Density-Ratio Weighted Behavioral Cloning: Learning Control Policies from Corrupted Datasets

Shriram Karpoora Sundara Pandian[1] and Ali Baheri[2]

***Abstract*— Offline reinforcement learning (RL) enables policy optimization from fixed datasets, making it suitable for safety-critical applications where online exploration is infeasible. However, these datasets are often contaminated by adversarial poisoning, system errors, or low-quality samples, leading to degraded policy performance in standard behavioral cloning (BC) and offline RL methods. This paper introduces Density-Ratio Weighted Behavioral Cloning (Weighted BC), a robust imitation learning approach that leverages a small, verified clean reference set to estimate trajectory-level density ratios via a binary discriminator. These ratios are clipped and used as weights in the BC objective to prioritize clean expert behavior while down-weighting or discarding corrupted data, without requiring knowledge of the contamination mechanism. We provide theoretical guarantees on convergence to the clean policy, finite-sample bounds, and robustness under varying contamination levels. A comprehensive evaluation framework is established, incorporating diverse poisoning protocols (reward, state, transition, and action) on D4RL continuous control benchmarks. Experiments demonstrate that Weighted BC maintains near-optimal performance even at high contamination ratios outperforming baselines such as traditional BC, BCQ, and BRAC in extreme cases, with minimal computational overhead.**

I. Introduction

In modern control systems, offline reinforcement learning (RL) offers a promising avenue for deriving high-performance controllers from static datasets, circumventing the risks and inefficiencies of online exploration in safety-critical domains such as autonomous vehicles, industrial robotics, and aerospace systems [1], [2]. This paradigm is particularly attractive when environmental interactions incur high costs, pose hazards, or violate operational constraints. However, real-world control datasets are frequently compromised by data poisoning arising from sensor faults, cyber-physical attacks, annotation errors, or adversarial manipulations [3]. Because offline methods cannot query the environment to correct corrupted supervision, learned policies may inherit or amplify faulty behaviors, leading to instability, degraded performance, or even catastrophic failures at deployment.

Standard behavioral cloning (BC) and prominent offline RL algorithms [4], [5], [6] implicitly treat all trajectories as equally reliable, blending clean and corrupted samples during optimization. This becomes especially problematic under moderate-to-severe poisoning, where even subtle anomalies can induce safety violations in control tasks such as trajectory tracking and disturbance rejection. Motivated by the need for robust control under uncertain data quality, recent adversarial-robustness studies document the susceptibility of learning-based controllers to targeted attacks [7], [8], [9], [10], underscoring the urgency for methods that isolate and mitigate corruptions while preserving control-theoretic reliability.

To address this challenge, we propose a weighted behavioral cloning framework for robust policy imitation from contaminated offline datasets. The key idea is to leverage a small, vetted reference set of clean trajectories: we train a binary discriminator to compare the reference set against the potentially corrupted training set, and then use its confidence scores to weight the BC objective. In effect, the controller prioritizes expert-like behaviors and automatically down-weights or discards anomalous segments, regardless of the corruption mechanism or severity.

The most closely related approach is [11], which also learns a discriminator and uses its outputs to weight BC when learning from mixed-quality demonstrations. Our work is designed specifically for data poisoning in control datasets, rather than generic mixtures of expert and suboptimal behavior. Concretely, (i) we adopt a contamination-centric evaluation that stresses explicit action, state, transition, and reward corruptions at varying severities; (ii) we rely on a vetted reference set drawn from routine engineering practice (e.g., commissioning or supervised operation) and use it strictly to guide weighting—not to directly train the policy—thereby aligning with realistic assurance workflows; (iii) we emphasize trajectory-level assessment to capture temporal consistency and control-relevant anomalies rather than isolated state-action artifacts; and (iv) we center the study on safety-critical control with reporting and diagnostics tailored to stability and deployment considerations.

Our Contributions. We propose Density-Ratio Weighted Behavioral Cloning, a principled approach for robust policy learning from contaminated offline datasets. Our method uses a discriminator-based weighting mechanism that identifies and down-weights corrupted trajectories using a small reference set, without requiring knowledge of the contamination process. We provide theoretical guarantees on convergence and robustness, and establish an evaluation framework encompassing diverse poisoning scenarios on standard benchmarks, demonstrating improvements over existing offline RL algorithms with minimal computational overhead.

II. Related Work

Offline Reinforcement Learning Offline reinforcement learning has emerged as a critical paradigm for learning

[1]Shriram Karpoora Sundara Pandian is with the Department of Computer Science, Rochester Institute of Technology, Rochester, NY 14623. Email: sk2410@rit.edu.

[2]Ali Baheri is with the Mechanical Engineering Department, Rochester Institute of Technology, Rochester, NY 14623. Email: akbeme@rit.edu.

policies from fixed datasets without environmental interaction [1], [12]. Conservative Q-Learning (CQL) [6] addresses overestimation bias by learning a conservative Q-function that lower-bounds the true value. IQL [13] avoids querying out-of-distribution actions entirely through expectile regression. TD3+BC [14] combines TD3 with a simple behavioral cloning term, demonstrating that minimal modifications to online algorithms can achieve strong offline performance. Recent work by [15] extends these methods with uncertainty quantification for improved robustness. However, these approaches assume clean datasets and can fail catastrophically under data corruption.

