

497 **Appendix**

498 The appendix contains the following details:

499 **Missing Proofs:** Refer to Appendix B for proofs omitted from the main paper.

500 **Experimental Details:** We provide information on:

- 501 • Baseline implementation (Appendix D.3)
- 502 • Hyper-parameter selection for each task domain (Appendix D.4)
- 503 • Task descriptions (Appendix D.1)
- 504 • Generation of undesirable and unlabeled datasets (Appendix D.2)

505 **Additional Experiments:** We address the following remaining questions:

- 506 • **(Q2)** How does the presence of an undesirable dataset contribute to the performance of
507 UNIQ and other baselines? (Appendix E.1)
- 508 • **(Q3)** How does the quality of unlabeled dataset affect the performance of all the algorithms?
509 (Appendix E.2 and Appendix E.3)
- 510 • **(Q4)** What happens if we do not use the unlabeled dataset (the training is solely based on
511 the undesirable dataset)? (Appendix E.5)
- 512 • **(Q5)** We use WBC for the policy extraction; what if we directly extract the policy from the
513 Q-function? (Appendix E.4)

514 Additionally, we present experiments demonstrating the performance of UNIQ and other baseline
515 methods on the dataset from the safeDICE paper (see Appendix E.12), performance in the Mujoco-
516 velocity benchmark(see Appendix E.11), comparison results using CVaR costs (see Appendix E.9),
517 How does τ effect to the overall results? (Appendix E.6), What if we want to avoid all possible
518 undesirable performance in Safetygym (low-reward low-cost, low-reward high-cost, high-reward
519 high-cost)? (Appendix E.8), and how to control the conservativeness in the Safety-gym tasks(see
520 Appendix E.10).

521 A Pseudo Code of UNIQ

Below we present a pseudo code of our UNIQ algorithm.

Algorithm 1 UNIQ: UNdesired Demonstrations driven Inverse Q-Learning

Require: $\mathcal{D}^{\text{UN}}, \mathcal{D}^{\text{MIX}}, \vartheta_\phi, \pi_\theta, N_\mu, N, Q_{w_q}$ and V_{w_v} networks.
1: **# Estimating the occupancy correction τ^***
2: **for** certain number of iterations: $i = 1 \dots N_\mu$ **do**
3: Update (ϕ) to maximize $g(\vartheta_\phi)$.
4: **end for**
5: **# Train Q and V, and policy functions**
6: **for** certain number of iterations $i = 1 \dots N$ **do**
7: Update w_q to minimize $\tilde{F}(Q_{w_q} | V_{w_v})$
8: Update w_v to minimize the Extreme-V function: $J(V_{w_v} | Q_{w_q})$
9: Update θ via the WBC: $\max_\pi \left\{ \sum_{(s,a) \sim \mathcal{D}^{\text{MIX}}} \exp(A(s,a)) \log \pi(a|s) \right\}$
10: **end for**

522

523 B Missing Proofs

524 We provide proofs that are omitted in the main paper.

525 B.1 Proof of Proposition 4.1

526 **Proposition.** *The following statements hold:*

- 527 (i) *The function $L(\pi, Q)$ is convex in π and the problem $\min_\pi L(\pi, Q)$ has a unique optimal*
528 *solution at $\pi^Q(s, a) = \frac{\exp(Q(s,a)/\beta)}{\sum_{a'} \exp(Q(s,a')/\beta)}$.*
529 (ii) *The learning objective function can be simplified as:*

$$\min_Q \min_\pi \{L(\pi, Q)\} = \min_Q \{\mathcal{F}(Q) = \mathbb{E}_{\rho^{\text{UN}}}[r^Q(s, a)] - (1 - \gamma)\mathbb{E}_{s_0}[V^Q(s_0)] + \psi(r^Q)\}$$

530 where $V^Q(s) = \beta \log(\sum_a \mu(a|s) \exp(Q(s, a)/\beta))$ and $r^Q(s, a) = Q(s, a) -$
531 $\gamma \mathbb{E}_{s' \sim P(\cdot|s, a)} V^Q(s')$.

532 *Proof.* We first express the second and third terms of the objective function $L(\pi, Q)$ in 5 as:

$$\mathbb{E}_{\rho_\pi}[\mathcal{T}^\pi[Q](s, a)] + H(\pi) = \mathbb{E}_{\rho_\pi}[Q(s, a) - \gamma \mathbb{E}_{s'}[V^\pi(s')]] - \beta \mathbb{E}_{\rho_\pi}[\log \frac{\pi(s, a)}{\mu(a|s)}]$$

533

$$= \mathbb{E}_{\rho_\pi}[Q(s, a) - \beta \log \frac{\pi(s, a)}{\mu(a|s)} - \gamma \mathbb{E}_{s'}[V^\pi(s')]] = \mathbb{E}_{\rho_\pi}[V(s) - \gamma \mathbb{E}_{s' \sim P(\cdot|s, a)}[V^\pi(s')]]$$

534

$$= (1 - \gamma)\mathbb{E}_{s_0 \sim P_0}[V^\pi(s_0)].$$

535 Thus, the objective function becomes:

$$L(\pi, Q) = \mathbb{E}_{\rho^{\text{UN}}}[Q(s, a) - \gamma \mathbb{E}_{s'}[V^\pi(s')]] - (1 - \gamma)\mathbb{E}_{s_0 \sim P_0}[V^\pi(s_0)] + \sum_{s, a} \psi(Q(s, a) - \gamma \mathbb{E}_{s'}[V^\pi(s')]).$$

536 We now observe that $V^\pi(s) = \mathbb{E}_{a \sim \pi(a|s)}[Q(s, a) - \beta \log \frac{\pi(a|s)}{\mu(a|s)}]$ is concave in π . Therefore, both
537 terms $\mathbb{E}_{\rho^{\text{UN}}}[Q(s, a) - \gamma \mathbb{E}_{s'}[V^\pi(s')]]$ and $-(1 - \gamma)\mathbb{E}_{s_0 \sim P_0}[V^\pi(s_0)]$ are convex in π . Additionally,
538 since $\psi(t)$ is convex and non-increasing in t , and $Q(s, a) - \gamma \mathbb{E}_{s'}[V^\pi(s')]$ is convex in π , each
539 function $\psi(Q(s, a) - \gamma \mathbb{E}_{s'}[V^\pi(s')])$ is convex in π . Thus, combining all terms, we conclude that
540 $L(\pi, Q)$ is convex in π .

541 Furthermore, each term $Q(s, a) - \gamma \mathbb{E}_{s'}[V^\pi(s')]$, $-(1 - \gamma)\mathbb{E}_{s_0 \sim P_0}[V^\pi(s_0)]$, and $\psi(Q(s, a) - \gamma \mathbb{E}_{s'})$
542 strictly decreases in V^π , implying that the minimization of $L(\pi, Q)$ over π is achieved when $V^\pi(s)$

is maximized for all s . Since $V^\pi(s)$ is strictly concave in π , maximizing $V^\pi(s)$ over π has a unique optimal solution:

$$\pi^Q(a|s) = \frac{\exp(Q(s, a)/\beta)}{\sum_a \exp(Q(s, a)/\beta)}.$$

This validates part (i) of the theorem.

For part (ii), we observe that the problem $\max_\pi V^\pi(s)$ has the optimal solution π^Q as shown above, and the optimal value is:

$$\begin{aligned} \max_\pi V^\pi(s) &= \max_\pi \left\{ \sum_a \pi(a|s) Q(s, a) - \beta \pi(a|s) \log \frac{\pi(a|s)}{\mu(a|s)} \right\} \\ &= \beta \log \left(\sum_a \mu(a|s) \exp(Q(s, a)/\beta) \right) \stackrel{\text{def}}{=} V^Q(s). \end{aligned}$$

This directly leads to:

$$\min_Q \min_\pi \{L(\pi, Q)\} = \min_Q \{F(Q) = \mathbb{E}_{\rho^{\text{UN}}} [r^Q(s, a)] - (1 - \gamma) \mathbb{E}_{s_0} [V^Q(s_0)] + \psi(r^Q)\},$$

as required. □

B.2 Proof of Proposition 4.2

Proposition: *The learning objective function $F(Q)$ is not convex in Q , even without any regularizer.*

Proof. We rewrite the function $F(Q)$ (without the regularizer $\psi(\cdot)$):

$$F(Q) = \mathbb{E}_{\rho^{\text{UN}}} [Q(s, a) - \gamma \mathbb{E}_{s'} [V^Q(s')]] - (1 - \gamma) \mathbb{E}_{s_0} [V^Q(s_0)],$$

where $V^Q(s) = \beta \log (\sum_a \mu(a|s) e^{Q(s, a)/\beta})$ is a log-sum-exp function, which is convex in Q . Since $V^Q(s)$ is convex in Q , all the components associated with $V^Q(s)$ in $F(Q)$ are concave in Q . This directly implies the non-convexity of $F(Q)$.

