PRELIMINARY PREPRINT VERSION – DO NOT CITE
This is not the version of record. The final version will appear in the Proceedings of IJCAI-ECAI 2026.

Persistent Safety Set Guided Offline Safe Reinforcement Learning

Ayan Choudhury[1], **Janaka Brahmanage**[2], **Akshat Kumar**[2] and **Praveen Paruchuri**[1]
[1]International Institute of Information Technology, Hyderabad
[2]Singapore Management University
{ayan.choudhury@research, praveen.p}@iiit.ac.in, {janakat.2022@phdcs, akshatkumar}@smu.edu.sg

Abstract

Offline safe reinforcement learning learns high-return policies that satisfy hard safety constraints using only a pre-collected dataset. This setting is challenging due to the inability to explore, and the risk of propagating value errors through unsafe state-space regions. To address this, *first*, we characterize the safe state region by developing a framework for learning control barrier functions (CBFs) using a novel *generalized Bellman* operator, yielding a *persistent safety set*, from which the agent can remain safe indefinitely. *Second*, we show that several existing safety set estimation methods (e.g., reachability-constrained RL) can be formulated within our CBF learning framework, highlighting its generality. We further propose a new CBF that ensures safety under environment dynamics uncertainty, unlike standard CBFs designed for deterministic settings. *Third*, we propose a new reward maximization algorithm that effectively exploits our learned persistent safety set for reward critic estimation. Empirical results on standard benchmarks show that our approach achieves state-of-the-art safety with fewer constraint violations while maintaining competitive returns.

1 Introduction

Safety-critical tasks such as autonomous driving [Gulino *et al.*, 2023; Osinski *et al.*, 2019], robotics [Gu *et al.*, 2023], and healthcare [Fang *et al.*, 2024] require RL algorithms [Sutton and Barto, 2020] that maximize performance while guaranteeing constraint satisfaction [Wachi *et al.*, 2024]. A single violation can cause irrecoverable harm, making online exploration risky. In offline RL, where learning is from static datasets [Fujimoto *et al.*, 2019], ensuring safety is even more challenging due to risks of unsafe generalization, value overestimation, and reward propagation through unsafe regions. Prior work on safe RL has mainly used constrained optimization, introducing Lagrange multipliers [Gu *et al.*, 2024; Wachi *et al.*, 2024; Achiam *et al.*, 2017; Tessler *et al.*, 2018]. However, these approaches only encourage safety in expectation. An alternative line formulates safety via reachability: RCRL [Yu *et al.*, 2022] analyzes worst-case constraint evolution, which BCRL [Brahmanage and Kumar, 2026] extends to cumulative constraints, while FISOR [Zheng *et al.*, 2024] uses diffusion models to estimate safe trajectories. Inspired by control theory, these methods use control barrier functions (CBFs) to enforce forward invariance [Ames *et al.*, 2019]. Recent work further shows that value functions can act as barrier certificates [Tan *et al.*, 2023].

Recent surveys on learning CBFs [Wachi *et al.*, 2024; Ganai *et al.*, 2024] emphasize their importance for connecting verification, optimization, and learning, but also highlight challenges: (a) learning barrier certificates from limited data, (b) integrating barrier constraints into value learning, and (c) preventing safety information “leakage” through unsafe regions. Most offline safe RL algorithms [Chemingui *et al.*, 2025b; Koirala *et al.*, 2025] enforce safety only at policy extraction, allowing reward propagation through unsafe regions in the value critic and weakening safety guarantees. Our work addresses these issues by learning CBFs from offline data via a generalized Bellman operator and introducing principled methods to prevent safety leakage.

Related work. Recent work has connected control barrier functions (CBFs) and reinforcement learning (RL), especially in data-driven and offline settings. Neural CBFs trained from demonstrations or safe trajectories enable safety without explicit models [Harms *et al.*, 2024; Lindemann *et al.*, 2024; So *et al.*, 2024]. Some approaches estimate CBFs from offline data, but often rely on conservative assumptions or are limited to specific systems [Yu *et al.*, 2025; Tabbara and Sibai, 2025]. In offline RL, strategies like boundary-to-region supervision, diffusion regularization, and online optimization layers [Su *et al.*, 2025; Guo *et al.*, 2025a; Chemingui *et al.*, 2025a] mitigate unsafe generalization, but do not provide a unified framework for learning persistent safety sets via Bellman updates. Our work addresses this gap by formulating safety as a global, trajectory-level property, propagating worst-case violations through Bellman updates, and integrating safety critics with offline value-based RL to prevent unsafe value propagation during training.

Contributions. Our main contributions are as follows:

- **Generalized Bellman operator for CBF learning:** We develop a framework for learning discrete time CBFs via a generalized Bellman operator, and show that several existing CBFs are special cases. The fixed point of this operator corresponds to the largest forward-invariant safe set, providing a data-driven safety certificate.