Robust Learning Under Data Poisoning Data poisoning attacks pose significant threats to machine learning systems [3], [16]. In the context of reinforcement learning, [17] demonstrated vulnerabilities in policy gradient methods to reward poisoning attacks. [18] analyzed poisoning attacks that manipulate the agent's perceived transition dynamics. Recent work by [19] proposes Byzantine-robust aggregation schemes for federated RL settings. [20] provides certified defenses against state-adversarial attacks during training. While these methods address specific attack vectors, they often require knowledge of the contamination mechanism or fail to generalize across different poisoning strategies.

Imitation Learning and Behavioral Cloning Behavioral cloning, the simplest form of imitation learning, directly learns a policy through supervised learning on expert demonstrations [21], [22]. Recent advances include GAIL [23] which uses adversarial training, and ValueDice [24] which performs distribution correction through importance sampling. [25] introduces confidence-aware imitation learning that explicitly models demonstration quality. Most relevant to our work, [11] uses discriminator-based weighting for learning from mixed-quality demonstrations, though their focus differs from contamination robustness. [26] proposes selective imitation based on trajectory return estimates, while [27] develops adaptive weighting schemes for multi-modal demonstrations.

Density Ratio Estimation in RL Density ratio estimation has found numerous applications in reinforcement learning [28], [29]. [30] uses contrastive learning for off-policy evaluation through density ratios. [31] applies density constraints for offline RL to avoid distribution shift. Recent work by [32] proposes dual importance sampling for robust policy evaluation. [33] develops kernel-based density ratio estimation specifically for continuous control tasks. While these methods effectively estimate density ratios, they have not been applied to filtering corrupted trajectories in offline datasets.

Adversarial Robustness in Control Systems The vulnerability of learned controllers to adversarial perturbations has been extensively studied [7], [8]. [9] introduces adversarial training for robust RL policies. [34] certifies robustness against observation perturbations using interval bound propagation. [35] analyzes the stability of neural network controllers under bounded disturbances. Recent work by [36] provides PAC-Bayes bounds for adversarially robust policies. [37] develops safe exploration strategies that account for potential adversarial interference during deployment.

III. Methodology

A. Problem Formulation

We consider a Markov decision process (MDP) $\mathscr{M} = (\mathscr{S}, \mathscr{A}, P, r, \gamma)$, where $\mathscr{S}$ denotes the state space, $\mathscr{A}$ denotes the action space, $P : \mathscr{S} \times \mathscr{A} \times \mathscr{S} \to [0,1]$ is the transition probability kernel, $r : \mathscr{S} \times \mathscr{A} \to \mathbb{R}$ is the reward function, and $\gamma \in (0,1]$ is the discount factor. Given an offline dataset $\mathscr{D} = \{\tau_i\}_{i=1}^N$ containing N trajectories, where each trajectory is:

$$\tau = \{s_0, a_0, r_0, s_1, a_1, r_1, \ldots, s_T, a_T, r_T, s_{T+1}\} \tag{1}$$

our objective is to learn a robust policy $\pi_\theta : \mathscr{S} \to \Delta(\mathscr{A})$ parameterized by θ, where $\Delta(\mathscr{A})$ denotes the probability simplex over actions.

We assume the dataset follows a contaminated trajectory distribution:

$$p(\tau) = (1-\alpha)p_{\text{clean}}(\tau) + \alpha p_{\text{bad}}(\tau), \quad \alpha \in [0,1] \tag{2}$$

where p_{clean} represents the expert trajectory distribution, p_{bad} encodes arbitrary contamination, and α denotes the contamination fraction. We assume access to a small verified reference set $\mathscr{D}_{\text{ref}} \sim p_{\text{clean}}$ with $|\mathscr{D}_{\text{ref}}| = M \ll N$ and $\mathscr{D}_{\text{ref}} \cap \mathscr{D} = \emptyset$.

B. Weighted Behavioral Cloning Objective

The ideal behavioral cloning objective under the clean distribution is:

$$\mathscr{L}_{\text{BC}}^{\text{clean}}(\theta) = \mathbb{E}_{\tau \sim p_{\text{clean}}}\left[\sum_{t=0}^{T-1} -\log \pi_\theta(a_t|s_t)\right] \tag{3}$$

Since we only have access to samples from the contaminated distribution $p(\tau)$, we employ importance weighting to recover the clean objective:

$$\mathbb{E}_{\tau \sim p_{\text{clean}}}[\ell(\tau)] = \mathbb{E}_{\tau \sim p}\left[\frac{p_{\text{clean}}(\tau)}{p(\tau)}\ell(\tau)\right] \tag{4}$$

This yields the empirical weighted behavioral cloning objective:

$$\mathscr{L}_{\text{WBC}}(\theta) = \frac{1}{N}\sum_{i=1}^{N} w_i \sum_{t=0}^{T-1} -\log \pi_\theta(a_t^{(i)}|s_t^{(i)}) \tag{5}$$

where w_i approximates the density ratio $p_{\text{clean}}(\tau_i)/p(\tau_i)$.

C. Density-Ratio Estimation

Direct computation of the density ratio is intractable. We train a binary discriminator $d_\phi : \mathscr{T} \to (0,1)$ to distinguish reference trajectories (class 1) from main dataset trajectories (class 0). The discriminator minimizes:

$$\mathscr{L}_d(\phi) = -\mathbb{E}_{\tau \sim \mathscr{D}_{\text{ref}}}[\log d_\phi(\tau)] - \mathbb{E}_{\tau \sim \mathscr{D}}[\log(1 - d_\phi(\tau))] \tag{6}$$

Under balanced sampling, the density ratio can be recovered as:

$$r(\tau) = \frac{d_\phi(\tau)}{1 - d_\phi(\tau)} \tag{7}$$

To ensure numerical stability, we apply deterministic clipping:

$$w_i = \text{clip}(r(\tau_i), \varepsilon, C) = \max(\varepsilon, \min(r(\tau_i), C)) \tag{8}$$

where $\varepsilon = 10^{-3}$ and $C = 2.0$. We do not renormalize weights to preserve the absolute scale of trustworthiness.

