Learning What to Do and What Not To Do: Offline Imitation from Expert and Undesirable Demonstrations

Huy Hoang
Singapore Management University
mh.hoang.2024@phdcs.smu.edu.sg

Tien Mai
Singapore Management University
atmai@smu.edu.sg

Pradeep Varakantham
Singapore Management University
pradeepv@smu.edu.sg

Tanvi Verma
Institute of High Performance Computing
Agency for Science, Technology and Research, Singapore
Tanvi_Verma@ihpc.a-star.edu.sg

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 [8, 21, 25, 15, 40] offers a compelling alternative to Reinforcement Learning (RL) [37, 32, 29] 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.

Traditional imitation learning typically assumes access to high-quality expert demonstrations, which can be expensive and difficult to obtain [34, 38, 44]. 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 [4, 30, 15].

While existing imitation learning approaches in the mixed-quality setting typically assume that mixed-quality demonstrations are not drastically different from expert behavior, they often frame learning as mimicking both expert and sub-optimal trajectories—albeit with different weights [21, 20, 40]. However, in practice, mixed-quality 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 demon-

Preprint. Under review.

strations may correspond to incorrect diagnoses or unsafe treatment plans that could harm patients if imitated. Existing imitation learning approaches are limited in their ability to handle contrasting demonstrations. Most methods are either not explicitly designed to avoid undesirable behaviors, or are ill-equipped to deal with scenarios where both expert and undesirable demonstrations coexist within the dataset [39, 42, 15]. 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 [21, 20]. 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*, where interaction with the environment is not available, and assume that the dataset contains both *expert* and *undesirable* demonstrations. We make the following 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 [34]. Recent work addresses this issue by leveraging the strong generalization capabilities of generative models [43, 5]. Inspired by GANs [11], methods like GAIL [14] and AIRL [7] use a discriminator to align the learner's policy with the expert's, while SQIL [33] 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 [31] and ValueDICE [22] employ Stationary Distribution Correction Estimation (DICE), though they often encounter stability issues. Building on ValueDICE, O-NAIL [3] avoids adversarial training, enabling stable offline imitation. More recently, several approaches have extended the DICE framework with stronger theoretical foundations and improved empirical performance [24, 28]. In parallel, IQ-Learn [8] has emerged as a unified framework for both online and offline imitation learning, inspiring a range of follow-up works [2, 16]. 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 [19, 18, 13]. Recent works, such as SPRINQL [15], take advantage of demonstrations

that exhibit varying levels of suboptimality, enabling the learner to better generalize beyond near-optimal behaviors. Another important line of research explores the use of unlabeled demonstrations in conjunction with a limited number of expert trajectories. Techniques like DemoDICE [21], SMODICE [27], and ReCOIL [35] apply Distribution Correction Estimation (DICE) [36, 24, 28] 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 [40], ISW-BC [25], and ILID [41], 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.

SafeDICE [17] was introduced to address this problem of avoiding undesirable or bad demonstrations. However, SafeDICE is not designed to handle scenarios where both expert and undesirable datasets are available. Moreover, their approach still relies on minimizing a divergence between the learning policy and a mixture of unlabeled and undesirable data—an approach that is vulnerable to the quality of the unlabeled dataset and may degrade when such data is of low quality.

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 reinforcement learning [1, 15, 10] and large language model training [26]. 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 [24, 21].

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 $d^{E,\text{MIX}}$ [21, 20, 27, 35]. 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 intro-

duced—as in our approach—this convexity may no longer hold, rendering many existing methods inapplicable.

4 ContraDICE: 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. 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 [31, 22], we propose to optimize the following training 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 (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 (1). To this end, we formulate the problem as the following constrained optimization:

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

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 (1):

Proposition 4.2. *The objective function in* (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).

Given the convexity of the objective in (1), we can equivalently move the constraints into the objective using Lagrangian duality, leading to the following Q-learning formulation (details of the derivation are given in the appendix):

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

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 [8, 12]. 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:

$$L(Q,\pi) = (1-\gamma)\, \mathbb{E}_{s\sim p_0} \left[V_Q^\pi(s) \right] + (1-\alpha) \mathbb{E}_{(s,a)\sim d^U} \left[\exp\left(\frac{\Psi(s,a) - \mathcal{T}^\pi[Q](s,a)}{1-\alpha} \right) \right]. \quad (3)$$

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.

Remark. The objective in Equation (3) is valid only when $\alpha < 1$. In the special case where $\alpha = 1$, i.e., when the bad demonstrations are weighted equally to the expert demonstrations—the training objective simplifies significantly. According to Proposition 4.2, the training objective reduces to a standard offline RL problem with reward function $\Psi(s,a)$: $\max_d\ \mathbb{E}_{(s,a)\sim d} [\Psi(s,a)] = \max \mathbb{E} \left[\sum_{t=0}^{\infty} \gamma^t \Psi(s_t,a_t) \right]$.

