(s,a)$ for the occupancy-flow constraint and form the Lagrangian:

$$\begin{aligned} \mathcal{L}(d,\pi,Q) = (1-\alpha)\, \mathbb{E}_d\left[\log \tfrac{d(s,a)}{d_U(s,a)}\right] - \mathbb{E}_d[\Psi(s,a)] \\ + \mathbb{E}_d\big[Q(s,a) - \gamma\, \mathbb{E}_{T,\pi}[Q(s',a')]\big] - (1-\gamma)\, \mathbb{E}_{p_0,\pi}[Q(s,a)]. \end{aligned} \tag{14}$$

Minimizing equation 14 with respect to $d(s,a)$ pointwise gives

$$\frac{\partial \mathcal{L}}{\partial d(s,a)} = 0 \quad \Rightarrow \quad \log \frac{d^*(s,a)}{d_U(s,a)} = \frac{1}{1-\alpha}\big[\Psi(s,a) + \gamma\, \mathbb{E}_{T,\pi}[Q(s',a')] - Q(s,a)\big] + c,$$

which leads to the optimal occupancy distribution

$$d^*(s,a) \propto d_U(s,a) \exp\left(\frac{\Psi(s,a) + \gamma\, \mathbb{E}_{T,\pi}[Q(s',a')] - Q(s,a)}{1-\alpha}\right). \tag{15}$$

Substituting equation 15 back into the Lagrangian eliminates the dependence on d, yielding the following dual optimization problem:

$$\begin{aligned} \max_{\pi} \min_{Q} \Big[(1-\gamma)\, \mathbb{E}_{p_0,\pi}[Q(s,a)] \\ + (1-\alpha)\, \mathbb{E}_{(s,a)\sim d_U}\left[\exp\left(\frac{\Psi(s,a) + \gamma\, \mathbb{E}_{T,\pi}[Q(s',a')] - Q(s,a)}{1-\alpha}\right)\right]\Big]. \end{aligned} \tag{16}$$

Equation 16 corresponds exactly to the Q-learning formulation presented in Eq. 3.

Here we note that we adopt the DualRL framework because it provides a principled and unified treatment of regularized reinforcement learning objectives. In particular, DualRL optimizes a KL-regularized objective that naturally arises from our derivation in Proposition 4.2, ensuring theoretical consistency between the recovered reward model and the downstream policy optimization. Moreover, DualRL has demonstrated strong empirical performance across diverse benchmarks and was highlighted as an ICLR 2024 spotlight presentation. Its stability and efficiency make it a more advanced and reliable choice compared to earlier DICE-based estimators or Q-learning variants such as OptDICE or XQL. Integrating DualRL therefore enables us to fully leverage our implicit reward formulation while benefiting from a state-of-the-art optimization framework.

B.2 A Note on DualCOIL under f-Divergence

We note that the convexity stated in Proposition 4.1 does not hold under arbitrary f-divergences, even under the same assumptions. To illustrate this, consider the following objective defined using an f-divergence:

$$F(d^\pi) = D_f(d^\pi \,\|\, d^G) - \alpha\, D_f(d^\pi \,\|\, d^B),$$

which can be written as:

$$F(d^\pi) = \sum_{(s,a)} d^G(s,a) f\left(\frac{d^\pi(s,a)}{d^G(s,a)}\right) - \alpha\, d^B(s,a) f\left(\frac{d^\pi(s,a)}{d^B(s,a)}\right).$$

Observe that each term

$$d^G(s,a) f\left(\frac{d^\pi(s,a)}{d^G(s,a)}\right) - \alpha\, d^B(s,a) f\left(\frac{d^\pi(s,a)}{d^B(s,a)}\right)$$

is not necessarily convex for any $\alpha > 0$. Whether this expression is convex depends on the values of $d^G(s,a)$ and $d^B(s,a)$. In particular, if $d^G(s,a) = 0$—i.e., the state-action pair (s,a) is never visited by the expert policy—then the term may become concave. Therefore, in general, the objective $F(d^\pi)$ defined under an f-divergence is not convex in d^π for arbitrary choices of α. Thus, the standard Lagrangian duality cannot be applied. For this reason, the KL divergence appears to be an ideal choice for our problem of learning from both expert and undesirable demonstrations.

B.3 Possible Failure Modes in Avoiding Bad Demonstrations

A known challenge in pushing the policy distribution away from the mean of the bad dataset is the potential emergence of new undesirable behaviors not covered by the dataset—often referred to as a "whack-a-mole" problem. If the bad dataset is incomplete, the policy may still converge to harmful behaviors. In this context, the primal objective $D_{KL}(d^\pi \,\|\, d^G) - D_{KL}(d^\pi \,\|\, d^B)$ highlights the importance of expert demonstrations, as safe learning requires that the influence of expert behavior outweighs that of bad behavior.

Our framework addresses this by prioritizing imitation of expert behavior whenever available, while using bad demonstrations only to avoid clearly undesirable actions. Thus, expert data anchors the policy, and bad data serves as a supplementary signal rather than requiring exhaustive coverage of all failure modes. This mitigates the "whack-a-mole" issue by ensuring the policy remains primarily guided by expert behavior.

Naturally, when both expert and bad demonstrations are scarce, policy learning becomes difficult—a limitation shared by most IL approaches. Nonetheless, our experiments show that the method is robust and consistently outperforms baselines, even with only limited expert data.

B.4 Tightness of the Surrogate Lower Bound and the Role of Regularization

In our algorithm, we adopt the surrogate objective $\widetilde{L}(Q,\pi)$ as a tractable lower bound of the true training objective $L(Q,\pi)$. A natural concern is: *How tight is the lower-bound objective $\widetilde{L}(Q,\pi)$ compared to the original objective $L(Q,\pi)$?* While the gap between $L(Q,\pi)$ and its surrogate can be nontrivial—reflecting the difference between the exponential function and its linear approximation—this does not undermine its effectiveness. The surrogate offers tractability while still guiding

the optimization of Q and π in a direction consistent with maximizing the original objective, since both e^x and $x + 1$ share the same monotonicity. Appendix D.12 provides a detailed comparison, and ablation studies confirm that the surrogate leads to significantly improved training performance.

Another question is whether DualCOIL benefits primarily from the implicit regularization within $L(Q, \pi)$ rather than being a faithful proxy for the original $D_{KL}(d^\pi \,\|\, d^G) - D_{KL}(d^\pi \,\|\, d^B)$ objective. In practice, this regularization mainly stabilizes training by preventing extreme Q-values, a technique also found in baselines such as SafeDICE and DWBC. However, regularization alone does not enable meaningful learning from both expert and undesirable datasets. The superior performance of DualCOIL arises instead from the structure of $L(Q, \pi)$ itself, which is grounded in the original KL-divergence formulation.

C EXPERIMENT SETTINGS

C.1 FULL PSEUDO CODE

The detailed implementation are provided in Algorithm 2.

Algorithm 2 DualCOIL: Offline Imitation Learning from Contrasting Behaviors (full)

Require: Good dataset $\mathcal{B}_G$, Bad dataset $\mathcal{B}_B$, unlabeled dataset $\mathcal{B}_U$
Require: Hyperparameters: $\alpha \in [0, 1)$, β, γ, N_μ, N, target update rate τ, batch size B
Initialize networks: $Q_{w_q}(s, a)$, $V_{w_v}(s)$, $\pi_\theta(a|s)$, classifiers $c^G_{w_G}(s, a)$, $c^B_{w_B}(s, a)$
Initialize target Q-network: $Q_{\text{target}} \leftarrow Q_{w_q}$

Step 1: Estimate occupancy ratios
for $i = 1$ to N_μ **do**
  Sample batch $\{(s^G_i)'\}^B_{i=1} \sim \mathcal{B}_G$; $\{(s^B_i)'\}^B_{i=1} \sim \mathcal{B}_B$; $\{(s^U_i)'\}^B_{i=1} \sim \mathcal{B}_U$
  Update $c^G_{w_G}$ by maximizing the objective in equation 6.
  Update $c^B_{w_B}$ by maximizing an analogous objective to equation 6 for the bad dataset.
end for

Step 2: Calculate Ψ function
Calculate $\Psi(s, a) = \log\left(\frac{c^G_{w_G}(s')}{1-c^G_{w_G}(s')}\right) - \alpha \log\left(\frac{c^B_{w_B}(s')}{1-c^B_{w_B}(s')}\right)$.