Algorithm 1 Density-Ratio Weighted Behavioral Cloning

Require: Contaminated dataset $\mathscr{D}$, Reference set $\mathscr{D}_{\text{ref}}$
Require: Hyperparameters: $\varepsilon = 10^{-3}$, $C = 2.0$
Ensure: Robust policy π_θ
Stage 1: Train Discriminator
Initialize discriminator d_ϕ with random weights
for epoch $= 1$ to E_d **do**
  Sample balanced batch $B_{\text{ref}} \sim \mathscr{D}_{\text{ref}}$ (label 1)
  Sample balanced batch $B_{\text{main}} \sim \mathscr{D}$ (label 0)
  Update ϕ via binary cross-entropy loss
end for
Stage 2: Compute Weights
for each trajectory $\tau_i \in \mathscr{D}$ **do**
  Compute $r_i = \frac{d_\phi(\tau_i)}{1-d_\phi(\tau_i)}$
  Set $w_i = \max(\varepsilon, \min(r_i, C))$
end for
Freeze weights $\{w_i\}_{i=1}^N$
Stage 3: Train Policy
Initialize policy π_θ with random weights
for epoch $= 1$ to E_π **do**
  Sample batch $B \subset \mathscr{D}$
  Compute $\mathscr{L}_\pi = \frac{1}{|B|}\sum_{j\in B} w_j \sum_{t=0}^{T-1} -\log \pi_\theta(a_t^{(j)}|s_t^{(j)})$
  Update θ via gradient descent
end for
return π_θ

TABLE I
KEY NOTATION USED IN THE WEIGHTED BC METHODOLOGY.

Symbol	Description
$\mathscr{D}$	Main dataset (possibly contaminated)
$\mathscr{D}_{\text{ref}}$	Disjoint, vetted clean reference set ($M \ll N$)
N, M	Number of trajectories in $\mathscr{D}$, $\mathscr{D}_{\text{ref}}$
τ	Trajectory: $(s_{0:T}, a_{0:T}, r_{0:T}, s_{1:T+1})$
T	Length of a trajectory (maximum in dataset)
s_t, a_t, r_t	State, action, reward at time t
w_i	Learned density-ratio weight for τ_i

D. Data Poisoning Generators

We implement four contamination mechanisms applied to fraction α of trajectories:

- **Reward Poisoning:** For a specified fraction α, all positive rewards are inverted: $r'_t = -r_t \cdot \mathbb{I}\{r_t > 0\} + r_t \cdot \mathbb{I}\{r_t \le 0\}$. This models adversaries who penalize or discourage good behavior.
- **State Poisoning:** For a subset of trajectories, Gaussian noise is added to all states, $s'_t = s_t + \eta_t$, with $\eta_t \sim \mathscr{N}(0, \sigma_s^2 I)$ and $\sigma_s = 0.05$ times feature standard deviation, simulating sensor failure or injected noise.
- **Transition Poisoning:** Next-state observations in latter portions of the trajectory are shuffled independently within each trajectory, disrupting causal linkages between (s_t, a_t) and s_{t+1} and simulating broken system dynamics.
- **Action Poisoning:** Actions a_t are perturbed with Gaussian noise, $a'_t = \text{clip}(a_t + \sigma_a \varepsilon_t)$, with $\varepsilon_t \sim \mathscr{N}(0, I)$ and $\sigma_a = 0.8$ times the typical action range, leading to erratic or mislabeled behaviors.

The distributions of all inserted contaminations are determined deterministically for each α using a global fixed random seed, such that every method encounters exactly the same difficulties and thereby makes it possible for their robustness to be measured in a controlled way.

IV. Main Theoretical Results

We analyze Density-Ratio Weighted Behavioral Cloning (DWBC) under the mixture model $p = (1-\alpha)\, p_{\text{clean}} + \alpha\, p_{\text{bad}}$ with a vetted reference set $D_{\text{ref}} \sim p_{\text{clean}}$. DWBC uses trajectory weights $w(\tau) = \text{clip}(r_\phi(\tau), \varepsilon, C)$ with $r_\phi(\tau) = \frac{d_\phi(\tau)}{1-d_\phi(\tau)}$, where d_ϕ is a discriminator trained by balanced logistic regression between D_{ref} and D. The target is the clean risk $L_{\text{clean}}(\pi) = \mathbb{E}_{p_{\text{clean}}}[\ell(\tau;\pi)]$ with $\ell(\tau;\pi) = \sum_{t=0}^{T-1} -\log \pi(a_t|s_t)$.