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 (3) 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:*

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

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. We note that the training objective in Equation (4) generalizes the IQ-Learn objective [8] 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:

$$\max_{c^G} \left\{ \mathbb{E}_{(s,a)\sim d^G}\left[\log c^G(s,a)\right] + \mathbb{E}_{(s,a)\sim d^U}\left[\log(1-c^G(s,a))\right]\right\}. \tag{5}$$

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) [9], 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:

$$\widetilde{L}(Q \mid V) = (1-\gamma)\,\mathbb{E}_{s\sim p_0}\left[V(s)\right] - \mathbb{E}_{(s,a)\sim d^U}\left[\exp\left(\frac{\Psi(s,a)}{1-\alpha}\right)\left(Q(s,a) - \gamma\mathbb{E}_{s'}[V(s')]\right)\right]. \tag{6}$$

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 [8, 9] 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) [23, 9, 13, 35]:

$$\max_{\pi} \sum_{(s,a)\sim\mathcal{B}^U} \exp\left(\frac{1}{\beta}\left(Q(s,a) - V(s)\right)\right) \log \pi(a \mid s). \tag{7}$$

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* (7)*:*

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

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 ContraDICE algorithm in Algorithm 1.

Algorithm 1 ContraDICE

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 5.
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 ContraDICE 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 ContraDICE? **(Q3)** ContraDICE relies on an important parameter α to balance the objectives for good and bad data—how does this parameter affect overall performance? Moreover, we also provide some additional experiments in the Appendix.

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 $\mathcal{B}^{\text{MIX}}$. The use of such unlabeled data is common in offline imitation learning from mixed-quality demonstrations. 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 [6]. 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 [35]. 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 Behavioral Cloning 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 [27], ILID [41], and ReCOIL [35]. We exclude DWBC [40] 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(4), 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 ContraDICE-G. For methods that incorporate support from bad data $\mathcal{B}^B$, we evaluate our approach against SafeDICE [17]. 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.

Reproducibility. We provide detailed hyperparameters and network architectures for each task in the Appendix. To ensure reproducibility and comparison, the source code is publicly available at: https://github.com/hmhuy0/ContraDICE .