Step 3: Train Q, V, and Policy
for $i = 1$ to N **do**
  Sample batch $\{(s_i, a_i, s'_i, \Psi_i)\}^B_{i=1} \sim \mathcal{B}_U$
  Q-Update: Minimize the objective $\tilde{L}(Q_{w_q}|V_{w_v}) + \frac{1}{2}(Q_{w_q}(s_i, a_i) - \gamma V_{w_v}(s'_i))^2$.
  (reference: $\tilde{L}(Q|V)$ from Sec 5/ Eq equation 7)
  V-Update: Minimize the Extreme-V objective:

$$\min_{w_v} \frac{1}{B} \sum_{i=1}^{B} \left[\exp\left(\frac{Q_{\text{target}}(s_i, a_i) - V_{w_v}(s_i)}{\beta} \right) - \frac{Q_{\text{target}}(s_i, a_i) - V_{w_v}(s_i)}{\beta} - 1 \right].$$

  Policy Update: Maximize the policy by using Q-weighted Behavior Cloning.
  (reference: Sec 5/ Eq equation 9)
  Target Q-Update: Soft update:$Q_{\text{target}} \leftarrow \tau Q_{w_q} + (1 - \tau) Q_{\text{target}}$
end for

return Trained policy π_θ

C.2 DATASET CONSTRUCTION

From the official D4RL dataset we use three different domains:

- MuJoCo Locomotion[CHEETAH,ANT,HOPPER,WALKER] with three types of dataset:
  - EXPERT
  - MEDIUM
  - RANDOM
- Adroit [PEN,HAMMER,DOOR,RELOCATE] with three types of dataset:
  - EXPERT
  - HUMAN
  - CLONED
- FrankaKitchen [KITCHEN] with three types of dataset:
  - COMPLETE
  - MIXED
  - PARTIAL

Following the approach of (Sikchi et al., 2024), we also provide several combinations across all three domains, as shown in Table 2. Notably, the unlabeled dataset $\mathcal{B}^{\text{MIX}}$ is constructed by combining the entire suboptimal dataset with the expert dataset, resulting in an overlap between $\mathcal{B}^B$ and $\mathcal{B}^{\text{MIX}}$. Nevertheless, this setup is practical: given an good dataset $\mathcal{B}^G$ and an unlabeled dataset $\mathcal{B}^{\text{MIX}}$, users can randomly sample trajectories and assign them to either $\mathcal{B}^G$ or $\mathcal{B}^B$ without the need for any additional external data.

Task	Unlabeled name	$\mathcal{B}^G$	$\mathcal{B}^B$	$\mathcal{B}^{\text{MIX}}$
CHEETAH	RANDOM+EXPERT	1 EXPERT	10 RANDOM	Full RANDOM+30 EXPERT
	MEDIUM+EXPERT	1 EXPERT	10 MEDIUM	Full MEDIUM+30 EXPERT
ANT	RANDOM+EXPERT	1 EXPERT	10 RANDOM	Full RANDOM+30 EXPERT
	MEDIUM+EXPERT	1 EXPERT	10 MEDIUM	Full MEDIUM+30 EXPERT
HOPPER	RANDOM+EXPERT	1 EXPERT	10 RANDOM	Full RANDOM+30 EXPERT
	MEDIUM+EXPERT	1 EXPERT	10 MEDIUM	Full MEDIUM+30 EXPERT
WALKER	RANDOM+EXPERT	1 EXPERT	10 RANDOM	Full RANDOM+30 EXPERT
	MEDIUM+EXPERT	1 EXPERT	10 MEDIUM	Full MEDIUM+30 EXPERT
PEN	CLONED+EXPERT	1 EXPERT	25 CLONED	Full CLONED+100 EXPERT
	HUMAN+EXPERT	1 EXPERT	25 HUMAN	Full HUMAN+100 EXPERT
HAMMER	CLONED+EXPERT	1 EXPERT	25 CLONED	Full CLONED+100 EXPERT
	HUMAN+EXPERT	1 EXPERT	25 HUMAN	Full HUMAN+100 EXPERT
DOOR	CLONED+EXPERT	1 EXPERT	25 CLONED	Full CLONED+100 EXPERT
	HUMAN+EXPERT	1 EXPERT	25 HUMAN	Full HUMAN+100 EXPERT
RELOCATE	CLONED+EXPERT	1 EXPERT	25 CLONED	Full CLONED+100 EXPERT
	HUMAN+EXPERT	1 EXPERT	25 HUMAN	Full HUMAN+100 EXPERT
KITCHEN	PARTIAL+COMPLETE	1 COMPLETE	25 PARTIAL	Full PARTIAL+1 COMPLETE
	MIXED+COMPLETE	1 COMPLETE	25 MIXED	Full MIXED+1 COMPLETE

Table 2: **Dataset Construction.** The numbers in Table 2 indicate the number of trajectories drawn from each corresponding dataset. For the KITCHEN task, we follow the setting of (Sikchi et al., 2024), where only a single trajectory from the COMPLETE dataset is included in $\mathcal{B}^{\text{MIX}}$.

C.3 Baselines Implementation

We compare our method against several established baselines. For methods with publicly available code, we utilized their official implementations without algorithmic modifications.

C.3.1 Behavior Cloning (BC)

We employ the standard Behavior Cloning (BC) objective, which aims to minimize the negative log-likelihood of the demonstrated actions under the learned policy:

$$\min_{\pi} -\mathbb{E}_{(s,a)\sim\mathcal{B}} \log \pi(a \mid s), \tag{17}$$

where $\mathcal{B}$ denotes the dataset of state-action pairs. Specifically, $\mathcal{B}$ corresponds to $\mathcal{B}^{\text{MIX}}$ in the case of BC-MIX, or $\mathcal{B}^{G}$ for BC-G.

C.3.2 Other Baselines with Official Implementations

For the following baselines, we used their official, unmodified implementations:

- **SMODICE** (Ma et al., 2022): Applied to both the good dataset ($\mathcal{B}^{G}$) and the mixed dataset ($\mathcal{B}^{\text{MIX}}$). The official code is available at [GitHub].
- **ILID** (Yue et al., 2024): Applied to $\mathcal{B}^{G}$ and $\mathcal{B}^{\text{MIX}}$. The official code is available at [GitHub].
- **ReCOIL** (Sikchi et al., 2024): Applied to $\mathcal{B}^{G}$ and $\mathcal{B}^{\text{MIX}}$. The official code is available at [GitHub].
- **SafeDICE** (Jang et al., 2024): Applied to the bad dataset ($\mathcal{B}^{B}$) and the mixed dataset ($\mathcal{B}^{\text{MIX}}$). The official code is available at [GitHub].

C.3.3 DWBC-GB

DWBC-GB is our adaptation of DWBC (Xu et al., 2022) (original official implementation: [GitHub]). While the original DWBC is designed for scenarios involving $\mathcal{B}^{G}$ and $\mathcal{B}^{\text{MIX}}$, our modified version, DWBC-GB, is extended to handle all three dataset types: $\mathcal{B}^{G}$, $\mathcal{B}^{B}$, and $\mathcal{B}^{\text{MIX}}$.

This adaptation involves training two discriminators: c^{G} for good data and c^{B} for bad data. Their respective loss functions are:

$$\begin{aligned} L_{c^G} = \eta\, \mathbb{E}_{(s,a)\sim\mathcal{B}^G}[-\log c^G(s,a,\log\pi(a|s))] \\ + \mathbb{E}_{(s,a)\sim\mathcal{B}^{\text{MIX}}}[-\log(1-c^G(s,a,\log\pi(a|s)))] \\ - \eta\, \mathbb{E}_{(s,a)\sim\mathcal{B}^G}[-\log(1-c^G(s,a,\log\pi(a|s)))], \end{aligned} \tag{18}$$

$$\begin{aligned} L_{c^B} = \eta\, \mathbb{E}_{(s,a)\sim\mathcal{B}^B}[-\log c^B(s,a,\log\pi(a|s))] \\ + \mathbb{E}_{(s,a)\sim\mathcal{B}^{\text{MIX}}}[-\log(1-c^B(s,a,\log\pi(a|s)))] \\ - \eta\, \mathbb{E}_{(s,a)\sim\mathcal{B}^B}[-\log(1-c^B(s,a,\log\pi(a|s)))]. \end{aligned} \tag{19}$$

The policy π is then learned by minimizing the objective:

$$\begin{aligned} \min_{\pi} \Bigg(\mathbb{E}_{(s,a)\sim\mathcal{B}^G}\left[-\log\pi(a|s)\cdot\left(\alpha - \frac{\eta}{c(s,a)\,(1-c(s,a))}\right)\right] \\ + \mathbb{E}_{(s,a)\sim\mathcal{B}^{\text{MIX}}}\left[-\log\pi(a|s)\cdot\frac{1}{1-c(s,a)}\right]\Bigg), \end{aligned} \tag{20}$$

where $c(s,a) = c^{G}(s,a) - c^{B}(s,a)$. (Note: η and α are hyperparameters.)

C.4 Hyper Parameters

Our method features two primary hyperparameters: α (weighting for balancing positive and negative samples) and β (Extreme-V update). Sections 6.4, D.11, and D.13 present ablation studies detailing the sensitivity to these parameters.