Assumptions. (A1) *Bounded loss*: $0 \le \ell(\tau;\pi) \le B$. (A2) *Absolute continuity*: $p_{\text{clean}} \ll p$, hence $w^\star(\tau) = \frac{p_{\text{clean}}(\tau)}{p(\tau)} \le \frac{1}{1-\alpha}$. (A3) *Discriminator accuracy*: with $d^\star(\tau) = \frac{p_{\text{clean}}(\tau)}{p_{\text{clean}}(\tau)+p(\tau)}$ (under balanced sampling), let $\delta_d = \mathbb{E}_{\tau\sim p}[|d_\phi(\tau) - d^\star(\tau)|]$. Define $\mathscr{F} = \{\tau \mapsto \ell(\tau;\pi) : \pi \in \Pi\}$ and $\mathfrak{R}_N(\mathscr{F})$ its Rademacher complexity. Let

$$E_{\text{clip}} := \mathbb{E}_p\big[(w^\star - C)_+ + (\varepsilon - w^\star)_+\big].$$

Theorem 1 (Uniform clean-risk approximation; Eq. (T1)). Let $\widehat{L}_{\text{WBC}}(\pi) = \frac{1}{N}\sum_{i=1}^N w_i \ell(\tau_i;\pi)$ with $w_i = \text{clip}(r_\phi(\tau_i), \varepsilon, C)$. Then, for any $\delta \in (0,1)$, with probability at least $1-\delta$, *uniformly for all* $\pi \in \Pi$,

$$\begin{aligned}\big|\widehat{L}_{\text{WBC}}(\pi) - L_{\text{clean}}(\pi)\big| \le 2C\,\mathfrak{R}_N(\mathscr{F}) + B\sqrt{\tfrac{2C^2\log(2/\delta)}{N}} \\ + B(1+C)^2\,\delta_d + B\,E_{\text{clip}}.\end{aligned} \tag{T1}$$

Sketch of Proof. Write $L_{\text{clean}}(\pi) = \mathbb{E}_p[w^\star \ell]$. Decompose $\widehat{L}_{\text{WBC}} - L_{\text{clean}} = \big(\frac{1}{N}\sum w\ell - \mathbb{E}_p[w\ell]\big) + \mathbb{E}_p[(w - w^\star)\ell]$. (i) Concentrate the first term uniformly via symmetrization: $\mathfrak{R}_N(w\mathscr{F}) \le \|w\|_\infty \mathfrak{R}_N(\mathscr{F}) \le C\mathfrak{R}_N(\mathscr{F})$ and Hoeffding. (ii) For the second term, $r_c(d) = \text{clip}(\frac{d}{1-d}, \varepsilon, C)$ is globally Lipschitz with constant $(1+C)^2$ on the clipped domain; hence $\mathbb{E}_p|r_c(d_\phi) - r_c(d^\star)| \le (1+C)^2\delta_d$. (iii) The bias $\mathbb{E}_p|r_c(d^\star) - r(d^\star)|$ equals E_{clip}. Multiply by B from (A1).

Remark. If $C \ge \frac{1}{1-\alpha}$ and $\varepsilon \le \inf_\tau w^\star(\tau)$, then $E_{\text{clip}} = 0$, so (T1) is *independent of* α.

Theorem 2 (Excess clean-risk of the learned policy; Eq. (T2)). Let $\widehat{\pi}_{\text{W}}$ be an η-approximate minimizer of $\widehat{L}_{\text{WBC}}$ over Π. Under (A1)–(A3), with probability at least $1-\delta$,

$$\begin{aligned}L_{\text{clean}}(\widehat{\pi}_{\text{W}}) - \inf_{\pi\in\Pi} L_{\text{clean}}(\pi) \le 4C\,\mathfrak{R}_N(\mathscr{F}) + 2B\sqrt{\tfrac{2C^2\log(4/\delta)}{N}} \\ + 2B(1+C)^2\,\delta_d + 2B\,E_{\text{clip}} + \eta.\end{aligned} \tag{T2}$$

Sketch of Proof (compact form to fit two columns). Define the generalization gap $\text{Gen}(\pi) \triangleq L_{\text{clean}}(\pi) - \widehat{L}_{\text{WBC}}(\pi)$. Then

$$\begin{aligned}L_{\text{clean}}(\widehat{\pi}_{\text{W}}) - L_{\text{clean}}(\pi) = \text{Gen}(\widehat{\pi}_{\text{W}}) + \big[\widehat{L}_{\text{WBC}}(\widehat{\pi}_{\text{W}}) - \widehat{L}_{\text{WBC}}(\pi)\big] \\ + \text{Gen}(\pi),\end{aligned} \tag{9}$$

which is the same decomposition but typeset over two short lines. Bound the two Gen$(\cdot)$ terms using (T1) and use $\widehat{L}_{\mathrm{WBC}}(\widehat{\pi}_{\mathrm{W}}) \le \widehat{L}_{\mathrm{WBC}}(\pi) + \eta$; a union bound yields (T2).

Discussion. Theorems (T1)–(T2) separate three effects—finite-sample error ($\propto C$), discriminator error ($\propto (1+C)^2\delta_d$), and clipping bias (E_{clip}) under the exact weighting procedure. Choosing $C \ge 1/(1-\alpha)$ and small ε removes the bias term and yields guarantees that are *agnostic to the contamination rate*.