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	ContraDICE-G	SafeDICE	DWBC-GB	ContraDICE	Expert
CHEETAH	RANDOM+EXPERT	$2.3_{\pm 0.0}$	$-0.6_{\pm 0.7}$	$4.6_{\pm 2.7}$	$21.1_{\pm 7.6}$	$2.0_{\pm 0.6}$	$84.4_{\pm 5.3}$	$-0.0_{\pm 0.0}$	$2.8_{\pm 1.1}$	**$86.7_{\pm 5.0}$**	90.6
	MEDIUM+EXPERT	$42.5_{\pm 0.5}$	$-0.6_{\pm 0.7}$	$42.4_{\pm 3.5}$	$40.3_{\pm 15.6}$	$42.5_{\pm 0.6}$	$48.6_{\pm 4.4}$	$37.7_{\pm 0.3}$	$5.6_{\pm 4.3}$	**$77.6_{\pm 8.1}$**	90.6
ANT	RANDOM+EXPERT	$30.9_{\pm 0.1}$	$-7.2_{\pm 10.3}$	$4.6_{\pm 21.6}$	$71.8_{\pm 19.4}$	$56.2_{\pm 11.2}$	$100.6_{\pm 22.1}$	$-2.6_{\pm 0.0}$	$6.5_{\pm 7.5}$	**$112.7_{\pm 12.9}$**	117.5
	MEDIUM+EXPERT	$91.2_{\pm 1.9}$	$-7.2_{\pm 10.3}$	$88.5_{\pm 9.3}$	$39.6_{\pm 25.7}$	$100.8_{\pm 9.0}$	$102.4_{\pm 7.8}$	$88.1_{\pm 0.9}$	$-4.3_{\pm 5.3}$	**$107.4_{\pm 11.0}$**	117.5
HOPPER	RANDOM+EXPERT	$4.9_{\pm 0.2}$	$17.9_{\pm 6.1}$	$56.4_{\pm 20.6}$	$81.6_{\pm 32.0}$	$81.0_{\pm 32.8}$	$79.4_{\pm 33.1}$	$41.1_{\pm 3.1}$	$40.8_{\pm 21.3}$	**$93.6_{\pm 20.5}$**	109.6
	MEDIUM+EXPERT	$52.2_{\pm 1.3}$	$17.9_{\pm 6.1}$	$53.0_{\pm 3.7}$	$87.9_{\pm 11.9}$	$46.1_{\pm 18.5}$	$70.6_{\pm 17.9}$	$55.8_{\pm 3.7}$	$21.6_{\pm 8.9}$	**$103.7_{\pm 16.3}$**	109.6
WALKER	RANDOM+EXPERT	$1.5_{\pm 0.1}$	$3.8_{\pm 3.3}$	$106.6_{\pm 1.5}$	$100.1_{\pm 9.8}$	$29.8_{\pm 33.4}$	$97.5_{\pm 24.0}$	$23.0_{\pm 1.8}$	$17.4_{\pm 16.7}$	**$107.4_{\pm 3.7}$**	107.7
	MEDIUM+EXPERT	$70.8_{\pm 0.7}$	$3.8_{\pm 3.3}$	$6.0_{\pm 5.0}$	$89.7_{\pm 23.7}$	$72.1_{\pm 12.1}$	$99.8_{\pm 15.5}$	$60.2_{\pm 2.9}$	$25.6_{\pm 16.6}$	**$108.2_{\pm 0.9}$**	107.7
PEN	CLONED+EXPERT	$56.0_{\pm 1.1}$	$8.8_{\pm 3.1}$	$10.9_{\pm 14.6}$	$1.9_{\pm 4.7}$	$79.2_{\pm 21.4}$	$66.3_{\pm 21.5}$	$19.9_{\pm 4.6}$	$9.5_{\pm 8.8}$	**$96.4_{\pm 19.4}$**	107.0
	HUMAN+EXPERT	$18.3_{\pm 1.4}$	$8.8_{\pm 3.1}$	$-2.5_{\pm 0.5}$	$5.1_{\pm 4.8}$	$99.9_{\pm 18.9}$	$95.5_{\pm 19.7}$	$21.8_{\pm 5.7}$	$6.5_{\pm 5.3}$	**$101.5_{\pm 18.7}$**	107.0
HAMMER	CLONED+EXPERT	$0.4_{\pm 0.8}$	$1.4_{\pm 0.7}$	$0.8_{\pm 0.9}$	$0.4_{\pm 1.3}$	$3.4_{\pm 4.6}$	$66.5_{\pm 26.3}$	$0.0_{\pm 0.2}$	$2.8_{\pm 5.6}$	**$74.3_{\pm 17.8}$**	119.0
	HUMAN+EXPERT	$12.8_{\pm 7.3}$	$1.4_{\pm 0.7}$	$1.9_{\pm 4.6}$	$1.2_{\pm 3.1}$	$113.2_{\pm 12.4}$	$113.2_{\pm 16.1}$	$0.6_{\pm 0.8}$	$3.4_{\pm 4.2}$	**$120.0_{\pm 8.3}$**	119.0
DOOR	CLONED+EXPERT	$0.4_{\pm 0.7}$	$-0.1_{\pm 0.1}$	$-0.1_{\pm 0.1}$	$-0.1_{\pm 0.2}$	$19.3_{\pm 16.7}$	$92.6_{\pm 11.3}$	$-0.0_{\pm 0.0}$	$-0.1_{\pm 0.1}$	**$102.4_{\pm 3.8}$**	105.3
	HUMAN+EXPERT	$4.0_{\pm 2.6}$	$-0.1_{\pm 0.1}$	$-0.1_{\pm 0.7}$	$0.2_{\pm 1.6}$	$100.3_{\pm 6.4}$	$104.7_{\pm 1.5}$	$0.9_{\pm 0.9}$	$1.1_{\pm 1.1}$	**$105.0_{\pm 1.2}$**	105.3
RELOCATE	CLONED+EXPERT	$-0.1_{\pm 0.1}$	$-0.1_{\pm 0.1}$	$0.1_{\pm 0.2}$	$-0.1_{\pm 0.1}$	$1.4_{\pm 2.4}$	$34.5_{\pm 13.9}$	$-0.1_{\pm 0.0}$	$-0.2_{\pm 0.1}$	**$92.1_{\pm 11.1}$**	100.9
	HUMAN+EXPERT	$0.0_{\pm 0.1}$	$-0.1_{\pm 0.1}$	$-0.2_{\pm 0.1}$	$-0.2_{\pm 0.2}$	$72.3_{\pm 12.6}$	$99.1_{\pm 6.9}$	$0.0_{\pm 0.1}$	$-0.1_{\pm 0.0}$	**$102.6_{\pm 5.3}$**	100.9
KITCHEN	PARTIAL+COMPLETE	$45.5_{\pm 1.9}$	$2.5_{\pm 5.0}$	$5.5_{\pm 8.2}$	$27.3_{\pm 5.4}$	$48.8_{\pm 8.9}$	$45.8_{\pm 14.8}$	$2.8_{\pm 1.1}$	$19.4_{\pm 4.6}$	**$53.1_{\pm 13.1}$**	75.0
	MIXED+COMPLETE	$42.1_{\pm 1.1}$	$2.2_{\pm 3.8}$	$3.1_{\pm 5.8}$	$13.3_{\pm 3.1}$	**$50.6_{\pm 3.8}$**	$20.3_{\pm 14.1}$	$1.5_{\pm 1.9}$	$6.7_{\pm 4.4}$	$48.9_{\pm 16.4}$	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 deviation.

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 (ContraDICE-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

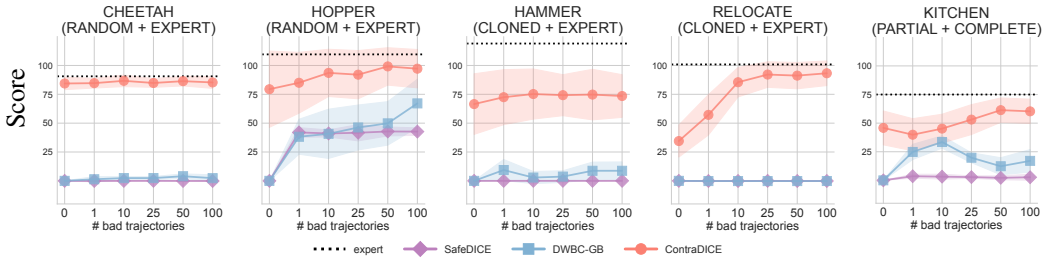

Figure 1: Effect of the size of the bad dataset $\mathcal{B}^B$ on learning performance: The results are averaged over five different training seeds and reported using normalized scores. As the number of bad trajectories increases, our method demonstrates a strong ability to leverage this data. In contrast, baseline methods such as SafeDICE and DWBC-GB struggle to make effective use of bad demonstrations.

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

From our objective function (1), 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 conduct experiments by varying 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.