When the regularizer is included, it can be observed that if $\psi(\cdot)$ is convex in Q , then $\psi(r^Q(s, a))$ is also convex in Q . As a result, the objective function $F(Q)$ takes the form of a difference-of-convex program. □

B.3 Proof of Proposition 4.3

Proposition: *Under any convex regularizer ψ , both $\tilde{F}(Q|V)$ and $J(V|Q)$ are convex in Q and V , respectively.*

Proof. We first write the loss function for the Q -updates as:

$$\tilde{F}(Q|V) = \mathbb{E}_{\rho^{\text{UN}}} [Q(s, a) - \gamma \mathbb{E}_{s'} [V(s')]] - (1 - \gamma) \mathbb{E}_{s_0} [V(s_0)] + \psi(Q(s, a) - \gamma \mathbb{E}_{s'} [V(s')]),$$

where:

- The first term is linear in Q ,
- The second term is a constant (since the V -function is fixed when updating the Q -function), and
- The last term involves a convex function of Q , making it convex.

Thus, $\tilde{F}(Q|V)$ is convex in Q .

For the convexity of $J(V|Q)$, we first recall its formulation:

$$J(V|Q) = \mathbb{E}_{(s, a) \sim \mu} \left[e^{\frac{Q(s, a) - V(s)}{\beta}} - \left(\frac{Q(s, a) - V(s)}{\beta} \right) - 1 \right].$$

The convexity of $J(V|Q)$ can be observed as follows:

- The first term is an exponential function of V , which is convex in V ,
- The remaining terms are either linear in V or constants.

Therefore, $J(V|Q)$ is convex in V . \square

C Connection between UNIQ and Statistical Distance Maximization

In this section, we connect the new MaxEnt objective function of UNIQ with the concept of statistical distance. Specifically, we show that solving the new MaxEnt objective $\min_r \min_\pi L(\pi, r)$ in (4) is equivalent to maximizing a statistical distance between the occupancy measures of the learning policy and the undesirable policy. In particular, if the feasible set of the reward function is unrestricted, the following equality holds:

$$\min_r \min_\pi \{L(\pi, r)\} = -\max_\pi \{d_\psi(\rho^\pi, \rho^{\text{UN}}) - H(\pi)\}, \quad (9)$$

where $d_\psi(\rho^\pi, \rho^{\text{UN}}) = \psi^*(\rho^\pi - \rho^{\text{UN}})$, and ψ^* is the convex conjugate of the convex function ψ , i.e.,

$$\psi^*(t) = \sup_z (tz - \psi(z)).$$

To verify (9), we write the objective function as:

$$L(\pi, r) = \sum_{s,a} r(s, a) (\rho^{\text{UN}}(s, a) - \rho^\pi(s, a)) + \psi(r) - H(\pi).$$

Thus, the minimization of $L(\pi, r)$ over r can be expressed as:

$$\begin{aligned} \min_r L(\pi, r) &= H(\pi) + \min_r \left\{ \sum_{s,a} r(s, a) (\rho^{\text{UN}}(s, a) - \rho^\pi(s, a)) + \psi(r) \right\} \\ &= H(\pi) - \max_r \left\{ \sum_{s,a} r(s, a) (\rho^\pi(s, a) - \rho^{\text{UN}}(s, a)) - \psi(r) \right\}. \end{aligned}$$

Since the feasible set of the reward function r is unrestricted, we have:

$$\max_r \{ \langle r, \rho^\pi - \rho^{\text{UN}} \rangle - \psi(r) \} = \psi^*(\rho^\pi - \rho^{\text{UN}}).$$

This allows us to rewrite the training problem as:

$$\begin{aligned} \min_r \min_\pi \{L(\pi, r)\} &= \min_\pi \{H(\pi) - \psi^*(\rho^\pi - \rho^{\text{UN}})\} \\ &= -\max_\pi \{ \psi^*(\rho^\pi - \rho^{\text{UN}}) - H(\pi) \} \\ &= -\max_\pi \{ d_\psi(\rho^\pi, \rho^{\text{UN}}) - H(\pi) \}. \end{aligned}$$

We thus obtain the desired equation in (9), proving the equivalence.

D Experimental settings

D.1 Environmental Details

D.1.1 Mujoco

The MuJoCo locomotion benchmark focuses on a set of environments designed to evaluate the ability of reinforcement learning algorithms to control agents with complex, multi-joint dynamics. These tasks involve training agents to move efficiently and stably in simulated environments. The illustrations are shown in Figure 5.

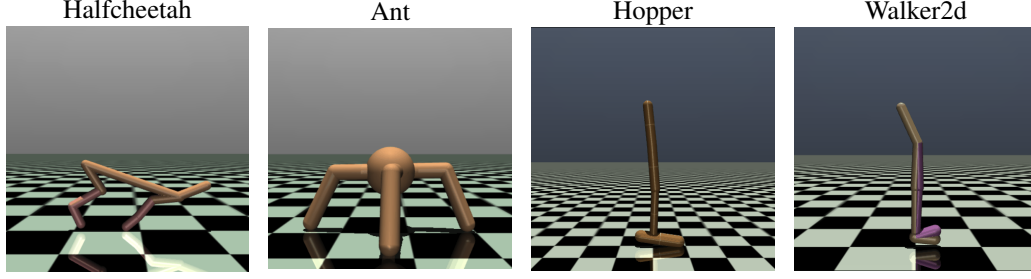


Figure 5: Mujoco environments.

D.1.2 Safe-Gym

Safe-Gym is a collection of reinforcement learning environments designed with a focus on safety, built on top of the OpenAI Gym framework. It introduces constraints that simulate safety-critical scenarios commonly encountered in real-world applications. In Safe-Gym, agents are rewarded for completing task-specific objectives but face penalties for violating safety constraints, such as surpassing speed limits, colliding with obstacles, or entering restricted zones. These constraints enable Safe-Gym to replicate environments where safety is paramount, including robotic navigation in congested areas, autonomous vehicle control, and industrial automation. The environment features two types of agents: Point (an easy agent) and Car (a more challenging agent), as well as two types of tasks: Goal (easy) and Button (hard). Additionally, the environment dynamics change with each new episode, introducing variability and increasing the complexity of the tasks. Illustrations of these tasks are shown in Figure 6.

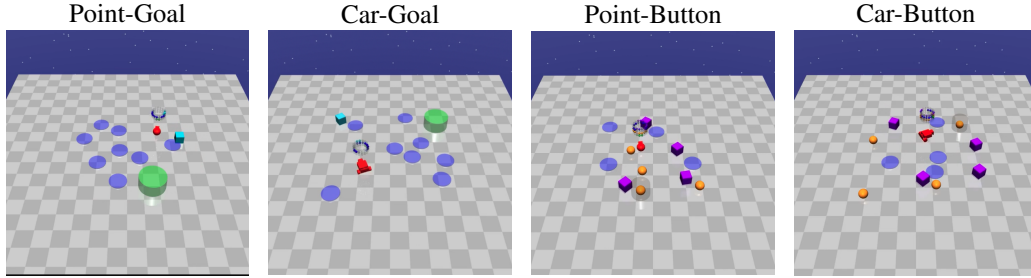


Figure 6: Safety-gym environments.

D.1.3 Mujoco-velocity

From original Mujoco, Mujoco-Velocity is a specialized environment within the Mujoco physics simulation suite, focusing on controlling the velocity of two specific agents: Cheetah and Ant. These agents must complete locomotion tasks while adhering to safety constraints on their speed. The goal is to balance task performance with maintaining safe velocity limits. For instance, Cheetah must run as fast as possible while staying within predefined speed bounds to avoid penalties, mimicking real-world scenarios where exceeding speed limits can cause system failure or unsafe operations. Similarly, Ant must navigate through its environment without violating velocity constraints, ensuring stability and safety.

D.2 Dataset generation details

In this paper, we tackle the problem of **Offline Learning with Undesirable Demonstrations** by utilizing two datasets:

- Unlabeled dataset \mathcal{D}^{MIX} : A large dataset comprising both desired and undesired demonstrations, reflecting real-world data (e.g., chat conversations, driving behaviors, treatment decisions, etc.).

- Undesired dataset \mathcal{D}^{UN} : A smaller, accurate dataset containing demonstrations that exhibit behaviors we aim to avoid.

D.2.1 Mujoco dataset

We use the official D4RL dataset with three different performance: random, medium (sub-optimal), and expert. We then combine them to collect the dataset as follow:

- Unlabeled dataset \mathcal{D}^{MIX} : We mix all three dataset into a large unlabeled dataset with a specific ratio.
- Undesired dataset \mathcal{D}^{UN} : We mix random and medium dataset with the same ratio (1 : 1).

D.2.2 Safety-gym and Mujoco-velocity dataset

We simulate this scenario using the Safety-gym and Mujoco-velocity environments. First, we train both unconstrained and constrained policies using PPO [41] and PPO-lagrangian [38]. We then collect the training datasets as follows:

- Unlabeled dataset \mathcal{D}^{MIX} : We roll out the constrained and unconstrained policies, mixing them at ratios of (1 : 4).
- Undesired dataset \mathcal{D}^{UN} : We roll out the unconstrained policy, gathering trajectories that violate the constraint.