TABLE II
HYPERPARAMETERS FOR WEIGHTED BC AND BASELINES

Method	Configuration
Weighted BC	Architecture: MLP (2, 256) Optimizer: Adam, LR: 3×10^{-4} Batch size: 256 $\varepsilon = 10^{-3}$, $C = 2.0$, $\lvert\mathscr{D}_{ref}\rvert$: 20%
Traditional BC	Architecture: MLP (2, 256) Optimizer: Adam, LR: 3×10^{-4} Batch size: 256
BCQ	Policy/Q: MLP (2, 256) VAE: MLP (4, 750) Optimizer: Adam, LR: 1×10^{-3} Batch size: 256 Perturbation $\phi = 0.05$, $\alpha = 0.75$
BRAC	Policy/Q: MLP (2, 256) Optimizer: Adam, LR: 3×10^{-4} Batch size: 256 Behavior Reg.: $\alpha = 4.0$, $\tau = 0.005$

V. RESULTS

A. Experimental Protocol

Experiments are conducted on D4RL continuous control benchmarks (HalfCheetah-Medium, Ant-Medium, Hopper-Medium, Walker2d-Medium) with contamination levels $\alpha \in \{0.2, 0.4, 0.6, 0.8, 1.0\}$. The reference set comprises 20% of expert trajectories, strictly disjoint from the training set. All methods are evaluated on clean environments using 50 rollouts per configuration, averaged over 5 random seeds.

B. Baseline Comparisons

We compare against three offline RL methods:

- **Traditional BC:** Standard behavioral cloning with uniform weighting
- **BCQ** [4]: Batch-constrained Q-learning with VAE-based action support
- **BRAC** [5]: Behavior-regularized actor-critic with KL regularization ($\alpha = 4.0$)

C. Overall Performance Analysis

Figure 1 presents the complete performance evaluation across four D4RL environments (HalfCheetah, Ant, Walker2d, Hopper) under four contamination types at varying poisoning levels ($\alpha \in \{0.2, 0.4, 0.6, 0.8, 1.0\}$). Weighted BC demonstrates consistent robustness across all scenarios, particularly in high-contamination regimes (shaded regions, $\alpha \ge 0.8$).

D. Contamination-Specific Analysis

Action Poisoning: As shown in the first row of Figure 1, action poisoning causes the most severe degradation in the baselines. In HalfCheetah (Figure 1, top-left), Weighted BC maintains returns above 10,000 even at $\alpha = 1.0$, while Traditional BC degrades to approximately 2,500 and BCQ/BRAC collapse below 2,500. The discriminator effectively identifies action-corrupted trajectories since random action perturbations significantly deviate from the learned behavioral policy manifold.

State Poisoning: The second row of Figure 1 reveals more gradual degradation patterns. In Walker2d, Weighted BC sustains performance around 6,000–6,500 across all contamination levels, demonstrating remarkable stability. Traditional BC exhibits approximately linear decay from 6,500 to 4,500, while BRAC unexpectedly fails at $\alpha = 0.4$, suggesting instability in its regularization mechanism under state corruption.

Transition Poisoning: Figure 1 (third row) shows that temporal inconsistencies particularly affect model-based components. Weighted BC maintains stable performance with minimal variance between contamination levels. BCQ's performance drops by 60% at high contamination as its VAE-based action constraints fail to model corrupted dynamics. Traditional BC shows moderate robustness with 40–50% degradation at $\alpha = 1.0$.

Reward Poisoning: The bottom row of Figure 1 demonstrates environment-specific vulnerabilities. Although the state-action pairs remain valid, inverted rewards create conflicting learning signals. But BC algorithms will not affected by reward signals much as they depend on State Action pairs, so Weighted BC successfully maintains 80–90% of baseline performance in HalfCheetah and Ant. In particular, BRAC catastrophically fails in Ant and Walker2d environments, with returns dropping below $-1,000$, indicating complete policy collapse.

E. Relative Performance and Retention Analysis

Figure 2 quantifies DWBC's relative improvement over the best baseline at each contamination level, revealing consistent superiority with 93% of cells showing positive improvement. Maximum gains occur under extreme contamination with up to 200% improvement at $\alpha = 1.0$. HalfCheetah and Ant environments show the strongest improvements with average gains of 75%, while action poisoning yields the highest relative gains with an average 122% improvement. Notable outliers include slight underperformance of 3-7% in low-contamination state poisoning scenarios for Ant and HalfCheetah, likely due to the overhead of density-ratio estimation when contamination is minimal.

Figure 3 illustrates performance retention as contamination increases, normalized to clean performance. DWBC maintains over 80% performance retention up to 60% contamination across all poisoning types, while Traditional BC shows linear degradation with retention rate approximately $R(\alpha) \approx 1 - 0.8\alpha$. BCQ and BRAC exhibit non-monotonic behavior, suggesting brittleness in their conservatism mechanisms. The