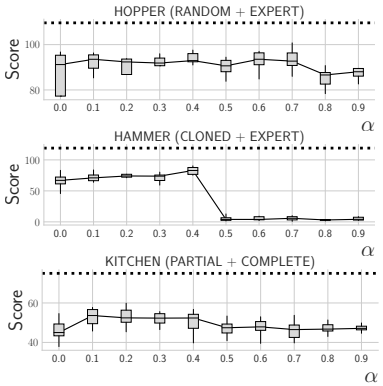

Figure 2: Sensitivity analysis on the trade-off parameter α.

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 shows strong empirical performance, it is currently limited to settings where $\alpha \leq 1$. Relaxing this constraint would make the learning objective more challenging to optimize, but represents a promising direction for future research. Additionally, we assume access to well-labeled expert and undesirable demonstrations, which may not hold in practice. Developing robust methods that can learn effectively from noisy or weakly labeled data would be a valuable extension of this work.

References

[1] 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.

[2] 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.

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

[4] 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*, pages 783–792. PMLR, 2019.

[5] 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*, page 02783649241273668, 2023.

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

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

[8] 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.

[9] Divyansh Garg, Joey Hejna, Matthieu Geist, and Stefano Ermon. Extreme q-learning: Maxent rl without entropy. In *International Conference on Learning Representations (ICLR)*, 2023.

[10] 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, pages 16880–16887, 2025.

[11] 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.

[12] 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.

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

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

[15] 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, pages 12439–12447, 2024.

[16] 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*, 2024.

[17] 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.

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

[19] 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.

[20] 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.

[21] 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.

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

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

[24] 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*, pages 6120–6130. PMLR, 2021.

[25] 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.

[26] 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.

[27] 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*, pages 14639–14663. PMLR, 2022.

[28] 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.

[29] 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.

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

[31] 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.

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

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

[34] 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*, pages 627–635. JMLR Workshop and Conference Proceedings, 2011.

[35] 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.

[36] 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.

[37] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. The MIT Press, second edition, 2018.

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

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

[40] 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*, pages 24725–24742, 2022.

[41] 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.

[42] 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.

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

[44] 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.

Appendix

A Missing Proofs

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.*

Proof. We write the objective function as:

$$\begin{aligned} f(d^\pi) &= \sum_{(s,a)\sim d^\pi} \log \frac{d^\pi(s,a)}{d^G(s,a)} - \alpha \sum_{(s,a)\sim d^\pi} \log \frac{d^\pi(s,a)}{d^B(s,a)} \\ &= \sum_{s,a} (1-\alpha) d^\pi(s,a) \log p^\pi(s,a) + d^\pi(s,a)(\alpha d^B(s,a) - d^G(s,a)) \end{aligned} \tag{9}$$

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* (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:*

$$\widetilde{L}(Q,\pi) = (1-\gamma)\, \mathbb{E}_{s\sim p_0}\left[V_Q^\pi(s)\right] - \mathbb{E}_{d^U}\left[\delta(s,a)\mathcal{T}^\pi[Q](s,a)\right] + (1-\alpha)\mathbb{E}_{d^U}\left[\delta(s,a)\right]. \tag{10}$$

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

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

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* (7)*:*

$$\max_\pi \sum_{(s,a)\sim\mathcal{B}^U} \exp\left(\frac{1}{\beta}Q(s,a)\right)\log\pi(a \mid s). \tag{11}$$

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 (7). This completes the proof.

□

B A Note on ContraDICE 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.

C Experiment Settings

C.1 Full Pseudo Code

The detailed implementation are provided in Algorithm 2.

Algorithm 2 ContraDICE: 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 (5).
    Update $c^B_{w_B}$ by maximizing an analogous objective to Equation (5) 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 (6))
    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 (8))
    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 [35], 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 [35], 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:

$$\min_{\pi} -\mathbb{E}_{(s,a)\sim\mathcal{B}} \log \pi(a \mid s), \tag{12}$$

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** [27]: 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** [41]: Applied to $\mathcal{B}^G$ and $\mathcal{B}^{\text{MIX}}$. The official code is available at [GitHub].
- **ReCOIL** [35]: Applied to $\mathcal{B}^G$ and $\mathcal{B}^{\text{MIX}}$. The official code is available at [GitHub].
- **SafeDICE** [17]: 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 [40] (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:

$$\begin{aligned} L_{c^G} = \eta\, &\mathbb{E}_{(s,a)\sim\mathcal{B}^G}[-\log c^G(s,a,\log\pi(a|s))] \\ &+ \mathbb{E}_{(s,a)\sim\mathcal{B}^{\text{MIX}}}[-\log(1-c^G(s,a,\log\pi(a|s)))] \\ &- \eta\, \mathbb{E}_{(s,a)\sim\mathcal{B}^G}[-\log(1-c^G(s,a,\log\pi(a|s)))], \end{aligned} \tag{13}$$

$$\begin{aligned} L_{c^B} = \eta\, &\mathbb{E}_{(s,a)\sim\mathcal{B}^B}[-\log c^B(s,a,\log\pi(a|s))] \\ &+ \mathbb{E}_{(s,a)\sim\mathcal{B}^{\text{MIX}}}[-\log(1-c^B(s,a,\log\pi(a|s)))] \\ &- \eta\, \mathbb{E}_{(s,a)\sim\mathcal{B}^B}[-\log(1-c^B(s,a,\log\pi(a|s)))]. \end{aligned} \tag{14}$$

The policy π is then learned by minimizing the objective:

$$\begin{aligned} \min_{\pi}\Bigg(&\mathbb{E}_{(s,a)\sim\mathcal{B}^G}\left[-\log\pi(a|s)\cdot\left(\alpha - \frac{\eta}{c(s,a)\,(1-c(s,a))}\right)\right] \\ &+ \mathbb{E}_{(s,a)\sim\mathcal{B}^{\text{MIX}}}\left[-\log\pi(a|s)\cdot\frac{1}{1-c(s,a)}\right]\Bigg), \end{aligned} \tag{15}$$

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.6, and D.8 present ablation studies detailing the sensitivity to these parameters.