The quality of Safety-gym dataset are shown in Table 2 while details information of Mujoco-velocity datasets are shown in Table 3.

	Point-Goal	Car-Goal	Point-Button	Car-Button
Mean Unconstrained return	26.8 ± 1.1	35.2 ± 2.1	21.49 ± 4.87	17.82 ± 5.57
Mean Constrained return	25.3 ± 2.1	25.7 ± 6.6	13.71 ± 6.44	14.99 ± 8.45
Mean Unconstrained cost	57.8 ± 38.9	61.9 ± 48.1	138.9 ± 77.3	236.5 ± 115.9
Mean Constrained cost	22.3 ± 28.0	21.0 ± 31.0	27.4 ± 34.1	113.5 ± 33.2
Cost Threshold	25.0	25.0	25.0	100.0

Table 2: Safety-Gym expert policies performance.

	Cheetah	Ant
Mean Unconstrained return	3027.4 ± 400.6	2972.2 ± 1020.0
Mean Constrained return	2751.2 ± 11.6	2830.0 ± 145.5
Mean Unconstrained cost	626.7 ± 95.0	624.0 ± 231.0
Mean Constrained cost	14.1 ± 4.5	18.7 ± 4.4
Cost Threshold	25.0	25.0

Table 3: Mujoco-velocity expert policies performance.

D.3 Baseline implementation details

D.3.1 BC

We use the original BC objective:

$$\min_{\pi} -\mathbb{E}_{s,a \sim \mathcal{D}} \log \pi(a|s) \quad (10)$$

642 D.3.2 LS-IQ

We use the official implementation of LS-IQ in [this link](#). We modify the objective to fit with our scenario when undesirable dataset available:

$$\min_Q \mathbb{E}_{s,a \sim \mathcal{D}^{\text{UN}}} [(Q(s,a) - Q_{\min})^2] + \mathbb{E}_{s,a,s' \sim \mathcal{D}^{\text{MIX}}} [(Q(s,a) - r_{\max} - \gamma V(s'))^2],$$

643 where $V(s) = Q(s, \pi(a|s)) + H(\pi)$; $Q_{\min} = r_{\min}/(1 - \gamma)$; and r_{\min} and r_{\max} are predefined.

644 D.3.3 IPL

We use the official implementation of IPL [17] from [this link](#). The only difference between our setting and IPL is the pairwise dataset. We create pairwise comparisons from the unlabeled dataset and the undesired dataset and train the Q-function with the following new loss function:

$$P_{Q^\pi}[\sigma^{\text{MIX}} > \sigma^{\text{UN}}] = \frac{\exp \sum_t (\mathcal{T}^\pi Q)(s_t^{\text{MIX}}, a_t^{\text{MIX}})}{\exp \sum_t (\mathcal{T}^\pi Q)(s_t^{\text{MIX}}, a_t^{\text{MIX}}) + \exp \sum_t (\mathcal{T}^\pi Q)(s_t^{\text{UN}}, a_t^{\text{UN}})},$$

where:

$$(\mathcal{T}^\pi Q)(s, a) = Q(s, a) - \gamma \mathbb{E}_{s'} [V^\pi(s')].$$

645 D.3.4 DWBC

We use the official implementation of DWBC [45] from [this link](#). We modify the algorithm to train a discriminator that assigns 0 to the undesired dataset \mathcal{D}^{UN} and 1 to the unlabeled dataset \mathcal{D}^{MIX} while keeping the Positive Unlabeled learning technique from the paper:

$$\begin{aligned} \min_{\theta} \eta \mathbb{E}_{(s,a) \sim \mathcal{D}^{\text{MIX}}} [-\log d_{\theta}(s, a, \log \pi)] \\ + \mathbb{E}_{(s,a) \sim \mathcal{D}^{\text{UN}}} [-\log (1 - d_{\theta}(s, a, \log \pi))] \\ - \eta \mathbb{E}_{(s,a) \sim \mathcal{D}^{\text{MIX}}} [-\log (1 - d_{\theta}(s, a, \log \pi))]. \end{aligned}$$

646 D.3.5 SafeDICE

647 We use the official implementation of SafeDICE [20] from [this link](#). As the algorithm has been
648 designed to solve this problem, we do not make any further modifications.

649 D.4 Hyper-parameter selection

650 For fair comparison, we keep the same basic hyper-parameters across all the baselines which are
651 detailed as follow for Mujoco, Safety-gym and Mujoco-velocity tasks as Table 4:

HYPER PARAMETER	MUJOCO	SAFETY-GYM	MUJOCO-VELOCITY
ACTOR NETWORK	256x3	256x3	256x3
CRITIC NETWORK	256x3	256x3	256x3
VALUE NETWORK	256x3	256x3	256x3
TRAINING STEP	1,000,000	1,000,000	1,000,000
GAMMA	0.99	0.99	0.99
LR ACTOR	0.0003	0.0001	0.0001
LR CRITIC	{0.0001, 0.0003}	0.0003	0.0003
LR VALUE	0.0003	0.0003	0.0003
LR DISCRIMINATOR	0.0001	0.0001	0.0001
BATCH SIZE	256	256	256
SOFT UPDATE CRITIC FACTOR	[0.001, 0.01]	0.005	0.005

Table 4: Hyper parameters.

In UNIQ, the α and β parameters are selected in range $[0.5, 10.0]$. Lastly, we apply state normalization for Mujoco and Mujoco-velocity datasets as follow:

$$s_{\text{normalized}} = \frac{s - \mu}{\sigma}$$

Where:

$$\mu = \frac{1}{|\mathcal{D}|} \sum_{s' \in \mathcal{D}} s'$$
$$\sigma = \sqrt{\frac{1}{|\mathcal{D}|} \sum_{s' \in \mathcal{D}} (s' - \mu)^2}$$

652 **D.5 Computational Resource**

653 Our experiments were carried out on a GPU cluster equipped with 8 NVIDIA RTX 3090 GPUs. Each
654 experimental setup was run with five distinct training seeds in parallel, sharing a single GPU, eight
655 CPU cores, and 64 GB of RAM. Within this shared configuration, completing one million training
656 steps across all five seeds took approximately 30 minutes. The software environment was built using
657 JAX version 0.4.28, with support for CUDA 12—specifically, CUDA version 12.3.2 and cuDNN
658 version 8.9.7.29.

659 It is important to note that in the Safety-Gym domain, evaluation is relatively slow due to the use of
660 50 different environment random seeds for each (we have 100 evaluations in total). While the training
661 phase takes around 30 minutes, the extensive evaluation increases the total runtime per experiment to
662 approximately 2 hours.

663 E Additional Experiments

664 E.1 Performance with Different Sizes of undesirable Dataset

665 In this section, we provide a complete experiment to evaluate the contribution of the undesirable
 666 dataset to our algorithm. We test our method with an increasing size of the undesirable dataset,
 667 $\mathcal{D}^{\text{UN}} = \{5, 10, 20, 50, 100, 200\}$ for the Mujoco tasks and $\mathcal{D}^{\text{UN}} = \{25, 50, 100, 200, 300, 500\}$. The
 668 full experiment results are shown in Figure 7 (Mujoco) and Figure 8 (Safety-Gym).