Task	Unlabeled name	α	β
CHEETAH	RANDOM+EXPERT	0.6	20.0
	MEDIUM+EXPERT	0.6	15.0
ANT	RANDOM+EXPERT	0.6	15.0
	MEDIUM+EXPERT	0.6	15.0
HOPPER	RANDOM+EXPERT	0.4	30.0
	MEDIUM+EXPERT	0.4	30.0
WALKER	RANDOM+EXPERT	0.6	20.0
	MEDIUM+EXPERT	0.6	20.0
PEN	CLONED+EXPERT	0.4	15.0
	HUMAN+EXPERT	0.4	10.0
HAMMER	CLONED+EXPERT	0.2	10.0
	HUMAN+EXPERT	0.6	20.0
DOOR	CLONED+EXPERT	0.4	15.0
	HUMAN+EXPERT	0.4	10.0
RELOCATE	CLONED+EXPERT	0.4	30.0
	HUMAN+EXPERT	0.8	3.0
KITCHEN	PARTIAL+COMPLETE	0.1	20.0
	MIXED+COMPLETE	0.3	20.0

Table 3: Hyper parameters.

Specific parameters for all tasks are provided in Table 3 below:

Beyond these, all other hyperparameters are consistently applied across all benchmarks and settings. The policy, Q-function, V-function, and discriminator all utilize a 2-layer feedforward neural network architecture with 256 hidden units and ReLU activation functions. For the policy, Tanh Gaussian outputs are used. The Adam optimizer is configured with a weight decay of 1×10^{-3}, all learning rates are set to 3×10^{-4}, mini batch size is 1024, and a soft critic update parameter $\tau = 0.005$ is used. These hyperparameters are summarized in Table 4:

Hyperparameter	**Value**
Network Architecture (Policy, Q-func, V-func, Discriminator)	2-layer Neural Network
Hidden Units per Layer	256
Batch size	1024
Activation Function (Hidden Layers)	ReLU
Policy Output Activation	Tanh Gaussian
Optimizer	Adam
Learning Rate (all networks)	3×10^{-4}
Weight Decay (Adam)	1×10^{-3}
Soft Critic Update Rate (τ)	0.005

Table 4: Consistent hyperparameters used across all benchmarks and settings.

C.5 Computational Resource

Our experiments were conducted using a pool of 12 NVIDIA GPUs, including L40, A5000, and RTX 3090 models. For each experimental configuration, five training seeds were executed in parallel, sharing a single GPU, eight CPU cores, and 64 GB of RAM. Under these shared conditions, completing 1 million training steps across all five seeds took approximately 30 minutes. The software environment was based on JAX version 0.4.28 (with CUDA 12 support), running on CUDA version 12.3.2 and cuDNN version 8.9.7.29.

Moreover, we evaluated all methods using the CHEETAH (RANDOM+EXPERT) task under identical hardware conditions: a single NVIDIA L40 GPU, 8 CPU cores, and 64GB of RAM. For SafeDICE, we were unable to utilize GPU acceleration with TensorFlow; consequently, the method was run in CPU-only mode, resulting in slower training times. For DualCOIL, we trained two discriminators,

which required approximately 5 minutes; this duration is included in the total training time. A complete training time for a single seed are reported in Table 5.

	DWBC-GB	SafeDICE (CPU)	SMODICE	ILID	ReCOIL	DualCOIL
time	∼130 mins	∼150 mins	∼110 mins	∼80 mins	∼20 mins	∼25 mins

Table 5: Comparison of training time across methods.

D ADDITIONAL EXPERIMENTS

D.1 IMPACT OF THE SIZE OF THE BAD DATASET: FULL DETAILS

To support the experiment in Section 6.3, we present the complete results for all MuJoCo Locomotion and Adroit manipulation tasks. In particular, we progressively increase the size of the suboptimal dataset $\mathcal{B}^B$ and evaluate the impact on each algorithm's performance. The results, shown in Figure 3, demonstrate that DualCOIL consistently outperforms all other baselines across all tasks, effectively leveraging the bad data to achieve superior performance. Notably, the results indicate that with only a single good trajectory in $\mathcal{B}^G$, increasing the number of bad trajectories in $\mathcal{B}^B$ to just 10 is sufficient for DualCOIL to achieve its highest performance across all tasks.

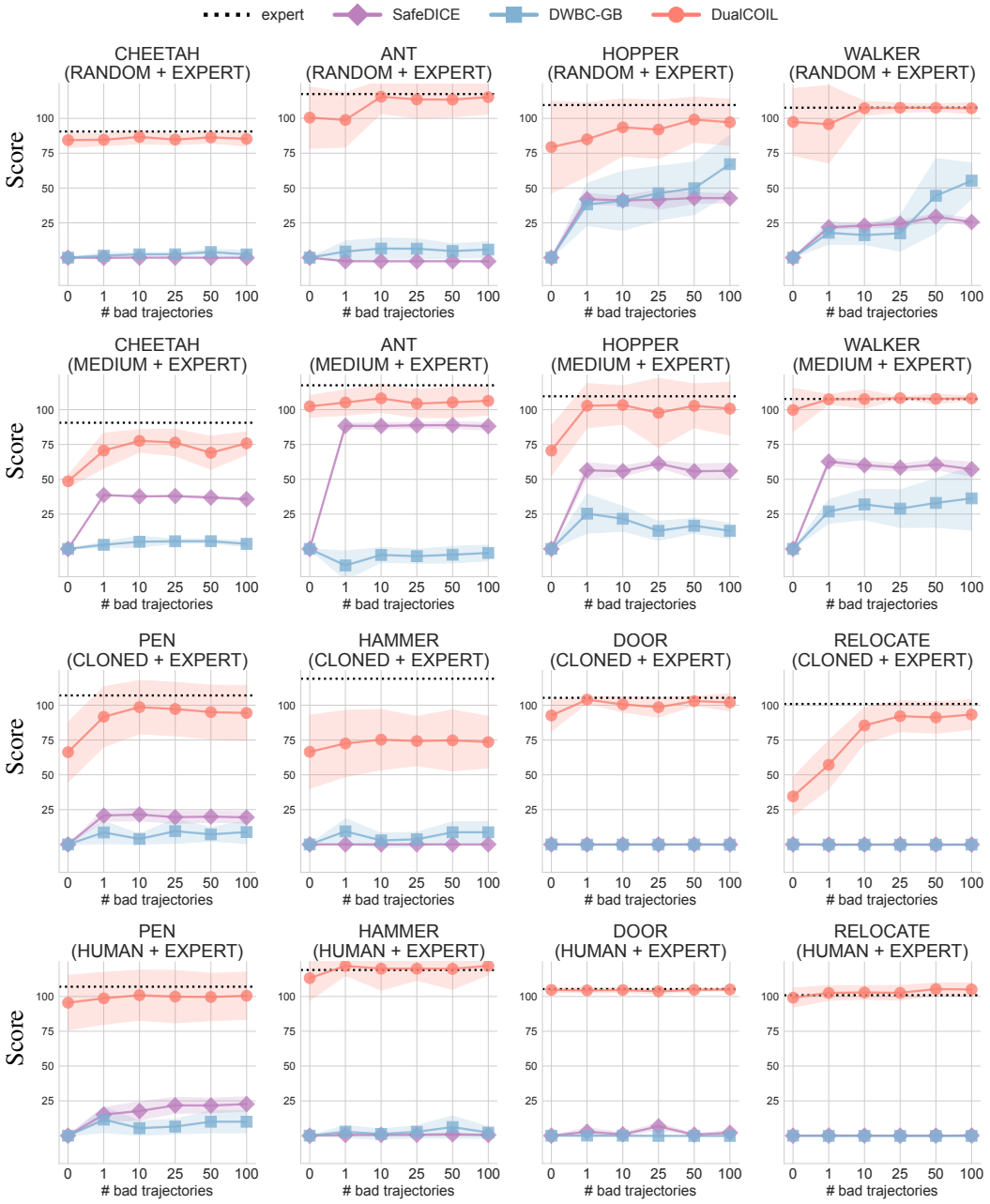

Figure 3: Full bad dataset size effect. SafeDICE and DWBC-GB do not have version that learn from 0 bad trajectory, we assign result 0.0 for them.

D.2 Impact of the Number of Expert Demonstrations in good dataset $\mathcal{B}^G$

In this section, we investigate how many expert trajectories in the good dataset $\mathcal{B}^G$ are sufficient to achieve optimal performance. To this end, the quantity of expert trajectories in $\mathcal{B}^G$ was incrementally increased through the set 1,3,5,10,25, while the composition of the unlabeled dataset ($\mathcal{B}^{\text{MIX}}$) remained fixed, as specified in Table 1. The detailed results are presented in Figure 4 and 5.

ILID performs well on the Mujoco locomotion tasks (CHEETAH, ANT, HOPPER, WALKER), but struggles in 3 out of 4 Adroit tasks (HAMMER, DOOR, RELOCATE). This indicates that ILID requires a sufficient number of expert trajectories to achieve stable expert performance, which is not met in the more complex Adroit tasks. In contrast, ReCOIL appears unable to effectively leverage the good data, as its performance does not improve significantly with more expert trajectories. Overall, DualCOIL demonstrates consistently strong performance, **requiring only 3 to 5 expert trajectories** to achieve near-optimal results in all tasks.