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

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	0.3	10.0
	MIXED+COMPLETE	0.3	30.0

Table 3: Hyper parameters.

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.

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 ContraDICE 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 ContraDICE to achieve its highest performance across all tasks.

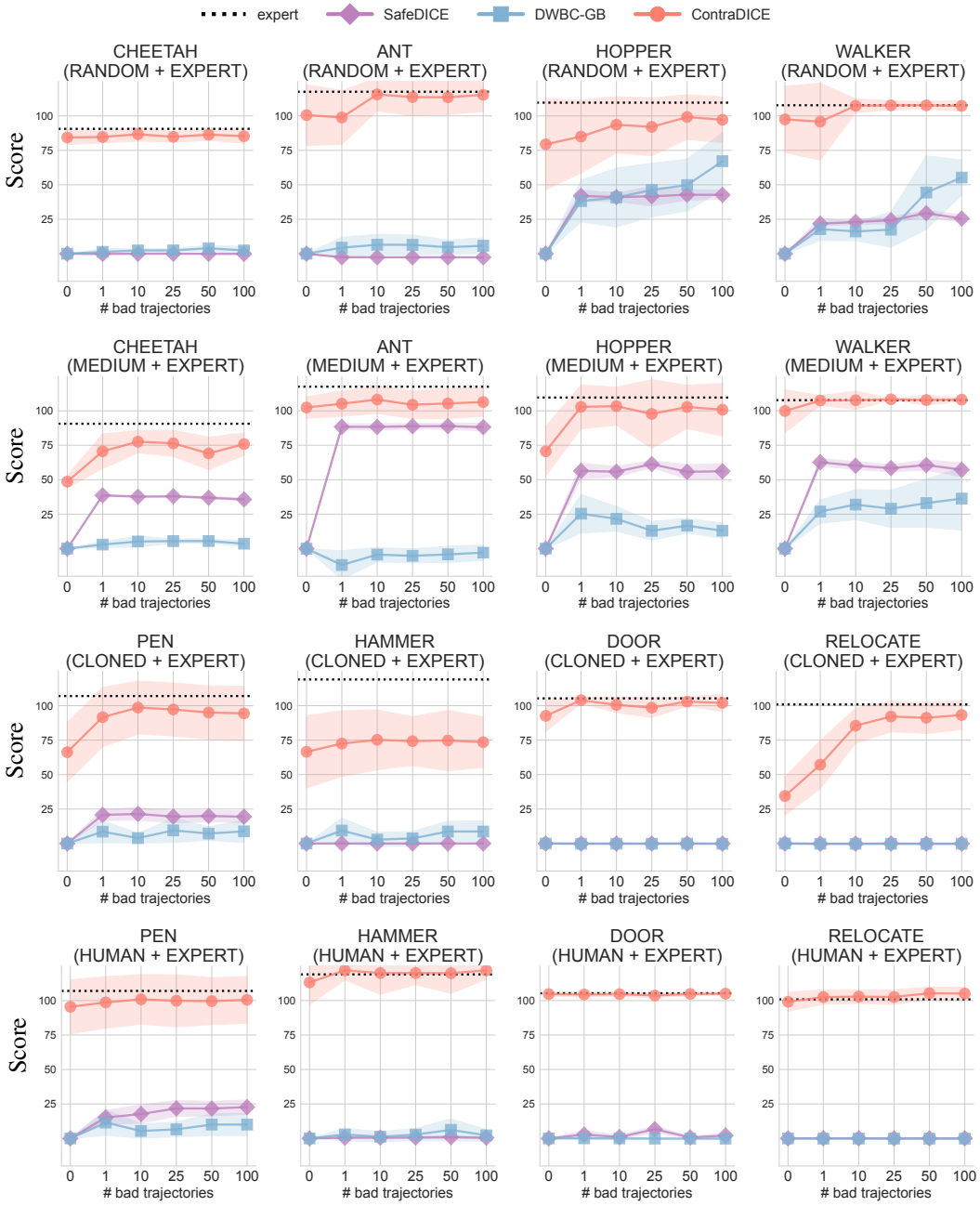

Figure 3: Full bad dataset size effect. SafeDICE and DWBC-GB do not have version that learn from 0 bad trajectory, we assign result 0.0 for them.

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

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, ContraDICE 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 ContraDICE: Through this experiment, we observe that in the Mujoco tasks, ILID can outperform ContraDICE-G when the size of the good dataset is sufficiently large. This highlights a limitation of ContraDICE, 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 ContraDICE by incorporating similar data filtering mechanisms. Specifically, enhancing ContraDICE 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 ContraDICE remains a robust choice in settings where such data is limited and scalalbe with bad dataset.