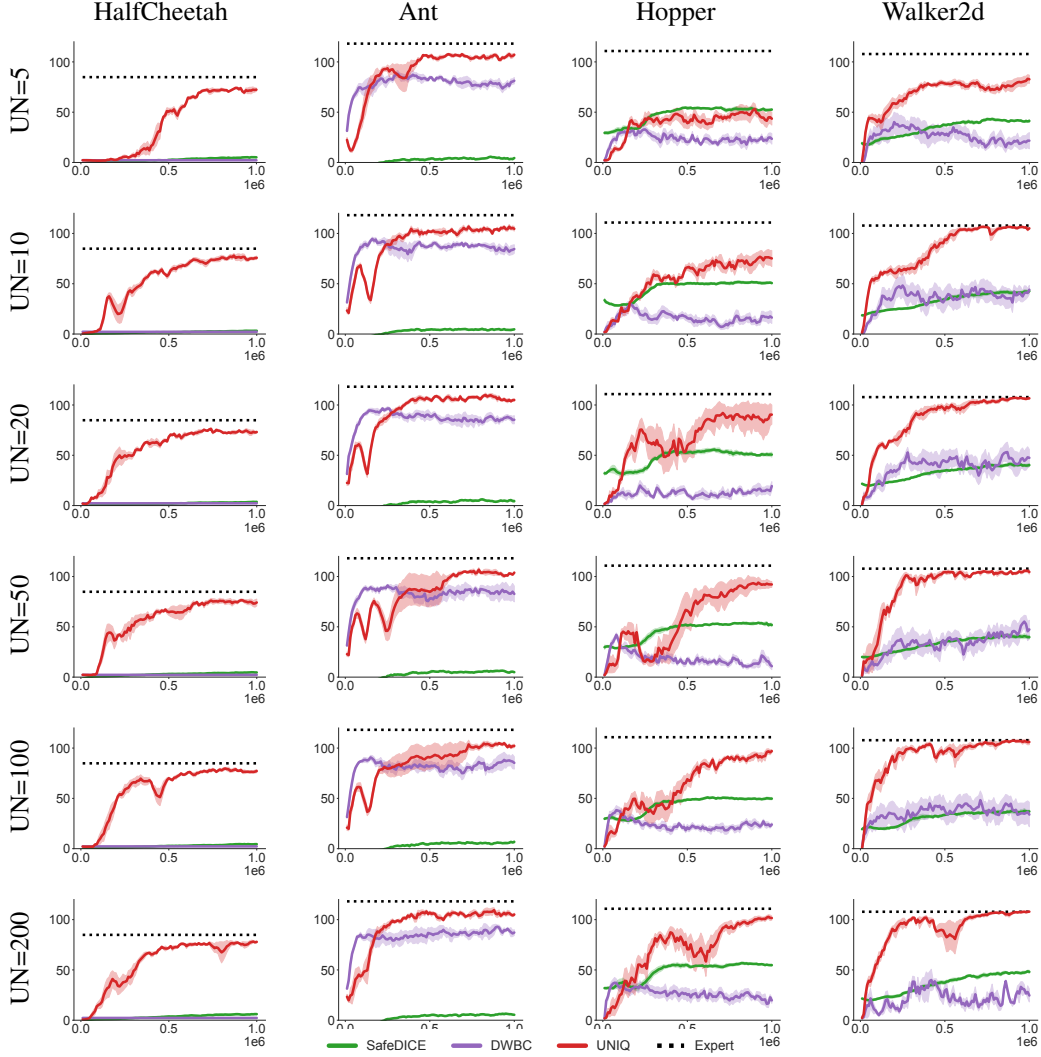


Figure 7: Comparison with different size of Undesirable dataset in Mujoco environments.

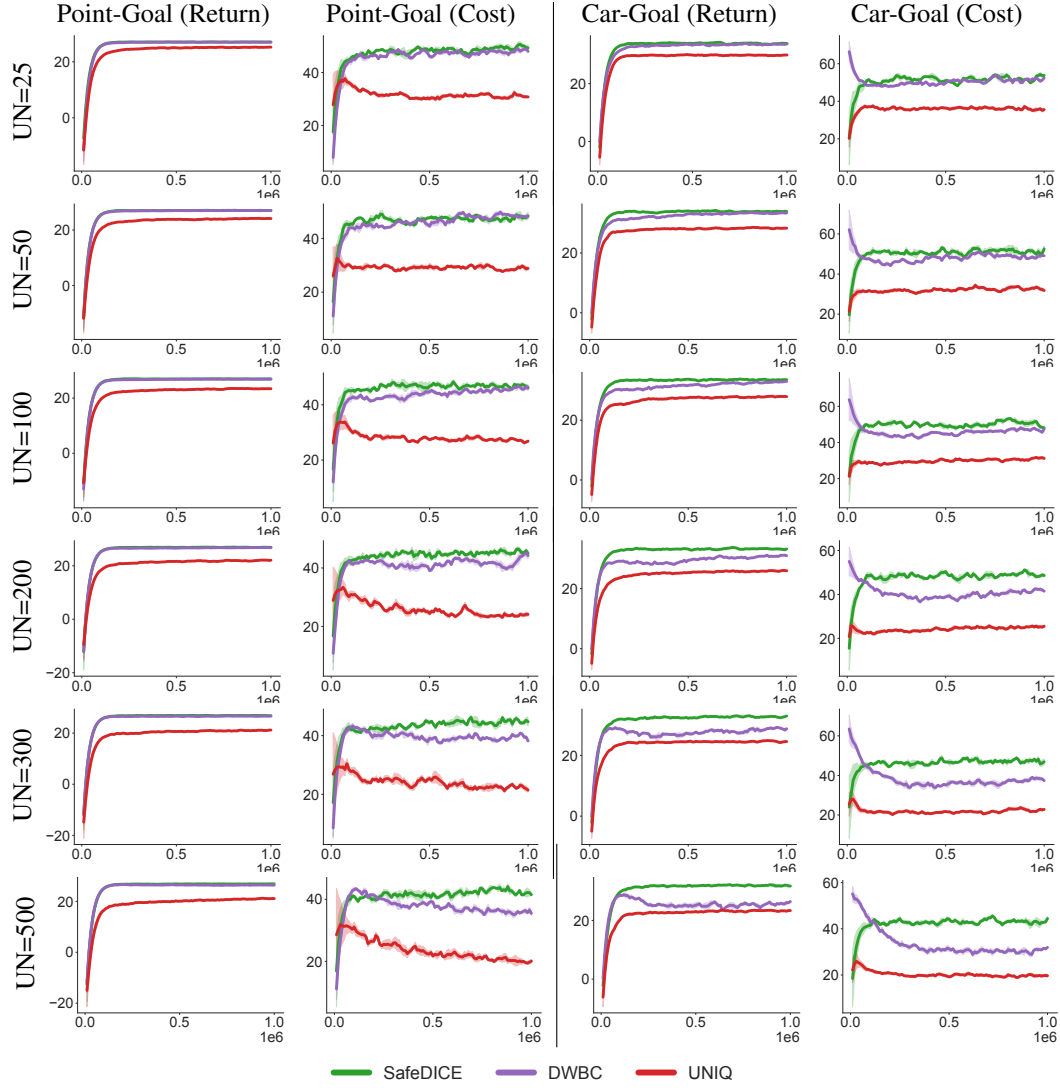


Figure 8: Comparison with different size of Undesirable dataset in Safety-gym environments.

669 **E.2 Performance with Different Number of Undesirable Demonstrations in the Unlabeled**
670 **Dataset**

671 We further change the number of undesirable demonstrations in the unlabeled dataset \mathcal{D}^{Mix} to examine
672 how it impacts the performance of the algorithm. The experimental results are shown in Figure 9.
673 The results show that when the quality of the dataset decreases significantly, the performance of all
674 algorithms worsens. However, UNIQ still achieves the highest performance across all environments.

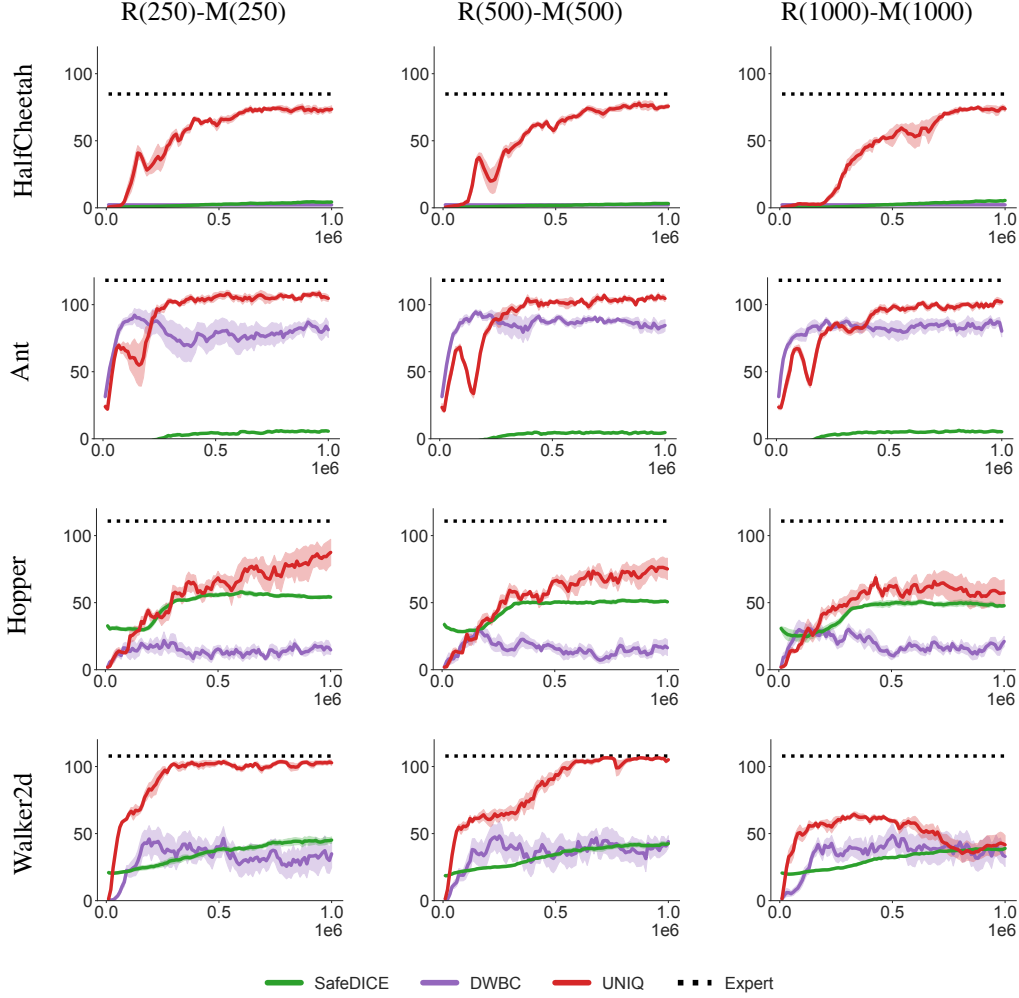


Figure 9: Comparison with different size of Undesirable dataset in Mujoco environments.