Discussion on the Use Cases of ILID and DualCOIL: Through this experiment, we observe that in the Mujoco tasks, ILID can outperform DualCOIL-G when the size of the good dataset is sufficiently large. This highlights a limitation of DualCOIL, where the policy extraction objective is defined as $\max_\pi \left\{\sum_{(s,a)\sim\mathcal{B}^U} \exp(\frac{1}{\beta}Q(s,a)) \log \pi(a|s)\right\}$. This objective uses data from the union dataset $\mathcal{B}^U$, which may assign high weights to poor-quality transitions, potentially harming training.

In contrast, ILID only retains transitions that are connected to good data and explicitly discards irrelevant or undesirable transitions (refer to the implementation details of ILID for more information). This targeted filtering strategy enables ILID to avoid the negative effects of poor transitions and scale more effectively with increasing amounts of good data.

These observations suggest a potential direction for improving DualCOIL by incorporating similar data filtering mechanisms. Specifically, enhancing DualCOIL to better isolate high-quality transitions could help it perform competitively with ILID in scenarios where the good dataset is large. We leave this exploration for future work, as it requires a careful study of how to construct an optimal dataset using Q-based methods.

In summary, ILID is a strong approach that scales well with the quality and size of the expert dataset. Practitioners may prefer discriminator-based methods like ILID when sufficient high-quality expert data is available, while DualCOIL remains a robust choice in settings where such data is limited and scalalbe with bad dataset.

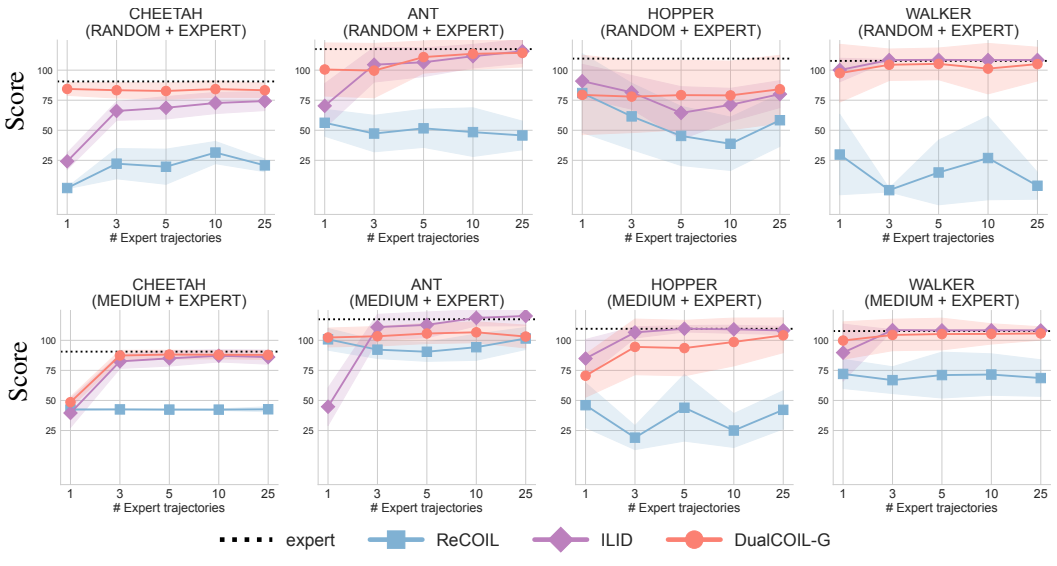

Figure 4: Different of good dataset size without impact from bad dataset in MuJoCo Locomotion tasks.

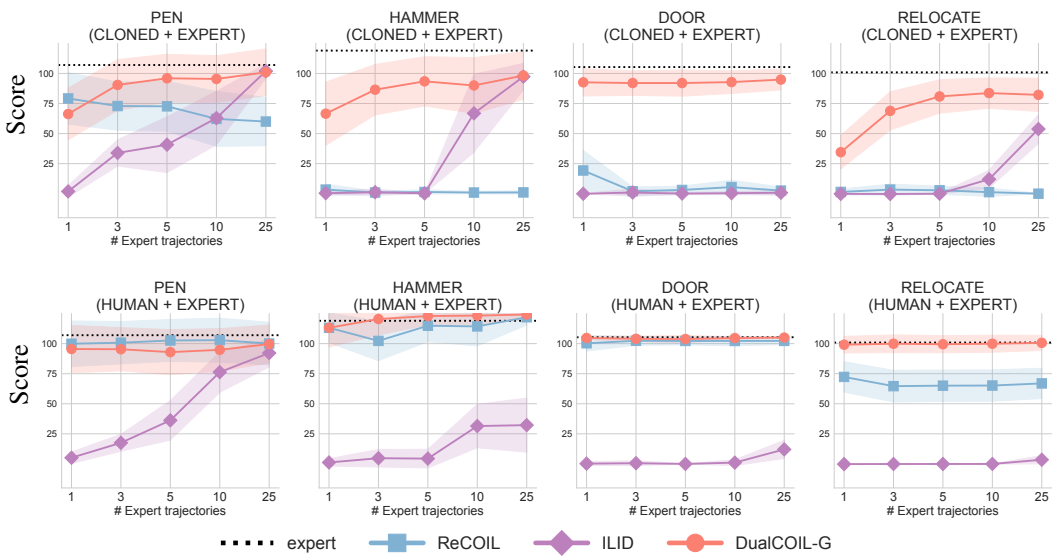

Figure 5: Different of good dataset size without impact from bad dataset in Adroit Manipulation tasks.

D.3 Discussion: How Many Bad Trajectories in $\mathcal{B}^B$ Are Sufficient to Replace a Good Trajectory in $\mathcal{B}^G$ for DualCOIL?

Based on the previous experiments:

- Section D.1 addresses the question: How does the size of the bad dataset $\mathcal{B}^B$ affect the performance of DualCOIL?
- Section D.2 investigates an additional question: How does the size of the good dataset $\mathcal{B}^G$ affect the performance of DualCOIL?

From these experiments, we derive the following observations:

- With only one good trajectory in $\mathcal{B}^G$, adding 10 bad trajectories in $\mathcal{B}^B$ is sufficient for DualCOIL to achieve its best performance.
- Without any bad data $\mathcal{B}^B$, 3 to 5 good trajectories in $\mathcal{B}^G$ are enough to reach peak performance.

These results suggest that DualCOIL can efficiently utilize bad data to reduce the need for good data, with an estimated ratio of 2 to 5 bad trajectories being roughly equivalent to one good trajectory across the benchmarks studied in this paper.

D.4 Comparison of Advantage-weighted BC and Q-weighted BC for the Policy Extraction

In this paper, we propose a novel policy extraction method called QW-BC (Objective equation 9), in contrast to prior approaches that rely on AW-BC (Objective equation 8). In this section, we present a comparison between QW-BC and AW-BC, as illustrated in Figure 6. Overall, QW-BC demonstrates superior policy extraction performance, attributed to its stability derived from relying on a single network estimation. In contrast, AW-BC often exhibits oscillations and instability, frequently assigning inconsistent and overly high weights to bad transitions.

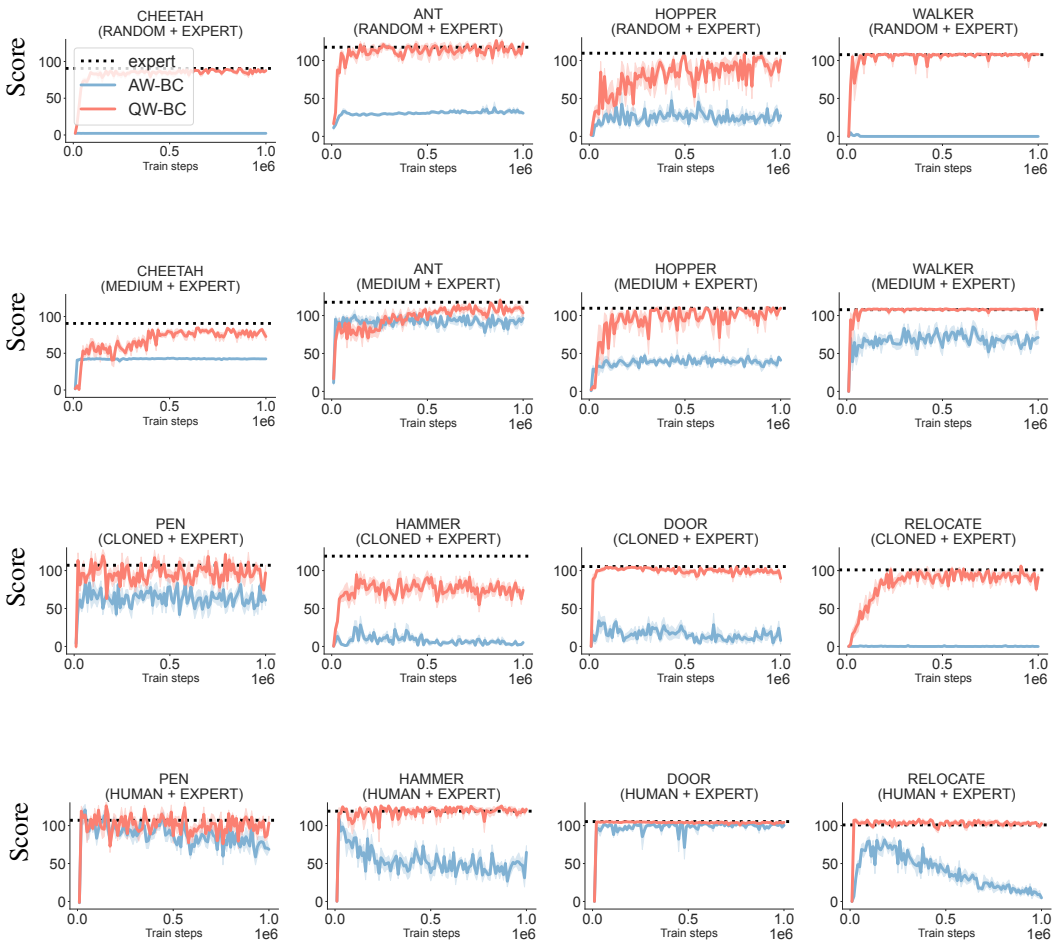

Figure 6: AW-BC and QW-BC comparison.