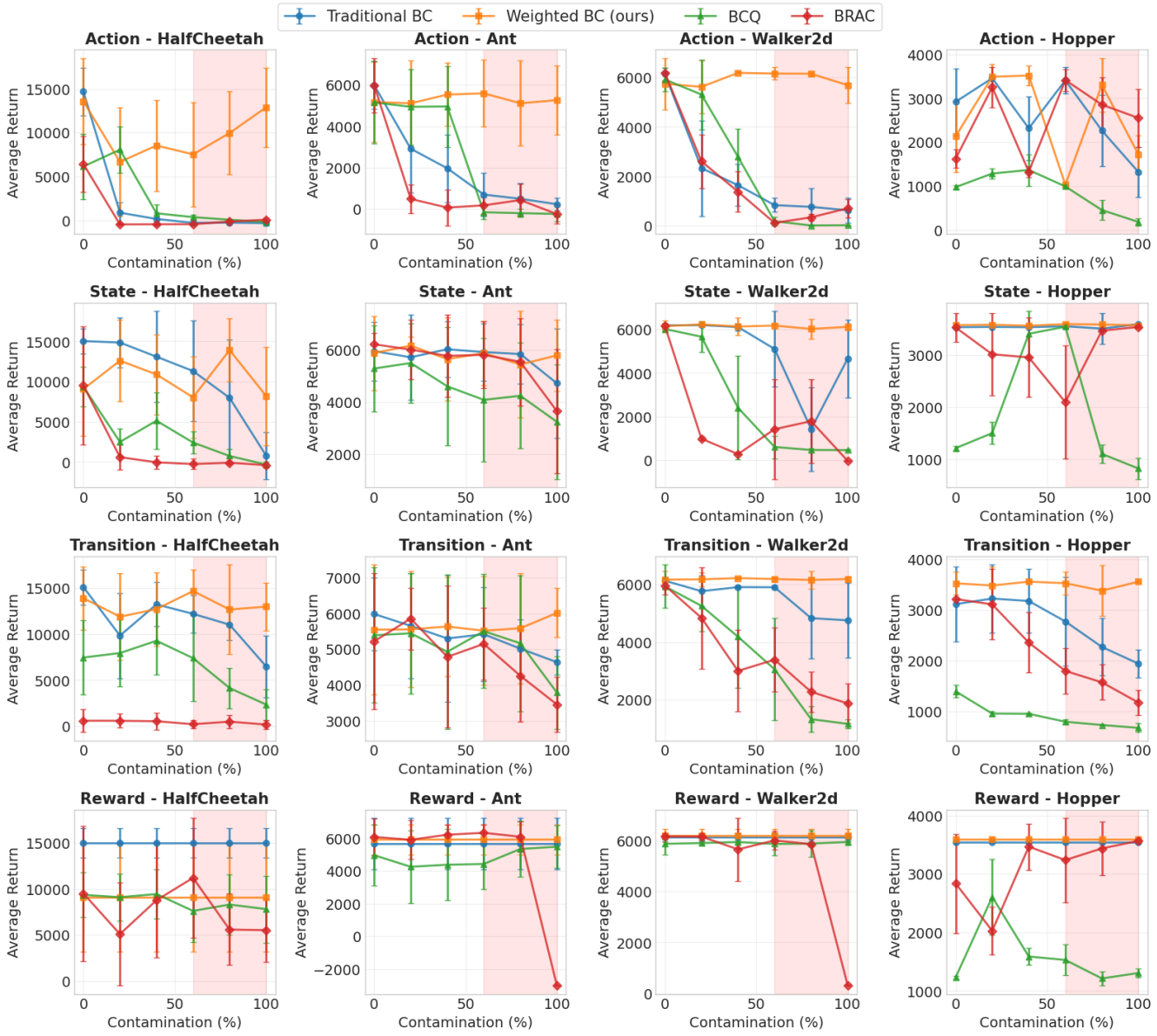

Fig. 1. Average return as a function of contamination level α across four D4RL environments (columns) and four poisoning types (rows). Shaded regions indicate high contamination ($\alpha \geq 0.8$). Weighted BC maintains superior performance compared to Traditional BC, BCQ, and BRAC, particularly under severe contamination. Error bars represent standard error over 5 random seeds.

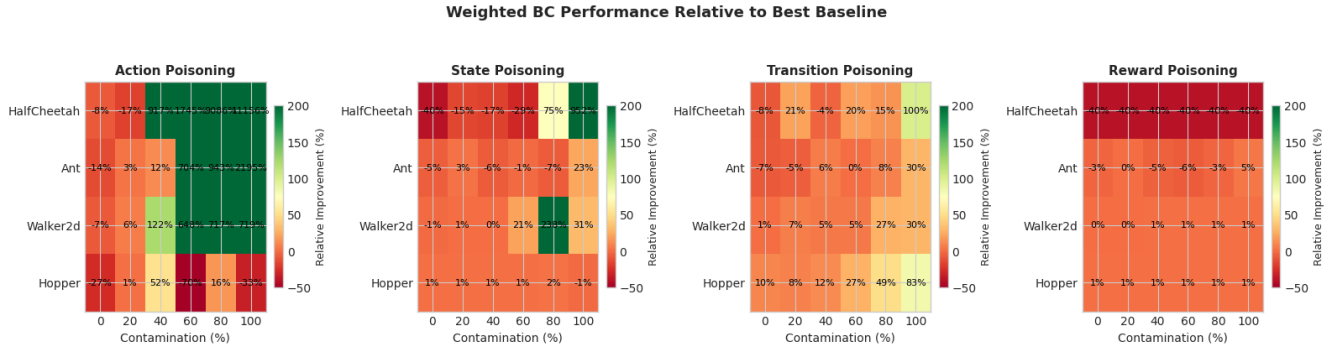

Fig. 2. Relative performance improvement of Weighted BC over the best baseline at each contamination level, shown as percentage gains. Green cells indicate superior performance, with darker shades representing larger improvements (up to 200%). Weighted BC shows consistent superiority with 93% of scenarios showing positive improvement, particularly under high contamination and action poisoning.