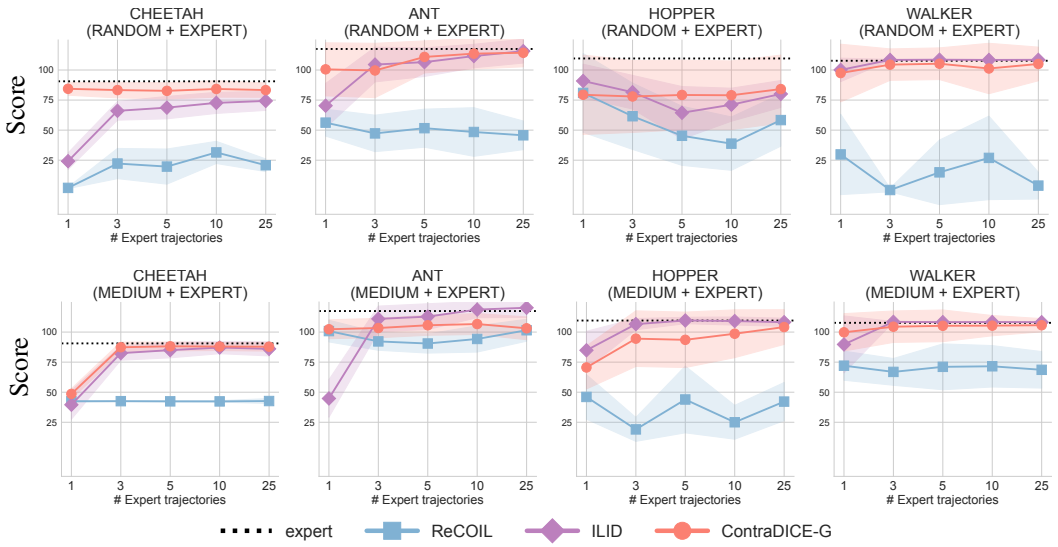

Figure 4: Different of good dataset size without impact from bad dataset in MuJoCo Locomotion tasks.

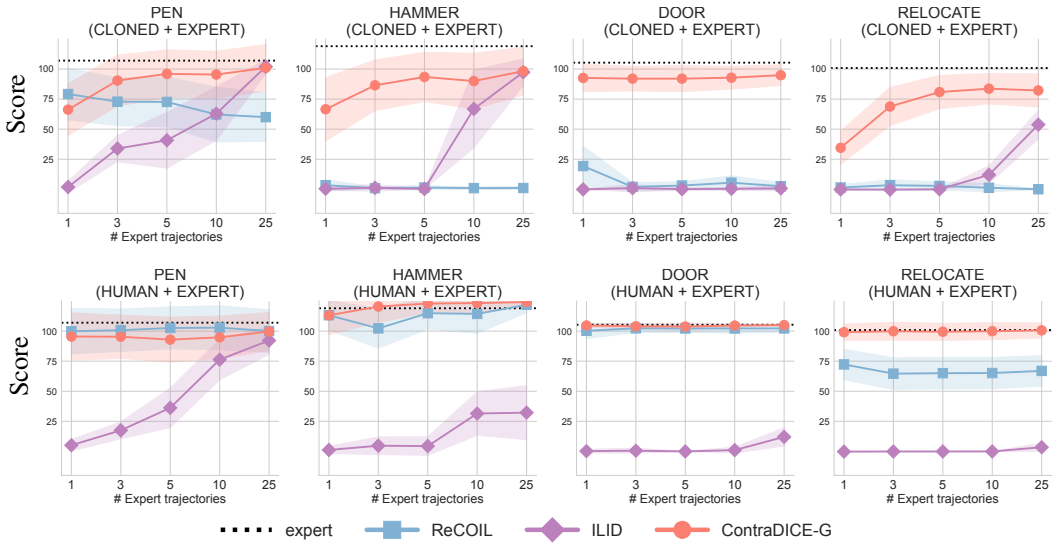

Figure 5: Different of good dataset size without impact from bad dataset in Adroit Manipulation tasks.

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

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

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 ContraDICE 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 ContraDICE 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 (8)), in contrast to prior approaches that rely on AW-BC (Objective (7)). 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.