D.5 Performance Across Varying Quality Levels of the Unlabeled Dataset $\mathcal{B}^{\text{MIX}}$

The performance of all methods is influenced by the quality of the unlabeled dataset $\mathcal{B}^{\text{MIX}}$. To evaluate the robustness of our method under varying dataset quality, we conduct experiments with different amounts of expert trajectories combined with the full set of undesirable trajectories in the unlabeled dataset. We compare our approach against ILID and ReCOIL—which leverage $\mathcal{B}^{G}$ and $\mathcal{B}^{\text{MIX}}$—as well as SafeDICE, which learns from $\mathcal{B}^{B}$ and $\mathcal{B}^{\text{MIX}}$. The detailed results of this study are presented in Figure 7.

In the Mujoco locomotion tasks, increasing the quality of the unlabeled dataset has minimal effect on SafeDICE and ILID, and both methods continue to underperform on the Adroit hand manipulation tasks regardless of the number of expert trajectories included. In contrast, ReCOIL shows improved

performance as the quality of the unlabeled dataset increases, successfully learning 4 out of 8 tasks across both locomotion and manipulation domains. Overall, our method achieves near-expert performance on 7 out of 8 tasks while requiring significantly lower-quality unlabeled datasets $\mathcal{B}^{\text{MIX}}$, demonstrating its superior data efficiency and robustness.

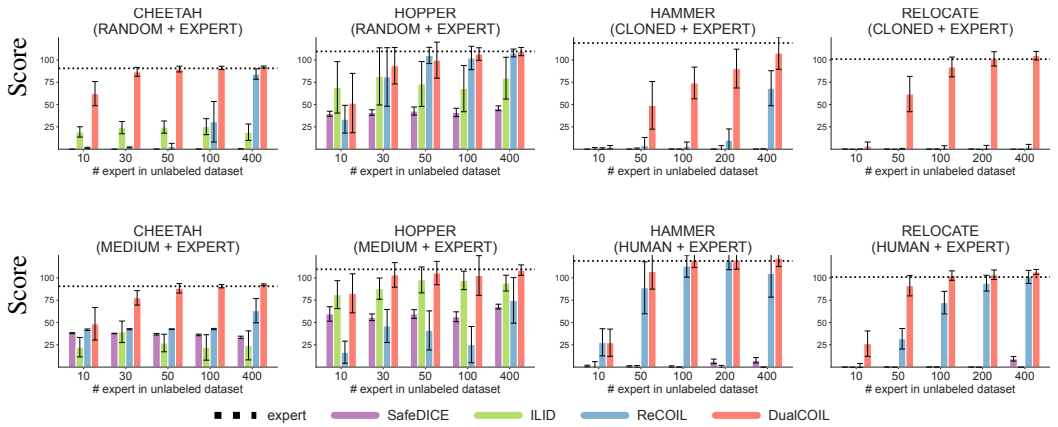

Figure 7: **Effect of Unlabeled Dataset Quality on Performance:** We evaluate the effect of increasing the number of expert trajectories in the unlabeled dataset $\mathcal{B}^{\text{MIX}}$. The results are calculated from 5 different training seeds, reported in normalized score. Our method outperforms SafeDICE, ILID and ReCOIL across both locomotion and manipulation tasks, achieving near-expert performance on most environments even with a small number of expert demonstrations.

D.6 Effect of Increasing the Proportion of Bad Data in the Unlabeled Dataset

In this experiment, we maintain the same good dataset ($\mathcal{B}^G$) and bad dataset ($\mathcal{B}^B$) as used in the main comparison in Section 6.2. Our modification focuses on the unlabeled dataset ($\mathcal{B}^{MIX}$). Within $\mathcal{B}^{MIX}$, the number of EXPERT trajectories remains consistent with Section 6.2, but the RANDOM dataset within it is duplicated multiple times to increase the proportion of bad data (each dataset contain about 1000 RANDOM trajectories). The results, presented in Table 6, indicate that increasing the amount of poor-quality data leads to a general decline in performance across all methods. Nevertheless, our algorithm remains consistently robust and continues to outperform the main baselines under these more challenging conditions.

CHEETAH (RANDOM + EXPERT)	1xRANDOM	2xRANDOM	3xRANDOM	5xRANDOM
SafeDICE	-0.0$_{\pm 0.0}$	-0.0$_{\pm 0.0}$	-0.0$_{\pm 0.0}$	-0.0$_{\pm 0.1}$
ILID	21.1$_{\pm 7.6}$	29.0$_{\pm 1.4}$	24.7$_{\pm 4.0}$	26.7$_{\pm 0.4}$
ReCOIL	2.0$_{\pm 0.6}$	2.3$_{\pm 0.1}$	2.0$_{\pm 0.2}$	1.8$_{\pm 0.7}$
DualCOIL	**86.7**$_{\pm 5.0}$	**81.8**$_{\pm 2.7}$	**75.9**$_{\pm 2.1}$	**59.9**$_{\pm 2.5}$
RELOCATE (CLONED + EXPERT)	**1xRANDOM**	**2xRANDOM**	**3xRANDOM**	**5xRANDOM**
SafeDICE	-0.1$_{\pm 0.0}$	-0.1$_{\pm 0.0}$	-0.1$_{\pm 0.0}$	-0.1$_{\pm 0.0}$
ILID	-0.1$_{\pm 0.1}$	-0.2$_{\pm 0.1}$	-0.2$_{\pm 0.0}$	-0.2$_{\pm 0.0}$
ReCOIL	1.4$_{\pm 2.4}$	0.4$_{\pm 0.3}$	0.1$_{\pm 0.0}$	0.1$_{\pm 0.1}$
DualCOIL	**92.1**$_{\pm 11.1}$	**64.7**$_{\pm 2.4}$	**35.8**$_{\pm 14.3}$	**9.3**$_{\pm 9.2}$
KITCHEN (PARTIAL + COMPLETE)	**1xRANDOM**	**2xRANDOM**	**3xRANDOM**	**5xRANDOM**
SafeDICE	2.8$_{\pm 1.1}$	3.8$_{\pm 2.3}$	4.9$_{\pm 1.5}$	3.0$_{\pm 1.2}$
ILID	27.3$_{\pm 5.4}$	7.6$_{\pm 9.7}$	13.0$_{\pm 4.9}$	11.3$_{\pm 4.4}$
ReCOIL	48.8$_{\pm 8.3}$	41.6$_{\pm 1.8}$	44.5$_{\pm 3.7}$	44.3$_{\pm 8.2}$
DualCOIL	**53.1**$_{\pm 13.1}$	**57.6**$_{\pm 5.4}$	**56.5**$_{\pm 9.2}$	**56.8**$_{\pm 7.0}$

Table 6: Increase the proportion of bad in the unlabeled dataset $\mathcal{B}^{MIX}$ in three different environments.

D.7 Experiments with Extremely Large Bad Dataset

Although we previously examined the effect of the size of the bad dataset $\mathcal{B}^B$ in Appendix D.1, that study was restricted to at most 100 trajectories. In this experiment, we aim to further investigate how enlarging $\mathcal{B}^B$ can improve performance. Since the RANDOM dataset from D4RL is relatively small (which limit the analysis of Appendix D.1), we augment it by generating additional random trajectories through direct interaction with the environment. The experiment results are shown in Table 7.

From the results, we observe that increasing the quantity of bad demonstrations generally leads to improved performance in most cases. This is likely because a richer set of bad data provides better coverage of the undesirable regions in the action space, which helps the algorithm more effectively learn what to avoid. However, in a few cases, the performance either improves only marginally or even slightly decreases. This can be attributed to the fact that adding more bad demonstrations does not always guarantee broader or more informative coverage of poor actions. If the additional bad data is redundant or fails to introduce new undesirable behavior patterns, its benefit to learning may be limited or even slightly detrimental due to noise.