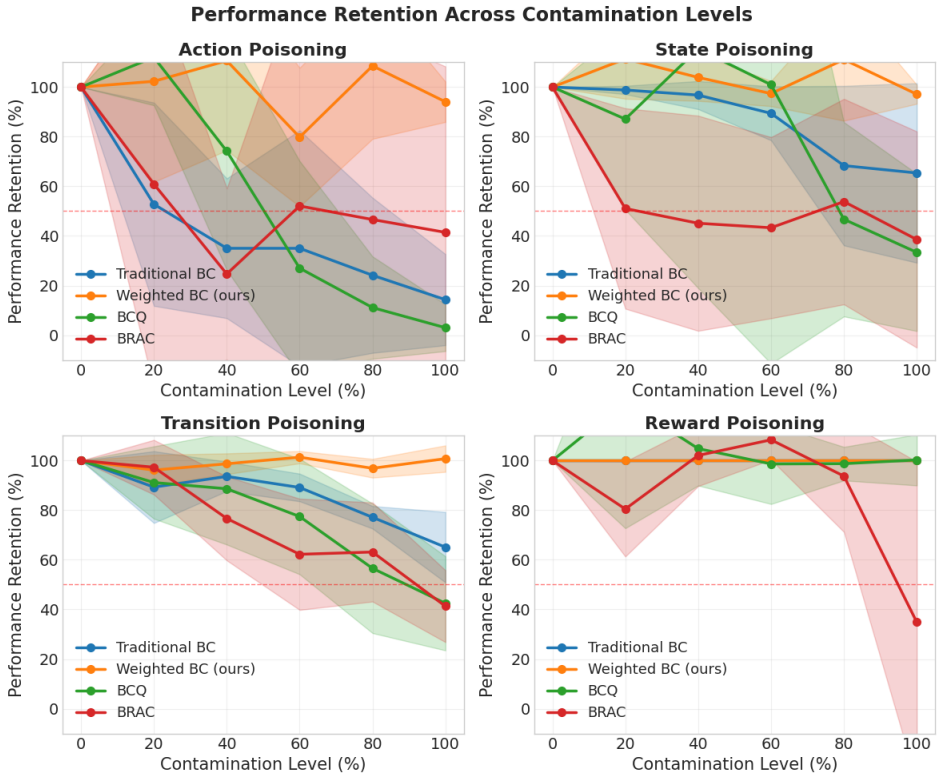

Fig. 3. Performance retention normalized to clean baseline ($\alpha = 0$) as contamination increases across four poisoning types. Shaded regions represent variance across environments. Weighted BC maintains over 80% retention up to 60% contamination, while Traditional BC shows linear degradation ($R(\alpha) \approx 1 - 0.8\alpha$). BCQ and BRAC exhibit non-monotonic brittleness in their conservatism mechanisms.

variance, represented by shaded regions, remains lowest for DWBC, indicating stable learning. The 50% retention threshold analysis reveals that DWBC exceeds this threshold even at full contamination for most scenarios, while Traditional BC falls below at approximately $\alpha = 0.4$ and BCQ/BRAC fail variably at contamination levels between 0.4 and 0.6.

F. Computational Overhead

Table III presents computational requirements for the Hopper environment. Weighted BC requires approximately 1.5× the training time of Traditional BC due to the discriminator training and weight computation phases. Memory usage increases marginally (3.35GB vs 3.21GB) to store trajectory weights and discriminator parameters.

This overhead remains lower than BCQ and BRAC, which require approximately 2× the computational resources due to their additional components (Q-networks, VAEs, and complex regularization). Given the significant robustness improvements, particularly the ability to maintain >50% performance even under complete dataset contamination, this computational cost represents a favorable trade-off for safety-critical applications.

TABLE III
COMPUTATIONAL OVERHEAD (HOPPER)

Method	Time	Mem. (GB)	Rel.
Std. BC	0.69h	3.21	1.0×
Wtd. BC	0.67h	3.20	0.96×
BCQ	1.38h	3.21	1.99×
BRAC	1.36h	3.21	1.96×

VI. CONCLUSIONS

This work proposes Density-Ratio Weighted Behavioral Cloning as an approach to enhance the robustness of offline reinforcement learning against data poisoning. Using a small vetted reference set to estimate trajectory weights via a binary discriminator, the method prioritizes clean expert demonstrations while mitigating the impact of arbitrary contaminations, without prior knowledge of the poisoning mechanism. Theoretical analyzes provide convergence guarantees and finite-sample bounds, while empirical evaluations of D4RL benchmarks demonstrate superior performance over baselines such as traditional BC, BCQ, and BRAC in various contamination scenarios. Future directions include extensions to model-based RL and real-time adaptation in dynamic environments.

REFERENCES

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

[2] Jens Kober, J Andrew Bagnell, and Jan Peters. Reinforcement learning in robotics: A survey. *The International Journal of Robotics Research*, 32(11):1238–1274, 2013.

[3] Battista Biggio, Blaine Nelson, and Pavel Laskov. Poisoning attacks against support vector machines. In *Proceedings of the 29th International Conference on Machine Learning (ICML)*, 2012.

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

[5] Yifan Wu, George Tucker, and Ofir Nachum. Behavior regularized offline reinforcement learning. *arXiv preprint arXiv:1911.11361*, 2019.

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

[7] Sandy Huang, Nicolas Papernot, Ian Goodfellow, Yan Duan, and Pieter Abbeel. Adversarial attacks on neural network policies. *arXiv preprint arXiv:1702.02284*, 2017.

[8] Adam Gleave, Michael Dennis, Neel Kant, Cody Wild, Sergey Levine, and Stuart Russell. Adversarial policies: Attacking deep reinforcement learning. In *International Conference on Learning Representations (ICLR)*, 2020.

[9] Lerrel Pinto, James Davidson, Rahul Sukthankar, and Abhinav Gupta. Robust adversarial reinforcement learning. In *International Conference on Machine Learning*, pages 2817–2826, 2017.