675 E.3 Performance with Different Number of Desirable Demonstrations in the Unlabeled 676 Dataset

677 We also test how changing the amount of desirable data in the unlabeled dataset affects the perfor-
678 mance. The experimental results are shown in Figure 10 (Mujoco) and Figure 11 (Safety-Gym). We
679 observe an increasing trend in performance as the size of the desirable data increases (higher returns
680 in Mujoco and lower costs in Safety-Gym).

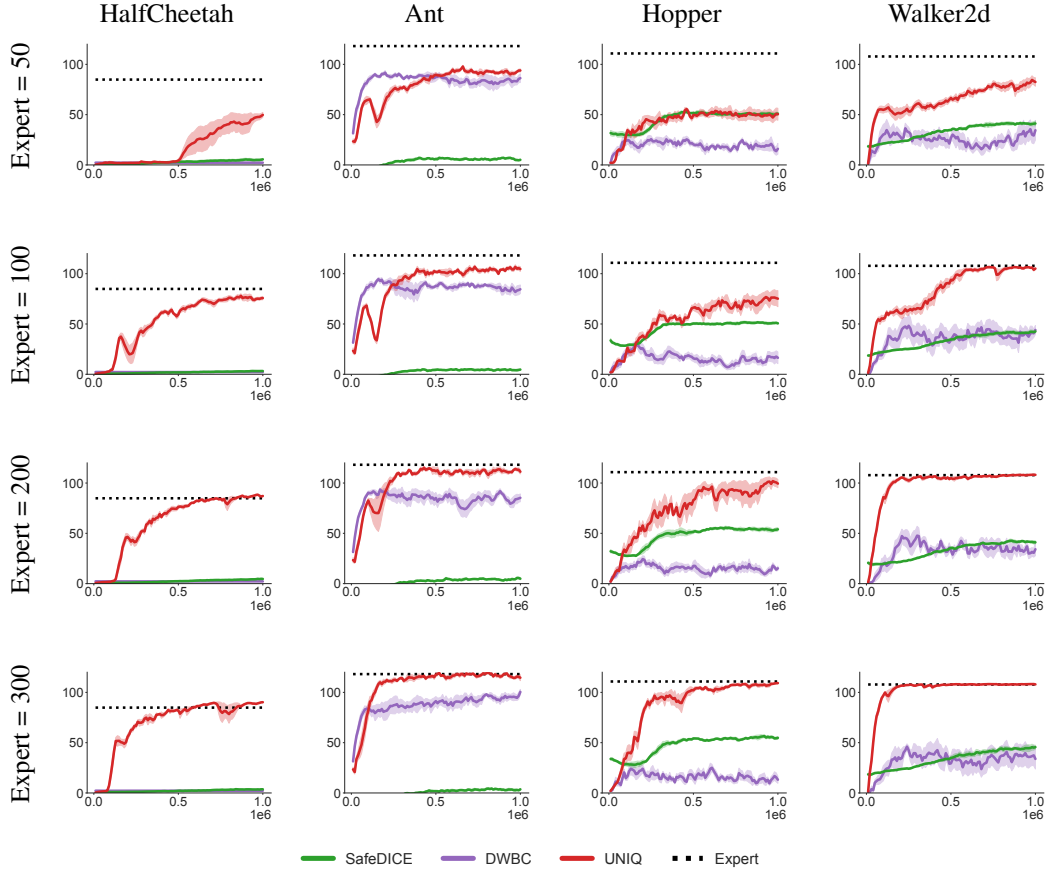


Figure 10: Comparison with different size of Undesirable dataset in Mujoco environments.

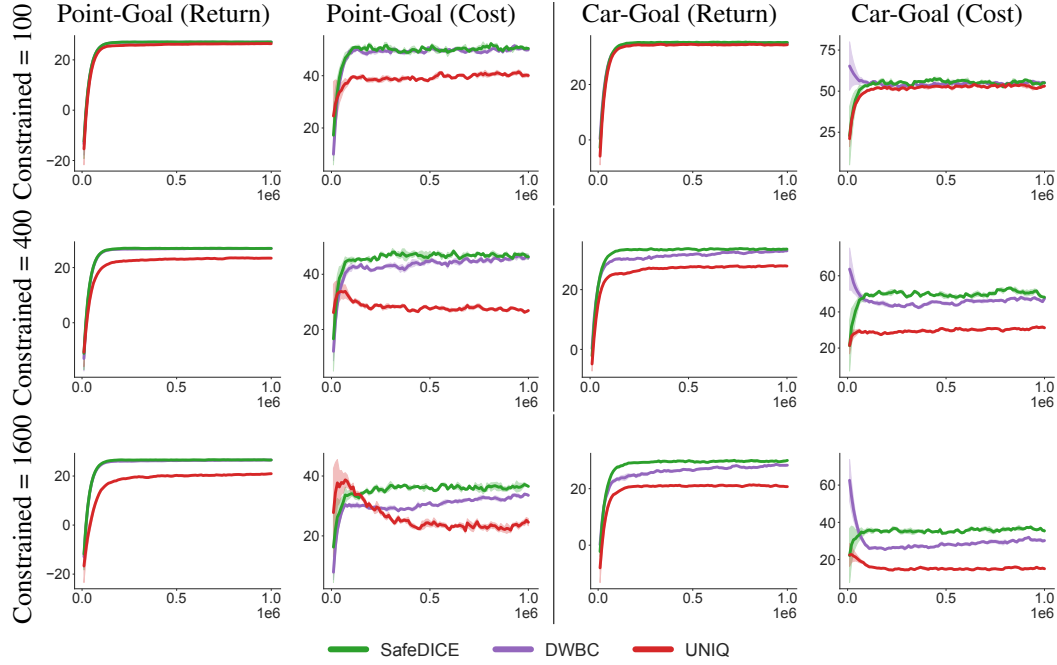


Figure 11: Comparison with different size of Undesirable dataset in Safety-gym environments.

681 E.4 Direct Policy Extraction from Q-functions

682 As our actor update objective uses Weighted Behavioral Cloning, a question arises: what if we directly
 683 extract the actor from the Q-function [12, 3]? We conduct experiments to evaluate the performance of
 684 UNIQ when the actor is learned directly from the Q-function. The results are presented in Figure 12
 685 and Figure 13..

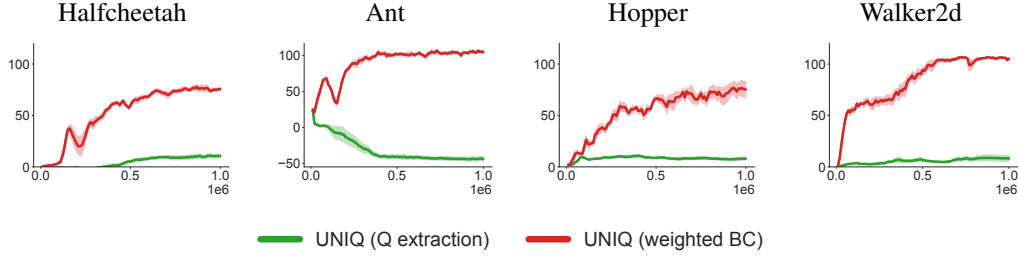


Figure 12: compare UNIQ BC update with Q extraction update in D4RL

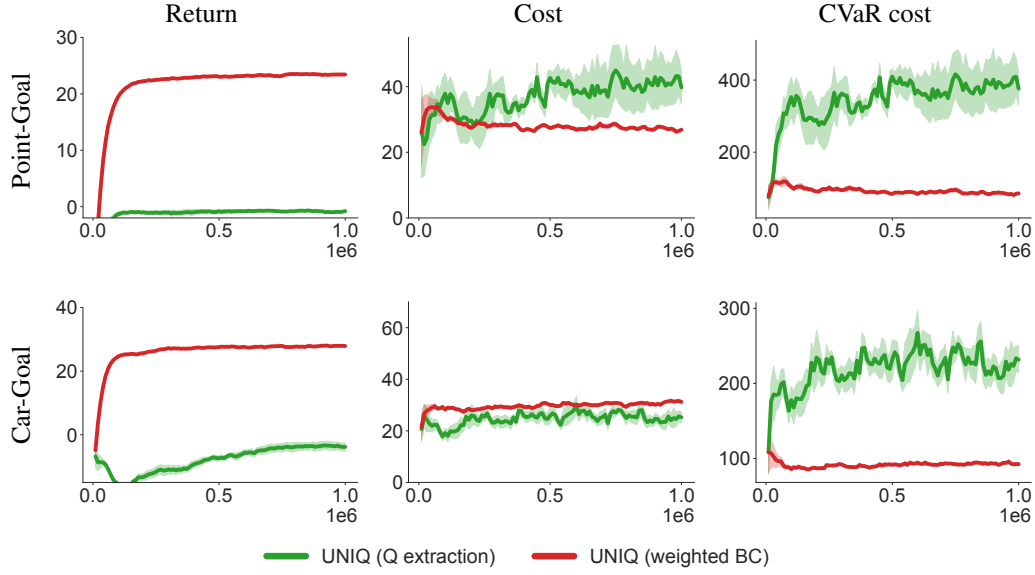


Figure 13: compare UNIQ BC update with Q extraction update in Safetygym

686 E.5 BC without Unlabeled Dataset

687 As UNIQ uses an unlabeled dataset to learn by avoiding undesirable behaviors and learning from the
 688 remaining parts of the dataset, an interesting question arises: what if we only avoid the undesirable
 689 dataset (no unlabeled dataset is given)? To explore this, we train a Behavioral Cloning (BC) agent
 690 that, instead of following the dataset, explicitly avoids the actions from the undesirable dataset by
 691 minimizing the probability of those assigned actions, denoted as BC-UN. The detailed results are
 692 shown in Figure 14 (Mujoco) and Figure 15 (Safety-Gym).