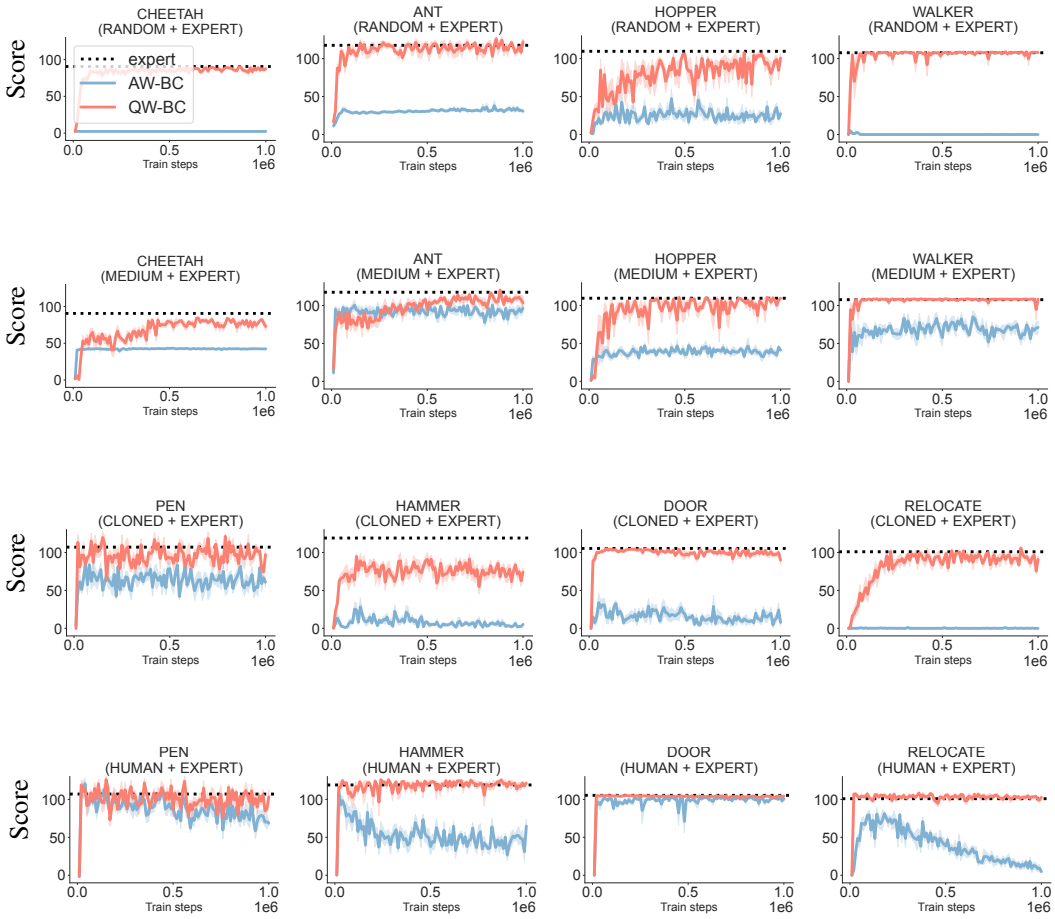

Figure 6: AW-BC and QW-BC comparison.

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.

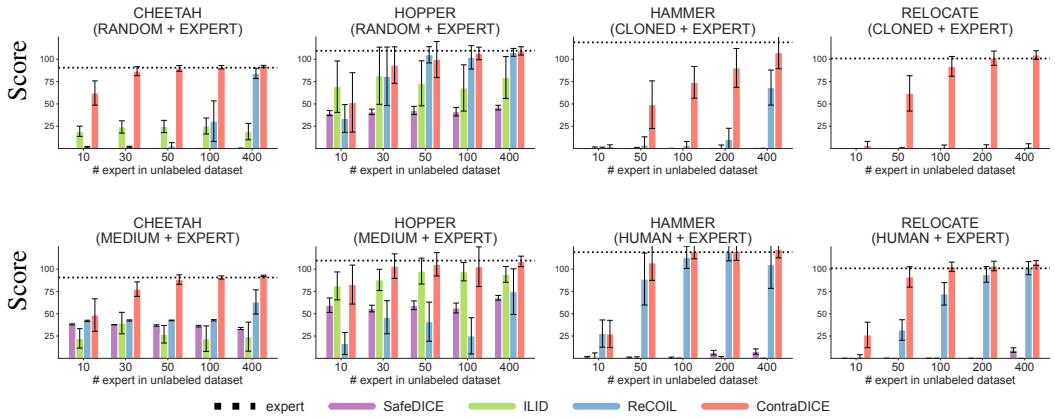

Figure 7: **Effect of Unlabeled Dataset Quality on Performance:** We evaluate the effect of increasing the number of expert trajectories in the unlabeled dataset $\mathcal{B}^{\text{MIX}}$. The results are calculated from 5 different training seeds, reported in normalized score. Our method outperforms 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 Adaptations and Experiments with $\alpha > 1$

From our objective function (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 (6) that accommodates $\alpha \geq 1$ while preserving practical applicability. The revised objective is defined as:

$$\widetilde{L}(Q \mid V) = (1-\gamma)\,\mathbb{E}_{s\sim p_0}\left[V(s)\right] - \mathbb{E}_{(s,a)\sim d^U}\left[\exp\left(\Psi(s,a)\right)\left(Q(s,a) - \gamma \mathbb{E}_{s'}[V(s')]\right)\right], \quad (16)$$

which enables empirical investigation into the high-α regime while sidestepping theoretical limitations. The experiment results are provided in Figure 8. Overall, $\alpha \geq 1$ does not provide good performance, which raises the limitation of the naive adaptation.