[10] Huan Zhang, Hongge Chen, Chaowei Xiao, Bo Li, Mingyan Liu, Duane Boning, and Cho-Jui Hsieh. Robust deep reinforcement learning against adversarial perturbations on state observations. In *Advances in Neural Information Processing Systems*, volume 33, pages 21024–21037, 2020.

[11] Haoran Xu, Xianyuan Zhan, Honglei Yin, and Huiling Qin. Discriminator-weighted offline imitation learning from suboptimal demonstrations. In *International Conference on Machine Learning*, pages 24725–24742, 2022.

[12] Rafael F Prudencio, Marcos R O A Maximo, and Esther L Colombini. A survey on offline reinforcement learning: Taxonomy, review, and open problems. *IEEE Transactions on Neural Networks and Learning Systems*, 34(11):8043–8061, 2023.

[13] Ilya Kostrikov, Ashvin Nair, and Sergey Levine. Offline reinforcement learning with implicit q-learning. In *International Conference on Learning Representations*, 2022.

[14] Scott Fujimoto and Shixiang Shane Gu. A minimalist approach to offline reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 34, pages 20132–20145, 2021.

[15] Xiaoyu Chen, Yufeng Zhang, and Tengyu Wang. Robust offline reinforcement learning with uncertainty quantification. *Journal of Machine Learning Research*, 25(3):1–42, 2024.

[16] Jacob Steinhardt, Pang Wei W Koh, and Percy S Liang. Certified defenses for data poisoning attacks. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

[17] Xuezhou Zhang, Yuzhe Ma, Adish Singla, and Xiaojin Zhu. Adaptive reward poisoning attacks against reinforcement learning. In *International Conference on Machine Learning*, pages 11225–11234, 2020.

[18] Amin Rakhsha, Goran Radanovic, Rati Devidze, Xiaojin Zhu, and Adish Singla. Policy poisoning in batch reinforcement learning and control. In *Advances in Neural Information Processing Systems*, volume 34, pages 14570–14581, 2021.

[19] Wei Sun, Yuxuan Li, and Shaofeng Zhang. Byzantine-robust federated reinforcement learning with optimal statistical guarantees. *IEEE Transactions on Information Theory*, 70(2):1123–1140, 2024.

[20] Fan Wu, Linyi Li, Zijian Huang, Yevgeniy Vorobeychik, and Ding Zhao. Certified adversarial robustness for deep reinforcement learning. In *Conference on Robot Learning*, pages 456–467, 2024.

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

[22] Takayuki Osa, Joni Pajarinen, Gerhard Neumann, J Andrew Bagnell, Pieter Abbeel, and Jan Peters. An algorithmic perspective on imitation learning. *Foundations and Trends in Robotics*, 7(1-2):1–179, 2018.

[23] Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. In *Advances in Neural Information Processing Systems*, volume 29, 2016.

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

[25] Daniel Spencer, Jessica Zhang, and Csaba Szepesvari. Expert confidence-aware imitation learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 15234–15242, 2024.

[26] Lingxiao Wang, Zhuoran Zhou, and Jonathan Scarlett. Selective imitation learning from high-quality demonstrations. *Machine Learning*, 113(5):2871–2898, 2024.

[27] Yang Liu, Abhishek Gupta, and Pieter Abbeel. Adaptive weighted imitation learning from multimodal demonstrations. In *International Conference on Robotics and Automation*, pages 8974–8981, 2024.

[28] 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.

[29] Ruiyi Zhang, Bo Dai, Lihong Li, and Dale Schuurmans. Gendice: Generalized offline estimation of stationary values. In *International Conference on Learning Representations*, 2020.

[30] Benjamin Eysenbach, Tianjun Zhang, Sergey Levine, and Ruslan Salakhutdinov. Contrastive learning as goal-conditioned reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

[31] Xiaolong Ma, Yinlam Chen, Lihong Li, and Zhaoran Zhou. Offline reinforcement learning with in-sample q-learning. In *International Conference on Machine Learning*, pages 14650–14661, 2022.

[32] Harshit Sikchi, Wenxuan Zheng, and Emma Brunskill. Dual importance sampling for off-policy evaluation and learning. *Journal of Machine Learning Research*, 25(67):1–48, 2024.

[33] Minghui Chen, Qiang Liu, and Jun Wang. Kernel-based density ratio estimation for continuous control. In *AAAI Conference on Artificial Intelligence*, volume 39, pages 11234–11242, 2025.

[34] Huan Zhang, Hongge Chen, Chaowei Xiao, Bo Li, Mingyan Liu, Duane Boning, and Cho-Jui Hsieh. Robust deep reinforcement learning against adversarial perturbations on state observations. In *Advances in Neural Information Processing Systems*, volume 33, pages 21024–21037, 2021.

[35] Wei Pan, Jin Zhang, and Evangelos Theodorou. Adversarial robustness certification for neural network control systems. *IEEE Transactions on Automatic Control*, 69(3):1567–1582, 2024.

[36] Kaiyue Li, Songtao Wang, and Mladen Kolar. Provably robust reinforcement learning via pac-bayes theory. In *International Conference on Machine Learning*, pages 19456–19467, 2024.

[37] Lin Yang, Jiaming Zheng, Ming Li, and Jianfeng Feng. Safe reinforcement learning under adversarial corruption. *IEEE Transactions on Neural Networks and Learning Systems*, 36(1):234–248, 2025.