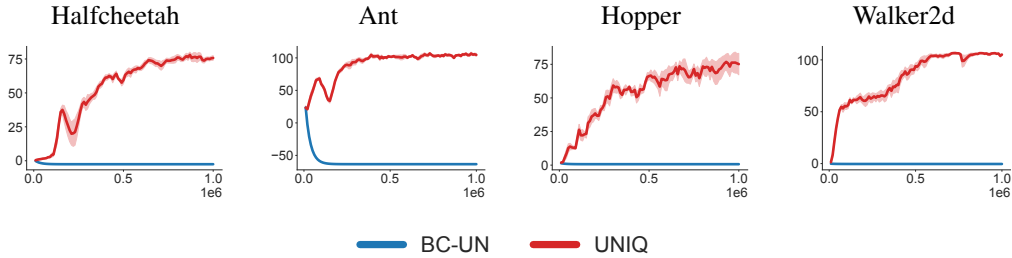


Figure 14: BC only avoid undesirable data in D4RL

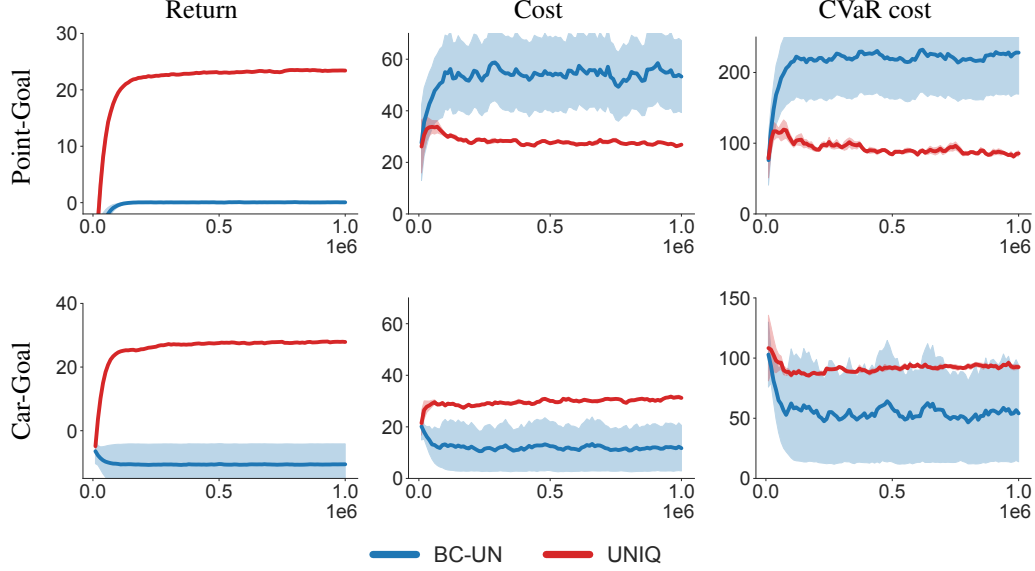


Figure 15: BC only avoid undesirable data in Safetygym

E.6 ablation study with different τ

Task	$\tau = 1$	Non-converged τ	Converged τ
HalfCheetah	18.5 ± 5.6	64.9 ± 5.2	75.7 ± 6.8
Ant	88.5 ± 8.2	95.2 ± 12.4	104.4 ± 10.5
Hopper	57.4 ± 12.4	58.0 ± 14.6	73.5 ± 20.6
Walker2d	12.8 ± 7.7	72.0 ± 6.1	105.9 ± 4.0

Table 5: Ablation study with different τ : We maintain the same dataset construction as in the main experiment.

To assess the effect of occupancy ratio estimation on our algorithm, we conducted additional ablation studies using three configurations: (i) a uniform occupancy ratio ($\tau = 1$), (ii) a non-converged estimation of τ , and (iii) a properly converged τ . The results of this ablation study (for the unconstrained RL setting) are shown in Table 5.

E.7 Additional comparison with modified baselines for avoiding undesirable

Task	BC-remove-bad	IS-WBC	ILID	UNIQ
HalfCheetah	2.3 ± 0.0	2.2 ± 0.0	2.1 ± 0.0	75.7 ± 6.8
Ant	36.2 ± 19.6	73.4 ± 12.3	106.44 ± 4.3	104.4 ± 10.5
Hopper	19.7 ± 18.9	7.6 ± 5.0	5.9 ± 6.5	73.5 ± 20.6
Walker2d	37.8 ± 33.3	18.8 ± 19.3	69.6 ± 13.0	105.9 ± 4.0

Table 6: Additional comparison with BC-remove-bad, IS-WBC, and ILID: while keeping the dataset construction consistent with the main experiment.

Since our method is designed to learn from undesirable demonstrations, we evaluate it alongside several baseline algorithms that were originally developed for expert demonstrations but have been adapted for our setting:

- **BC-remove-bad:** We first train a classifier to detect and remove undesirable demonstrations from the unlabeled dataset. Behavior Cloning (BC) is then applied to the filtered data to learn the policy.
- **ISW-BC:** We modify the original discriminator objective to better suit the learning-from-undesirable-demonstrations framework. Specifically, we train the discriminator to differ-

entiate between undesirable and unlabeled demonstrations, rather than between expert and non-expert data as in the original method. This adaptation helps guide the policy away from undesirable behaviors without requiring expert data.

- **ILID**: We adapt the original ILID approach by removing all states in the unlabeled dataset that resemble those in the undesirable dataset. However, the original ILID also includes a regularization component that prevents the learned policy from deviating significantly from a BC-trained expert policy. Since expert demonstrations are unavailable in our setting, this regularization is excluded in our adaptation.

The comparative performance results are reported in Table 6. Overall, UNIQ yield the best performance in Mujoco tasks.

E.8 Performance with 4 category types (low-cost low-reward, low-cost high-reward, high-cost low-reward, high-cost high-reward) of dataset in Safety-gym

		DWBC	SafeDICE	UNIQ	Expert
Point-Goal	Return	20.1 ± 1.1	19.9 ± 0.4	24.3 ± 0.8	25.9 ± 0.2
	Cost	47.0 ± 4.5	41.7 ± 5.0	26.0 ± 4.4	26.0 ± 2.6
Car-Goal	Return	20.9 ± 1.5	22.0 ± 3.6	22.5 ± 2.2	26.2 ± 0.7
	Cost	30.3 ± 3.3	30.5 ± 7.7	19.9 ± 2.8	23.6 ± 2.8

Table 7: Performance of 4 different category of performance types on Point-Goal and Car-Goal Tasks.

The objective of this experiment is to evaluate whether the algorithm can successfully avoid all types of undesirable demonstrations in the Safety-Gym environment. While our main experiments (Table 1) primarily assess safety performance with minimal return degradation, this experiment focuses on the algorithm’s ability to comprehensively reject undesirable trajectories.

To achieve this, we construct a dataset based on four categories of Safety-Gym performance: high-return low-cost (H-L), high-return high-cost (H-H), low-return low-cost (L-L), and low-return high-cost (L-H). The dataset is divided into two subsets: the undesirable dataset \mathcal{D}^{UN} , which consists of 50 trajectories each from the H-H, L-L, and L-H categories (totaling 150 trajectories); and the unlabeled dataset \mathcal{D}^{MIX} , which contains 500 trajectories from each category, resulting in 2,000 trajectories in total.

The experimental results are reported in Table 7. may initially appear counterintuitive—SafeDICE and UNIQ exhibit reduced cost despite the more challenging data. However, this outcome is likely due to a trade-off inherent in the policy behavior: lower rewards are often correlated with lower costs. In particular, policies that prioritize minimizing cost may opt for conservative behaviors, such as remaining stationary, which avoids hazards but also leads to diminished task progress and, consequently, lower returns.