CHEETAH (RANDOM + EXPERT)	100	300	500	1000
DWBC-GB	$2.3_{\pm 2.9}$	$1.4_{\pm 1.3}$	$3.0_{\pm 2.3}$	$3.2_{\pm 2.1}$
SafeDICE	$-0.1_{\pm 0.1}$	$0.0_{\pm 0.0}$	$0.3_{\pm 0.1}$	$0.5_{\pm 0.3}$
DualCOIL	**85.3**$_{\pm 5.1}$	**91.4**$_{\pm 1.5}$	**91.8**$_{\pm 1.0}$	**91.5**$_{\pm 0.8}$
RELOCATE (CLONED + EXPERT)	**100**	**300**	**500**	**1000**
DWBC-GB	$-0.1_{\pm 0.1}$	$-0.2_{\pm 0.0}$	$-0.2_{\pm 0.0}$	$-0.2_{\pm 0.0}$
SafeDICE	$-0.1_{\pm 0.0}$	$-0.1_{\pm 0.0}$	$-0.1_{\pm 0.0}$	$-0.1_{\pm 0.0}$
DualCOIL	**93.2**$_{\pm 10.7}$	**96.1**$_{\pm 12.0}$	**96.2**$_{\pm 11.2}$	**97.9**$_{\pm 11.6}$
KITCHEN (PARTIAL + COMPLETE)	**100**	**300**	**500**	**1000**
DWBC-GB	$17.0_{\pm 9.8}$	$14.8_{\pm 6.7}$	$10.5_{\pm 9.2}$	$15.3_{\pm 8.9}$
SafeDICE	$2.7_{\pm 2.6}$	$1.7_{\pm 0.7}$	$1.9_{\pm 1.4}$	$0.4_{\pm 0.2}$
DualCOIL	**60.3**$_{\pm 10.6}$	**63.8**$_{\pm 9.2}$	**59.2**$_{\pm 8.7}$	**60.8**$_{\pm 9.4}$

Table 7: Increasing size of Bad dataset $\mathcal{B}^B$.

D.8 Comparison with Adapted Offline RL Methods

In this section, we compare our approach with offline RL methods adapted to learn from both good and bad datasets by assigning rewards of $+1$ to $\mathcal{B}^G$ and -1 to $\mathcal{B}^B$, and combining all three datasets into a single offline training set. We evaluate against two widely used baselines, CQL (Kumar et al., 2020) and IQL (Kostrikov et al., 2021), using the same dataset sizes as in Section 6.2 for fairness. The results in Table 8 show that our method consistently outperforms both baselines.

	CHEETAH	HOPPER	HAMMER	RELOCATE	KITCHEN
CQL	$-2.3_{\pm 1.1}$	$26.8_{\pm 13.6}$	$0.3_{\pm 0.0}$	$-0.3_{\pm 0.0}$	$0.0_{\pm 0.0}$
IQL	$-0.5_{\pm 0.6}$	$4.6_{\pm 2.8}$	$4.4_{\pm 3.5}$	$-0.1_{\pm 0.0}$	$11.5_{\pm 6.5}$
DualCOIL	**86.7**$_{\pm 5.0}$	**93.6**$_{\pm 20.5}$	**74.3**$_{\pm 17.8}$	**92.1**$_{\pm 11.1}$	**53.1**$_{\pm 13.1}$

Table 8: Comparison of DualCOIL with offline RL methods.

D.9 Discussion: Distribution-matching Approach vs Preference-based Approach

The good and bad data setup is reminiscent of preference-based methods. In this section, we want to discuss the difference between our approach (distribution-matching) and preference-based approach with two keys aspects:

- **Input data construction:** Our approach is based on contrastive demonstrations, explicitly labeled as good or bad. In contrast, preference-based methods rely on pairwise preference feedback between trajectories, where both trajectories can be good, bad, or of similar quality.

- **Learning objective:** ConstraDICE is designed to **explicitly imitate expert behavior while avoiding bad behavior**. Preference-based methods, on the other hand, aim to infer a reward function or policy that aligns with the provided preferences, without necessarily distinguishing between good and bad demonstrations in an absolute sense.

Intuitively, this means preference-based learning is conceptually different and not well-suited to our setting. Simply enforcing a preference like $r(good) > r(bad)$ does not capture the critical requirement of explicitly avoiding bad behaviors. Even if the method assigns lower rewards to bad trajectories, it does not guarantee that the resulting policy will avoid them.

To empirically support this argument, we conducted additional experiments using an offline preference-based learning approach which is IPL (Hejna & Sadigh, 2024) with the configuration $r(good) > r(bad)$. The results, presented in Table 9, further demonstrate that preference-based methods fail to learn effective policies in our contrastive good-bad setting.

	CHEETAH	HOPPER	HAMMER	RELOCATE	KITCHEN
IPL	$1.5_{\pm 0.1}$	$6.4_{\pm 0.4}$	$0.5_{\pm 0.1}$	$-0.1_{\pm 0.0}$	$34.7_{\pm 3.7}$
DualCOIL	**86.7**$_{\pm 5.0}$	**93.6**$_{\pm 20.5}$	**74.3**$_{\pm 17.8}$	**92.1**$_{\pm 11.1}$	**53.1**$_{\pm 13.1}$

Table 9: Comparison of DualCOIL with IPL.

D.10 Comparison with UNIQ: A State-of-the-Art Algorithm for Learning from Bad Demonstrations

In this section, we present an additional experiment comparing our approach with UNIQ (Hoang et al., 2024b), a state-of-the-art method specifically designed to avoid bad demonstrations (similar to SafeDICE). While both UNIQ and DualCOIL address offline imitation learning under the presence of bad-quality demonstrations, the two methods are fundamentally different in both formulation and learning principle. Specifically, UNIQ builds upon the IQ-Learn framework (Garg et al., 2021), which optimizes a max–min objective over reward and policy using an entropy-regularized formulation. In contrast, DualCOIL is derived from the DuaRL and DICE frameworks (Sikchi et al., 2024), employing a tractable reformulation based on minimizing the KL divergence between state–action visitation distributions. This yields a Q-learning–style objective that is computationally simpler and more stable to optimize. Moreover, DualCOIL introduces a surrogate approximation (Section 4.2) that further enhances efficiency without sacrificing alignment with the original theoretical objective.

For consistency, we adopt the same dataset setup as in Section 6.2, where learning is performed using $\mathcal{B}^B$ only. The results in Figure 8 show that, with expert support, DualCOIL achieves the best overall performance.

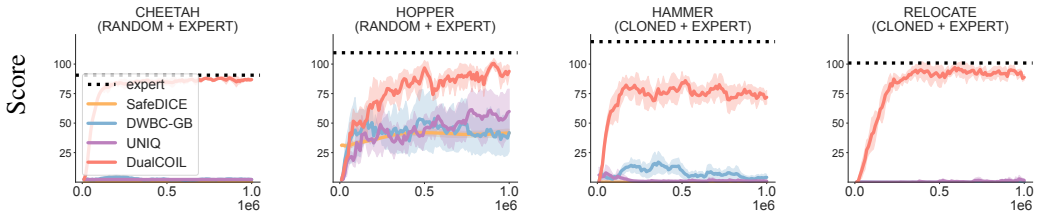

Figure 8: Comparison with UNIQ.

D.11 Adaptations and Experiments with $\alpha > 1$

From our objective function equation 1, we introduce a hyperparameter $0 \leq \alpha < 1$, which controls the weighting of the bad data objective—this corresponds to question **(Q3)**. To evaluate the sensitivity of our method to α, we conduct experiments by varying its value and observing its impact on final

performance. Specifically, we perform a full sweep over $\alpha \in \{0, 0.1, 0.2, \ldots, 0.9\}$ to illustrate how this key hyperparameter influences learning outcomes.

Interestingly, we observe that in some cases, settings with $\alpha \geq 1$ yield favorable performance, suggesting that avoiding bad data may, at times, be more critical than imitating good data. However, directly applying $\alpha \geq 1$ in our original formulation violates convexity conditions.

To address this, we propose a naive modification of Objective equation 7 that accommodates $\alpha \geq 1$ while preserving practical applicability. The revised objective is defined as:

$$\widetilde{L}(Q \mid V) = (1-\gamma)\,\mathbb{E}_{s\sim p_0}\left[V(s)\right] - \mathbb{E}_{(s,a)\sim d^U}\left[\exp\left(\Psi(s,a)\right)\left(Q(s,a) - \gamma \mathbb{E}_{s'}[V(s')]\right)\right], \quad (21)$$

which enables empirical investigation into the high-α regime while sidestepping theoretical limitations. The experiment results are provided in Figure 9. Overall, $\alpha \geq 1$ does not provide good performance, which raises the limitation of the naive adaptation.