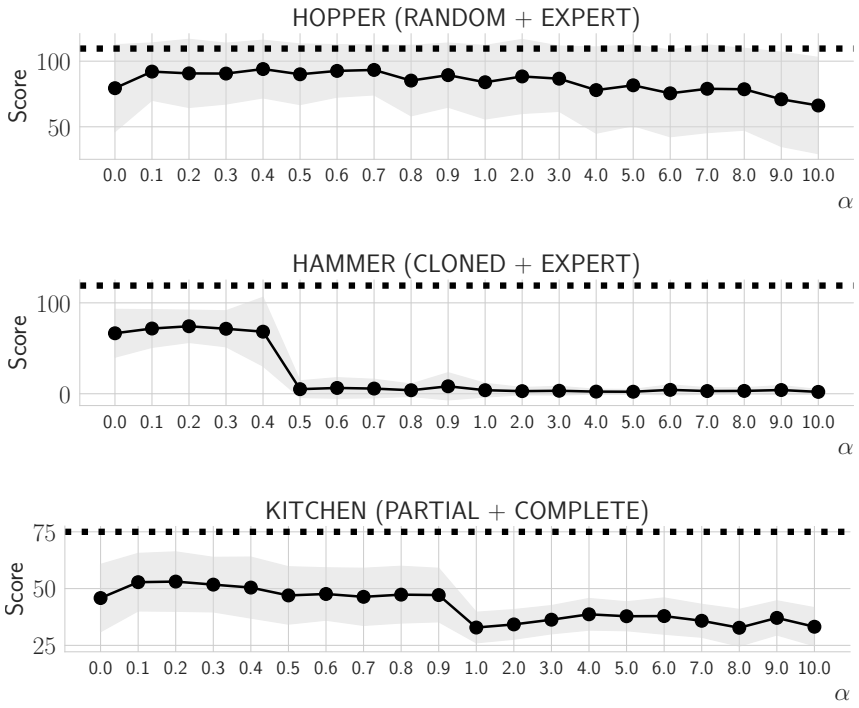

Figure 8: Performance of large $\alpha \geq 1$.

D.7 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 (3)) is transformed into a modified version $\widetilde{L}(Q \mid V)$ (Equation (6)). This experiment investigates the performance differences between the two objectives.

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:

$$\begin{aligned} L(Q, \pi) =& (1-\gamma)\, \mathbb{E}_{s \sim p_0}\left[V_Q^\pi(s)\right] \\ &+ (1-\alpha) \mathbb{E}_{(s,a)\sim d^U}\left[\exp\left(\left(\frac{\Psi(s,a) - \mathcal{T}^\pi[Q](s,a)}{1-\alpha}\right).\text{clip}(\text{minR}, \text{maxR})\right)\right], \end{aligned} \tag{17}$$

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

The results of this ablation study are presented in Figure 9, 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.

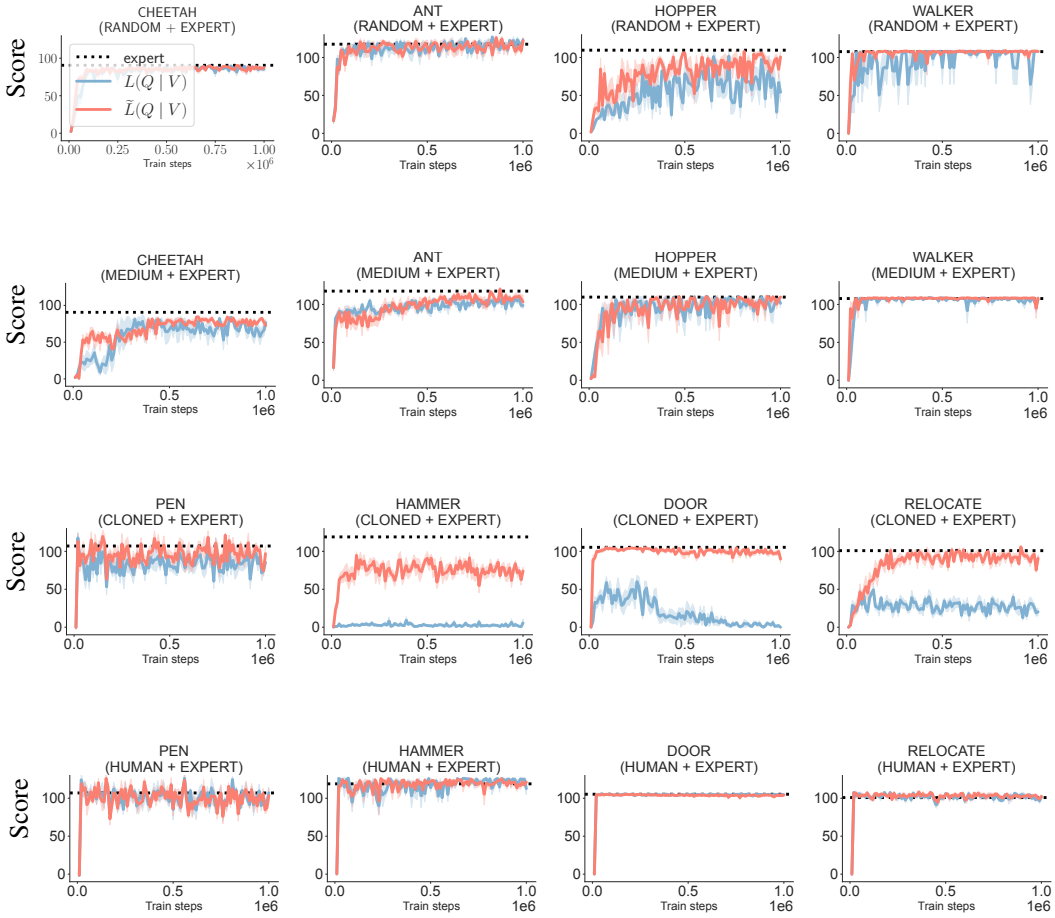

Figure 9: Exponetial ablation study.

D.8 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 5. 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 5: Performance of ContraDICE in different β value in MuJoCo locomotion tasks.