735 E.9 Comparisons with CVaR 10% Cost for Safety-gym Tasks

736 We also report the CVaR 10% cost for Safety-gym tasks, supporting the result of the Table 1 with
 737 CVaR is the mean of 10% highest in cost trajectories during the evaluation process. The full results
 738 are shown in Table 8.

		BC-mix	LS-IQ	IPL	DWBC	SafeDICE	UNIQ
Point-Goal	Return	27.1±0.1	−6.8 ± 4.4	26.9±0.1	26.9±0.1	27.0±0.1	23.4±0.4
	Cost	48.8±2.9	18.0 ± 29.3	52.7±3.4	45.8±3.4	46.8±3.1	27.1±3.0
	CVaR	115.4±7.7	133.3 ± 240.7	117.9±8.2	110.4±7.8	111.0±7.6	85.9± 18.9
Car-Goal	Return	34.1±0.5	−0.6 ± 2.5	34.7±0.3	32.8±0.7	33.5±0.7	27.9±0.8
	Cost	52.0±4.2	58.5 ± 41.7	54.4±3.7	47.4±3.8	50.5±4.0	31.0±2.8
	CVaR	132.8±10.6	311.1 ± 208.9	134.9±8.7	123.2±10.6	128.5±10.2	93.3 ± 8.4
Point-Button	Return	17.6 ± 0.7	−13.3 ± 6.7	16.9±0.9	17.2 ± 0.9	15.1 ± 0.5	12.6 ± 1.4
	Cost	120.2 ± 10.0	11.0 ± 10.2	124.8±11.3	123.5 ± 14.4	91.0 ± 6.4	23.0 ± 4.7
	CVaR	311.0 ± 47.8	58.2 ± 89.1	307.2 ± 47.3	309.6 ± 63.7	229.2 ± 21.2	98.7 ± 33.1
Car-Button	Return	17.6 ± 0.7	−8.6 ± 4.8	17.2±0.8	17.1 ± 1.0	17.4 ± 0.6	12.7 ± 1.1
	Cost	241.6 ± 15.3	28.0 ± 27.0	257±12.6	249.2 ± 20.9	201.3 ± 10.8	148.6 ± 18.7
	CVaR	545.3 ± 45.6	195.9 ± 204.1	553.2 ± 35.7	550.5 ± 66.3	410.5 ± 34.3	449.8 ± 63.4

Table 8: Full comparison results in Return, Cost, and CVaR 10%.

739 E.10 Controlling Conservativeness in UNIQ

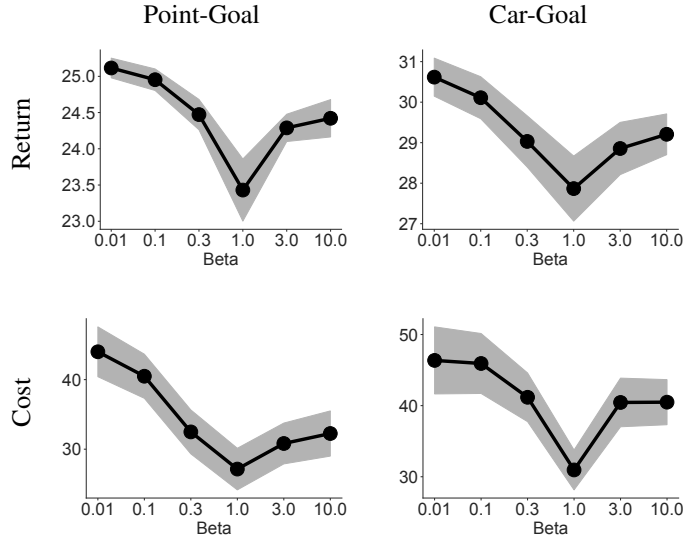


Figure 16: Comparison results of UNIQ with different α selections.

740 The main paper demonstrates that the policy returned by UNIQ achieves significantly lower costs
 741 (indicating safety) but, in some cases, also lower rewards compared to other imitation learning
 742 baselines. While this aligns well with our objective of learning safe policies by avoiding unsafe
 743 demonstrations, it also raises concerns about the algorithm’s conservativeness.

744 In this section, we show that the conservativeness of UNIQ can be effectively controlled by introducing
 745 a parameter to the Weighted BC formulation. Specifically, we adjust the conservativeness of the
 746 algorithm by adding a parameter β to the Weighted BC update:

$$\sum_{(s,a) \sim \mathcal{D}^{\text{Mix}}} \exp((Q_{w_q}(s,a) - V_{w_v}(s)) * \beta) \log \pi_{\theta}(a|s)$$

When $\beta = 1$, the Weighted BC theoretically returns the exact policy derived from Q-learning, as reported in the main paper. In contrast:

- As $\beta \rightarrow 0$, the Weighted BC returns a random policy.
- As $\beta \rightarrow \infty$, the resulting policy becomes deterministic, always selecting the best action with probability 1.

Thus, by varying β , we can deviate the outcome of the Weighted BC from the policy given by Q-learning, reducing the conservativeness of the learned policy.

To experimentally demonstrate this, we vary β and report the corresponding returns and costs on four MuJoCo environments. The results are presented in Figure 16 and Table 9, showing how different values of β impact the trade-off between safety and performance.

Figure 16 demonstrates that UNIQ achieves its safest (and most conservative) performance when $\beta = 1$. At this value, the policy prioritizes minimizing costs, making it the most risk-averse option. However, as β deviates from 1, both the cost and return increase. This indicates that the Weighted BC formulation produces less conservative policies that are less safe but capable of achieving higher rewards.

Table 9 provides a more detailed breakdown of the costs and returns for different values of β . The results show that UNIQ can effectively balance safety and performance: by adjusting β , it is possible to achieve a safer policy (i.e., lower cost) while maintaining competitive returns (compared to other baselines). This adaptability highlights the flexibility of UNIQ.

When safety is critical, setting $\beta = 1$ ensures the most conservative policy, aligning with the objective of avoiding unsafe demonstrations. On the other hand, by varying β , one can tune the trade-off to achieve policies that are less safe but yield higher rewards, making UNIQ suitable for a range of scenarios depending on the desired safety-performance balance. This versatility demonstrates its practicality across different applications with varying safety requirements.

		DWBC	SafeDICE	UNIQ (0.01)	UNIQ (0.1)	UNIQ (0.3)	UNIQ (1.0)	UNIQ (3.0)
Point-Goal	Return	26.9±0.1	27.0±0.1	25.1 ± 0.1	25.0 ± 0.1	24.5 ± 0.2	23.4±0.4	24.3 ± 0.2
	Cost	45.8±3.4	46.8±3.1	44.0 ± 3.6	40.5 ± 3.2	32.5 ± 3.2	27.1±3.0	30.8 ± 2.9
Car-Goal	Return	32.8±0.7	33.5±0.7	30.6 ± 0.5	30.1 ± 0.5	29.0 ± 0.6	27.9±0.8	28.9 ± 0.6
	Cost	47.4±3.8	50.5±4.0	46.4 ± 4.7	45.9 ± 4.2	41.2 ± 3.4	31.0±2.8	40.4 ± 3.4

Table 9: Comparison with different α

771 E.11 Full Numerical Experiment for Mujoco Velocity Tasks

772 We evaluate our method on two MuJoCo velocity tasks: Cheetah and Ant. we test the method
 773 with varying sizes of the undesired dataset, annotated as "env-UN= {1, 5, 10}" while the unlabeled
 774 dataset \mathcal{D}^{Mix} is combined from 1600 high-cost and 400 low-cost trajectories. The detailed results
 775 are summarized in Table 10 and learning curves are shown in Figure 17 and Figure 18. Overall,
 776 increasing the size of the undesired dataset helps SafeDICE and DWBC achieve higher performance,
 777 while UNIQ reaches its peak performance with just a single undesired trajectory.

		DWBC	SafeDICE	UNIQ
Cheetah-UN=10	Return	3135.6±127.4	2841.9±56.1	2662.0±33.1
	Cost	311.0±116.0	550.2±13.5	0.0±0.0
	CVaR	897.7±10.0	682.2±14.4	0.0±0.0
Cheetah-UN=5	Return	3430.9±107.5	2860.8±57.8	2661.2±29.7
	Cost	578.8±89.6	553.9±25.3	0.0±0.0
	CVaR	909.2±6.3	686.7±17.3	0.0±0.0
Cheetah-UN=1	Return	3720.7±39.2	2910.0±61.8	2755.3±23.8
	Cost	823.0±17.5	575.5±23.0	0.0±0.0
	CVaR	916.4±5.0	702.2±20.0	0.0±0.0
Ant-UN=10	Return	2225.0±759.3	2713.0±56.2	2850.5±177.5
	Cost	470.5±162.8	439.7±57.7	15.2±10.8
	CVaR	795.0±103.7	668.7±14.6	24.6±13.7
Ant-UN=5	Return	2210.0±655.7	2727.4±49.8	2838.2±177.9
	Cost	494.5±146.8	464.4±35.3	13.1±7.5
	CVaR	805.7±16.4	671.2±16.0	22.1±10.5
Ant-UN=1	Return	2259.4±653.8	2724.4±90.7	2841.4±214.9
	Cost	507.5±147.8	506.5±40.3	16.9±7.1
	CVaR	789.3±91.3	685.8±16.4	27.2±9.6

Table 10: Full comparison between UNIQ and other baselines in Mujoco-velocity domain. With decreasing of undesirable dataset size, the performance of DWBC and SafeDICE become worse. In contrast, UNIQ able to achieve highest performance with just a single undesirable trajectory.