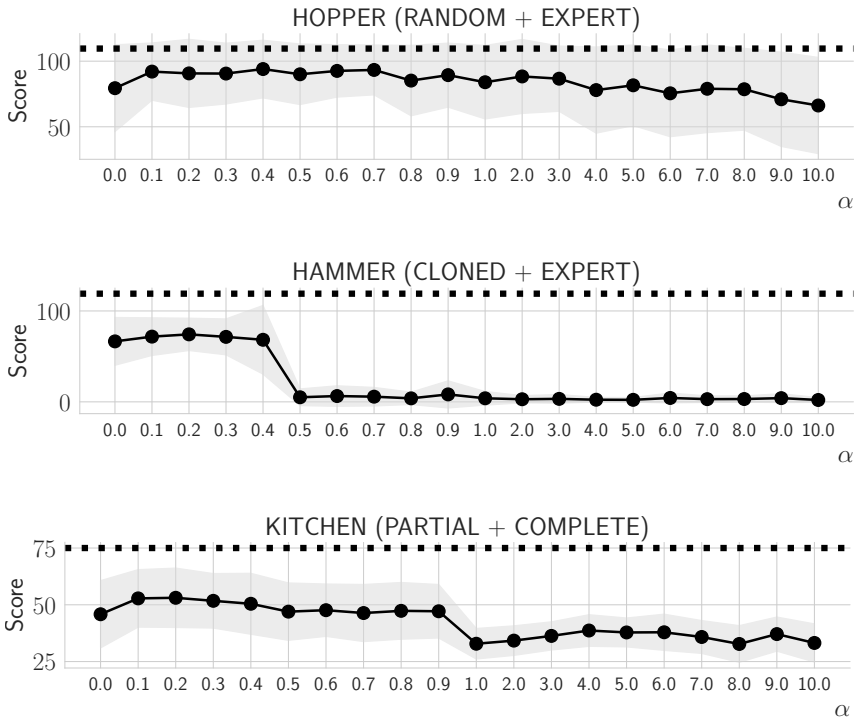

Figure 9: Performance of large $\alpha \geq 1$.

D.12 Comparison Between $L(Q, \pi)$ and the Surrogate $\widetilde{L}(Q, \pi)$

As shown in Proposition 4.3, the original objective $L(Q \mid V)$ (equation 4) is transformed into a modified version $\widetilde{L}(Q \mid V)$ (equation 7). This experiment investigates the performance differences between the two objectives.z

To improve the stability of the original objective $L(Q \mid V)$, we need to address the issue of exponential terms producing extremely large values, which can lead to numerical instability. A practical approach is to clip the input to the exponential function to a bounded range $[\text{minR}, \text{maxR}]$, resulting in the following formulation:

$$
\begin{aligned}
L(Q, \pi) =& (1-\gamma)\, \mathbb{E}_{s \sim p_0}\left[V_Q^\pi(s)\right] \\
&+ (1-\alpha) \mathbb{E}_{(s,a)\sim d^U}\left[\exp\left(\left(\frac{\Psi(s,a) - \mathcal{T}^\pi[Q](s,a)}{1-\alpha}\right).\text{clip}(\text{minR}, \text{maxR})\right)\right], \quad (22)
\end{aligned}
$$

where $\text{minR} = -7$ and $\text{maxR} = 7$ in our experiments.

The results of this ablation study are presented in Figure 10, illustrating the performance impact of this stability-enhancing modification. In general, the clipping technique effectively mitigates the instability caused by the exponential term, successfully preventing NaN errors during training. However, this modification also leads to a drop in performance and, in some tasks, causes the method to fail to learn effectively.

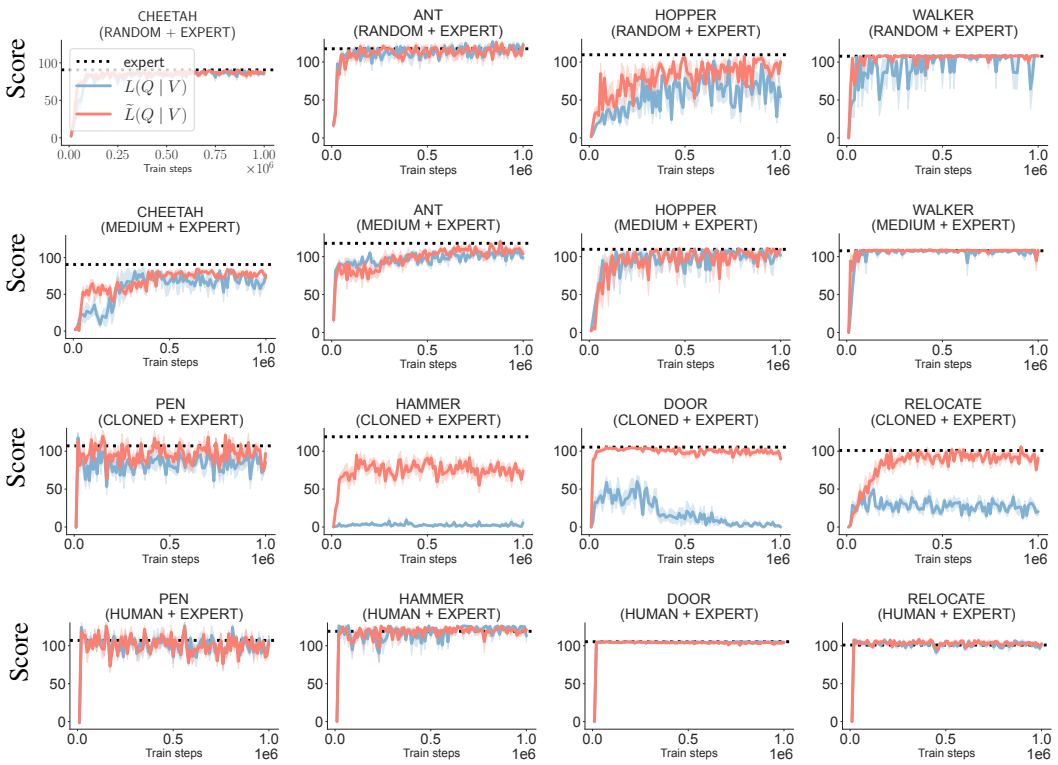

Figure 10: Exponetial ablation study.

D.13 Sensitivity Analysis of β

In this section, we explore how different values of the β parameter affect performance. The experiment results are provided in Table 10. The results show that while β significantly influences outcomes, performance remains consistent over a wide range of β values, implying that minimal tuning effort is needed for this hyperparameter.

Task	unlabeled $\mathcal{B}^{\text{MIX}}$	β value						
		1	3	5	10	15	20	30
CHEETAH	RANDOM+EXPERT	$2.25_{\pm 0.0}$	$2.25_{\pm 0.0}$	$2.25_{\pm 0.0}$	$2.24_{\pm 0.0}$	$83.2_{\pm 5.3}$	**$85.8_{\pm 2.1}$**	$84.3_{\pm 1.4}$
	MEDIUM+EXPERT	$42.4_{\pm 0.2}$	$42.9_{\pm 0.3}$	$53.9_{\pm 8.8}$	**$83.1_{\pm 4.9}$**	$80.1_{\pm 2.6}$	$78.7_{\pm 2.3}$	$76.7_{\pm 5.2}$
ANT	RANDOM+EXPERT	$39.5_{\pm 7.3}$	$69.3_{\pm 6.5}$	$60.9_{\pm 28.7}$	$115.6_{\pm 4.6}$	**$118.0_{\pm 2.1}$**	$114.5_{\pm 1.7}$	$116.0_{\pm 2.1}$
	MEDIUM+EXPERT	$91.0_{\pm 1.1}$	$90.6_{\pm 1.7}$	$93.7_{\pm 1.5}$	$104.8_{\pm 3.9}$	**$106.5_{\pm 2.4}$**	$101.1_{\pm 3.3}$	$95.1_{\pm 1.3}$
HOPPER	RANDOM+EXPERT	$4.7_{\pm 0.4}$	$5.2_{\pm 0.9}$	$7.2_{\pm 1.3}$	$7.9_{\pm 1.9}$	$20.4_{\pm 9.7}$	$67.4_{\pm 7.9}$	**$94.4_{\pm 6.3}$**
	MEDIUM+EXPERT	$52.1_{\pm 1.5}$	$46.0_{\pm 1.0}$	$85.8_{\pm 11.6}$	$96.3_{\pm 8.1}$	$96.9_{\pm 12.5}$	**$99.6_{\pm 4.1}$**	$98.0_{\pm 5.7}$
WALKER	RANDOM+EXPERT	$2.9_{\pm 2.6}$	$3.5_{\pm 2.9}$	$6.4_{\pm 4.6}$	$32.5_{\pm 27.7}$	$105.7_{\pm 4.5}$	$106.2_{\pm 2.0}$	**$107.5_{\pm 1.1}$**
	MEDIUM+EXPERT	$68.3_{\pm 3.7}$	$65.8_{\pm 3.2}$	$53.4_{\pm 3.6}$	$104.9_{\pm 2.5}$	$108.1_{\pm 0.1}$	**$108.2_{\pm 0.2}$**	**$108.2_{\pm 0.1}$**

Table 10: Performance of DualCOIL in different β value in MuJoCo locomotion tasks.