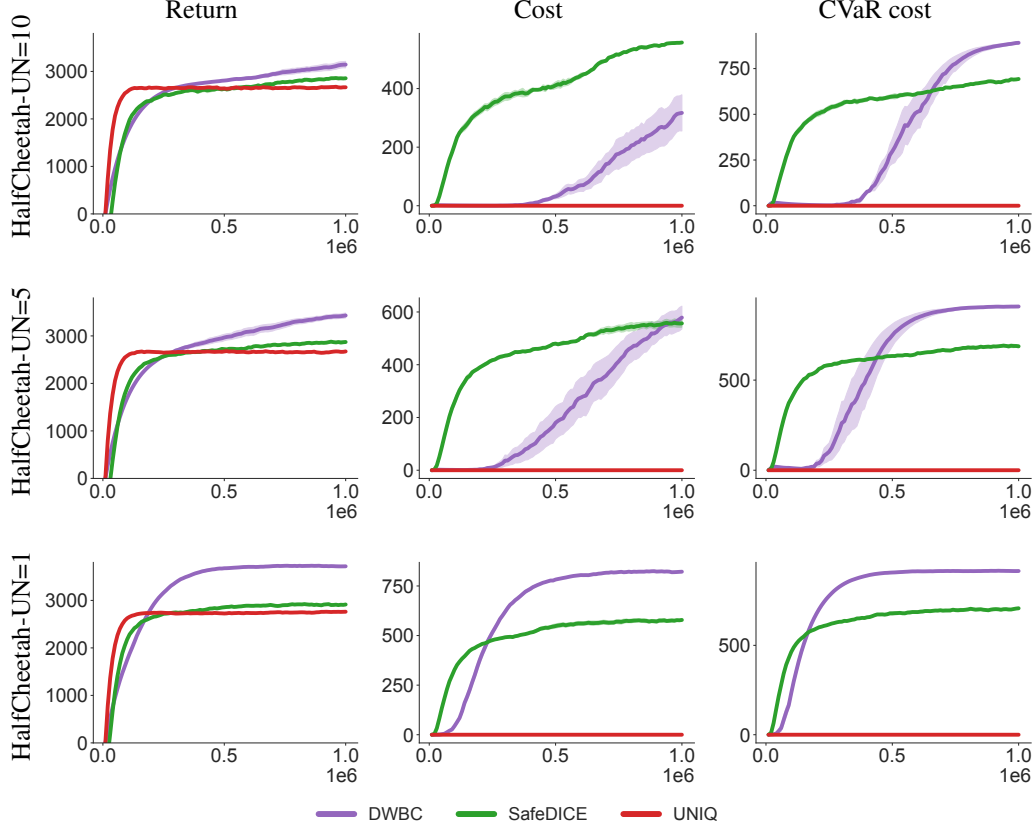


Figure 17: Cheetah task with unlabeled dataset(400-1600) and different undesired dataset.

E.12 Performance with the Dataset Employed in the SafeDICE Paper

As we are using a different dataset from the SafeDICE dataset, we also provide a comparison with the dataset from SafeDICE paper. The detailed performance of the expert dataset is shown in Table 11:

	Point-Goal	Point-Button
Mean non-preferred demonstrations cost	20.018	21.933
Mean preferred demonstrations return	19.911	8.286
Mean non-preferred demonstrations cost	107.977	166.099
Mean preferred demonstrations return	13.798	12.085

Table 11: SafeDICE dataset performance.

We mix 300 preferred demonstrations and 1200 non-preferred demonstrations for the unlabeled dataset and use 100 non-preferred demonstrations for the undesired dataset. The performance is shown in Figure 19. It is clearly that our method can achieve lower cost than SafeDICE.

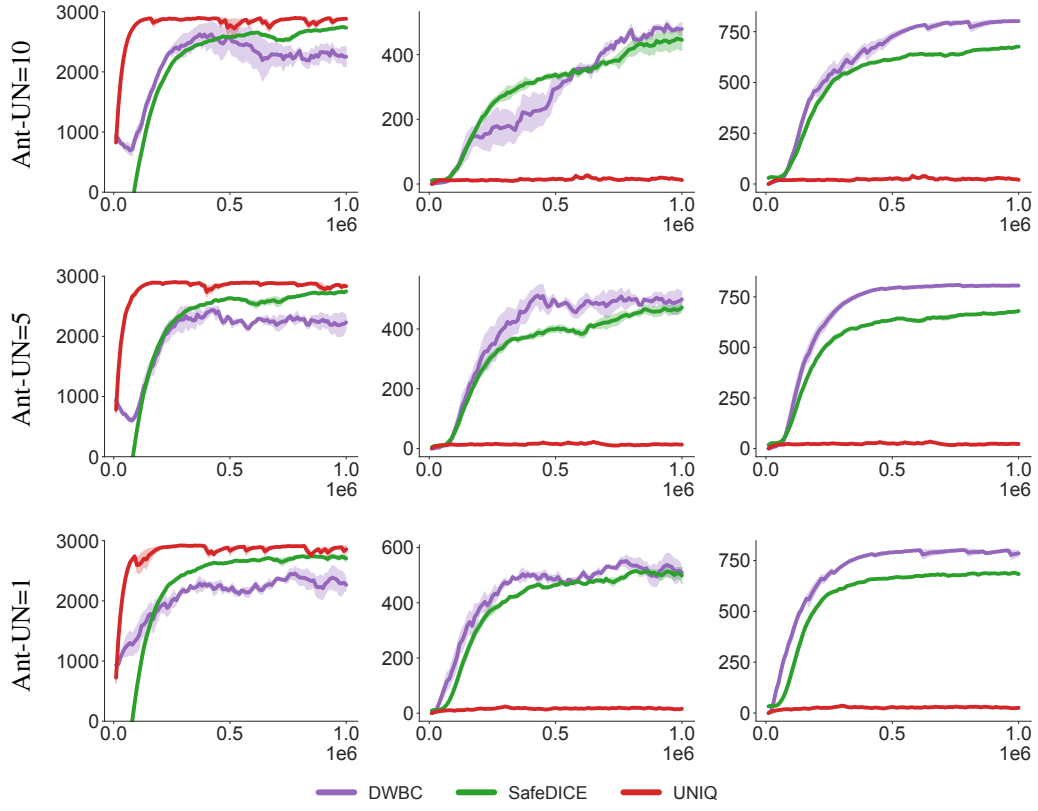


Figure 18: Ant task with unlabeled dataset(400-1600) and different undesired dataset.

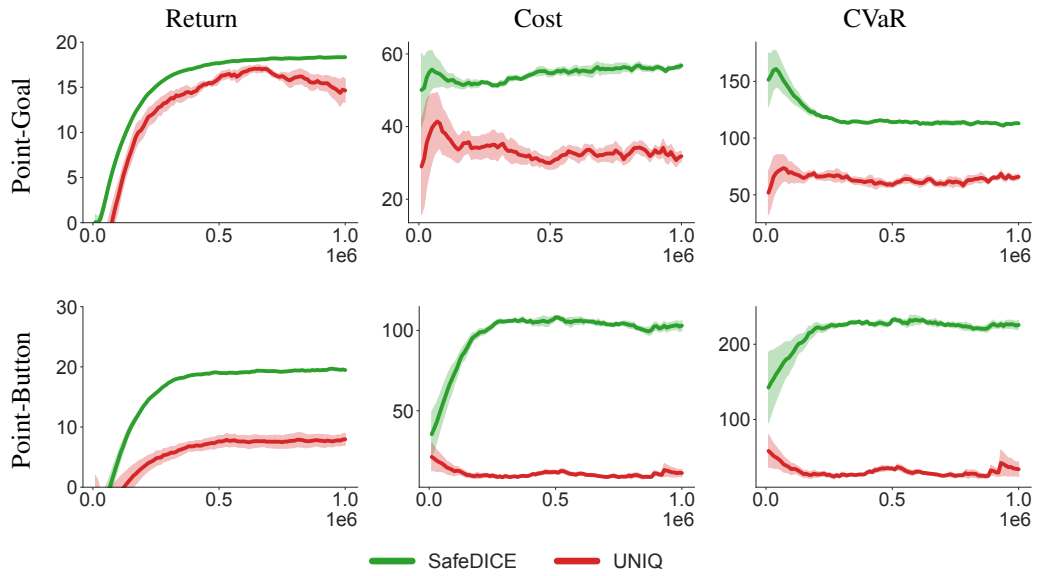


Figure 19: Comparison between UNIQ compared to SafeDICE in their dataset.