D.14 Comparison with Offline RL Methods

Keen readers may question the use of the reward function $\Psi(s, a)$, since it is computed using only two discriminators and can, in principle, be incorporated into other discriminator-based offline IL or offline RL methods. To examine this, we compare our approach against OptiDICE (a general offline RL formulation adopted by DemoDICE and SMODICE), as well as several state-of-the-art offline RL algorithms, including IQL, XQL, and f-DVL, each of which solves the offline RL problem using $\Psi(s, a)$ as the reward. The experimental results are presented in Figure 11.

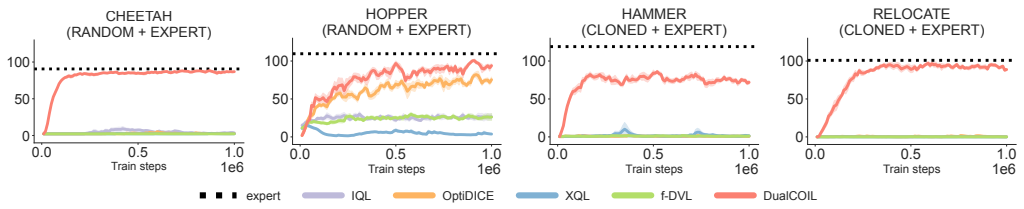

Figure 11: Compare with offline RL methods that utilize the reward function $\Psi(s, a)$.

In principle, a key advantage of our algorithm lies in its ability to fully leverage the implicit reward $\Psi(s, a)$ derived from our formulation in Proposition 4.2. Unlike Q-learning–style offline RL methods such as IQL and XQL, which optimize only the reward component $\Psi(s, a)$, DualCOIL directly optimizes the complete KL-regularized objective specified in Proposition 4.2. This allows the algorithm to exploit the full structure of the learning objective rather than a single term. Furthermore, compared to prior offline RL methods that explicitly incorporate a KL term—such as ReCOIL and OptiDICE — DualCOIL is built upon a SOTA KL-regularized framework inspired by DualRL. Moreover, DualCOIL is built with two additional benefits: (i) a stable and tractable approximation to the regularized learning objective (Prop. 4.4), and (ii) the Q-weighted behavioral cloning update (Proposition 5.1), which substantially improves optimization stability and robustness. Our ablation studies in Appendix D.4 and Appendix D.12 further demonstrate that these components jointly enable DualCOIL to outperform direct offline RL baselines and to utilize the implicit reward signal far more effectively.

D.15 Extreme-V versus Log-sum-exp for Value Function Estimation

In Section 5, we use Extreme-V to estimate the value function V instead of LogSumExp to handle continuous action spaces. However, in prior work related to soft Q-learning, in the continuous action space, researchers estimate V based on the current Q and policy as $V(s) = Q(s, \pi(a|s)) + \alpha H(\pi)$, where $H(\pi)$ is the entropy of the policy π and α is a multiplier controlling the contribution of the entropy to the value function. In this section, we compare the performance of Extreme-V with this entropy-based estimation. We follow the same α as LS-IQ (Al-Hafez et al., 2023) for MuJoCo tasks, while setting a fixed $\alpha = 0.01$ for all Adroit tasks. The comparison results are provided in Figure 12.

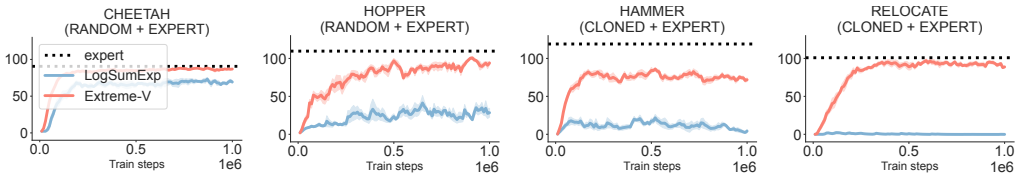

Figure 12: Compare Extreme-V with LogSumExp.

D.16 DUALCOIL FOR SAFETY TASKS

In this section, we evaluate our method in a safe domain where the objective is to maximize return while ensuring the cost remains below a specified threshold. We utilize the Safety Gymnasium benchmark for this experiment, specifically the SafetyPointGoal1 and SafetyPointButton1 environments, with a cost threshold of 25.0. Trajectories with an accumulated cost exceeding 25.0 are considered unsafe, while those with an accumulated reward below a specific return threshold are categorized as low-return.

Policy type	SafetyPointGoal1		SafetyPointButton1	
	Return	Cost	Return	Cost
Safe High-Return	$24.4_{\pm 2.6}$	$11.5_{\pm 19.0}$	$10.2_{\pm 5.0}$	$16.4_{\pm 33.3}$
Unsafe High-Return	$24.2_{\pm 1.6}$	$59.8_{\pm 41.3}$	$10.2_{\pm 3.2}$	$135.3_{\pm 37.2}$
Low-Return	$-4.0_{\pm 1.6}$	$21.8_{\pm 43.9}$	$-0.3_{\pm 0.9}$	$42.7_{\pm 79.2}$

Table 11: Quality of datasets in two tasks SafetyPointGoal1 and SafetyPointButton1 in mean and standard deviation.

We generate three types of policies: safe high-return (using PPO-Lag), unsafe high-return (using PPO), and low-return (using a random policy). The detailed quality of these datasets is reported in Table 11. Using these policies, we construct the following datasets:

- Good dataset (D^G): Consists of safe high-return trajectories.
- Bad dataset (D^B): Consists of unsafe high-return and low-return trajectories.
- Unlabeled dataset (D^{MIX}): A mixture of unsafe high-return, safe high-return, and low-return trajectories.

In our experiments, we fix the size of D^G to 5 safe high-return trajectories. D^{MIX} is composed of 500 unsafe high-return, 500 low-return, and 100 safe high-return trajectories. We aim to verify DualCOIL's ability to leverage varying sizes of D^B (comprising 50% unsafe high-return and 50% low-return data) to assist training, thereby helping the policy avoid constraint violations and low-return behaviors.

For evaluation, we test the policy over 100 independent episodes. We compute the violation rate, defined as the proportion of episodes where the return falls below the specified return threshold and the cost exceeds the cost threshold. Additionally, we report the average Return and Cost for these runs. Detailed results are provided in Figure 13. Overall, there are a clearly trend that increasing the size of bad dataset D^B lead to lower violation rate.

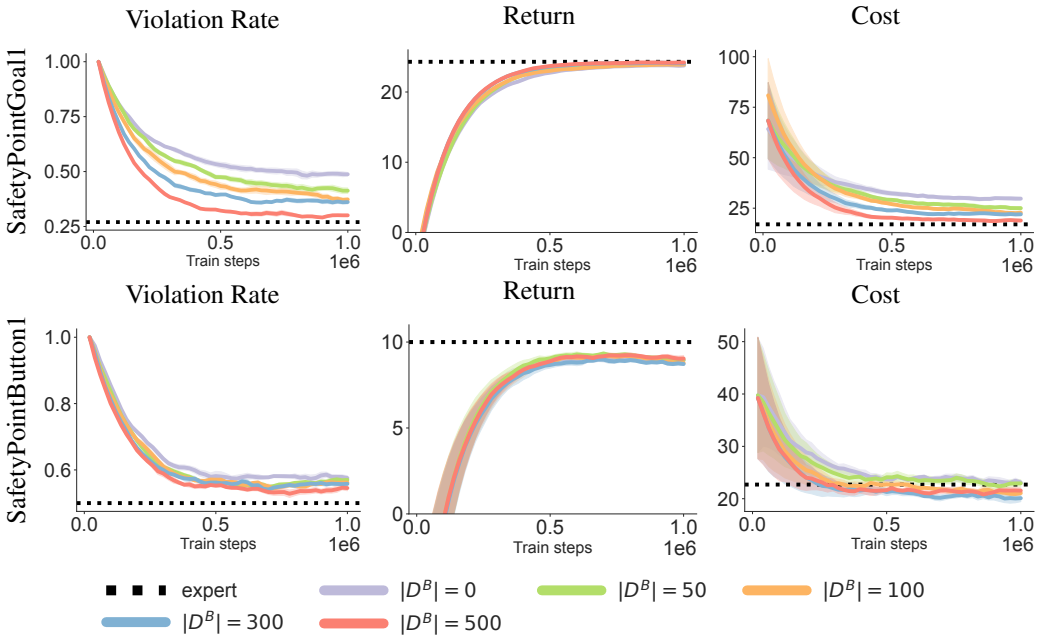

Figure 13: Performance of DualCOIL across different sizes of the bad dataset D^B on Safety Gymnasium tasks. The violation rate reflects the proportion of unsafe or low-return behaviors exhibited by the current policy during evaluation (lower is better). Higher return indicates better task performance, while lower cost corresponds to safer behavior.