- **Forward-invariant safety enforcement:** We integrate learned CBFs as an explicit safety critic in our offline safe RL algorithm, embedding them directly into actor–critic learning so that both value estimation and policy improvement are restricted to barrier-certified safe regions, preventing value propagation via unsafe regions, while still achieving high returns.
- **Empirical validation:** We demonstrate state-of-the-art safety constraint satisfaction with competitive returns on standard offline safe RL benchmarks.

Additional results, code and full paper with proofs are available at https://ayanroy24.github.io/GDCBF/.

2 Preliminaries

2.1 Offline Safe Reinforcement Learning

We formulate safe sequential decision making as a constrained MDP (CMDP), and subsequently address the challenges of learning from an offline dataset.

Constrained MDP. A CMDP is defined by the tuple $\mathcal{M} := \langle \mathcal{S}, \mathcal{A}, \mathcal{T}, r, c, \gamma, \mu_0, C_{\text{lim}} \rangle$. Here, $\mathcal{S}$ and $\mathcal{A}$ denote the state and action spaces, respectively [Altman, 2021]. The transition distribution $\mathcal{T}(s' \mid s, a)$ denotes the probability of transitioning to state s' given state s and action a. Reward function is defined as $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ and the cost function as $c : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^+$. The reward discount factor is $\gamma \in [0, 1)$, and μ_0 denotes the initial state distribution. The scalar C_{lim} represents the safety budget, defined as the maximum allowable expected cumulative cost.

The objective is to learn a policy $\pi : \mathcal{S} \to \mathcal{A}$ that maximizes the expected discounted return $J_R(\pi)$ subject to the safety constraint $J_C(\pi) \le C_{\text{lim}}$, formally stated as:

$$\begin{aligned} &\underset{\pi}{\text{maximize}} \quad J_R(\pi) = \mathbb{E}_\pi\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)\right] \\ &\text{subject to} \quad J_C(\pi) = \mathbb{E}_\pi\left[\sum_{t=0}^{\infty} \gamma^t c(s_t, a_t)\right] \le C_{\text{lim}} \end{aligned} \tag{1}$$

where $\mathbb{E}_\pi$ denotes the expectation over trajectories induced by π and $\mathcal{T}$. In this work, we set the safety budget $C_{lim} = 0$, addressing the challenging *hard constraint* setting where any constraint violation is inadmissible.

Learning from Offline Data. In complex real-world domains, obtaining an accurate analytical model of the environment (e.g., transition dynamics $\mathcal{T}$) is often intractable. Consequently, we adopt the *offline* safe RL setting, where the agent must learn solely from a fixed dataset $\mathcal{D} = \{(s_i, a_i, r_i, c_i, s'_i)\}_{i=1}^N$ collected by one or more behavioral policies π_β, without additional environment interactions.

2.2 Discrete Control Barrier Value Functions

Expected-cost constraints in CMDPs do not prevent transient or worst-case violations. In safety-critical domains, a single violation may be unacceptable regardless of the expected cost. Control barrier functions (CBFs) provide a stronger safety mechanism by characterizing safety through forward invariance of a feasible set. Next, we adapt CBFs [Agrawal and Sreenath, 2017] to discrete-time offline safe RL.

Safety is enforced through inequalities on value functions, enabling barrier-based reasoning directly from offline datasets. We learn CBFs via a Bellman-style update rule to learn a safety-critic $Q_h(s, a)$ that serves as a safety certificate. We define the safe set $\mathcal{C}$ based on state-action pairs, where the function $h : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ serves as a safety indicator.

Definition 2.1 (Safe Set)**.** The safe set $\mathcal{C}$ is defined as the set of state-action pairs satisfying the condition:

$$\mathcal{C} = \{(s, a) \in \mathcal{S} \times \mathcal{A} \mid h(s, a) \le 0\} \tag{2}$$

where $h(s, a)$ represents the immediate safety indicator (e.g., distance to a safety boundary), and positive values indicate a violation.

For CMDPs, the safety indicator $h(s, a)$ is directly derived from the cost function $c(s, a)$. Following [Zheng *et al.*, 2024], let $h(s, a) = c(s, a)$ if the transition is unsafe ($c(s, a) > 0$), and $h(s, a) = -1$ otherwise. This mapping ensures that $h(s, a) > 0$ corresponds to constraint violations, while $h(s, a) \le 0$ indicates safety compliance.

While $\mathcal{C}$ captures instantaneous safety, relying solely on it is myopic, as it does not ensure safety can be maintained over time. Guaranteeing long-term safety requires identifying regions where the dynamics allow indefinite constraint satisfaction, leading to two notions: forward invariance under a given policy, and the existence of at least one safe policy, which defines the persistent safety set. For exposition ease, we assume below a deterministic MDP; stochastic setting is addressed later.

Definition 2.2. A set $\mathcal{S}_\pi \subseteq \mathcal{S}$ is **forward invariant** with respect to a policy π if, for any initial state $s_0 \in \mathcal{S}_\pi$, the subsequent state trajectories generated by the deterministic transition function $f : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$, $s_{t+1} = f(s_t, \pi(s_t))$, satisfy:

$$s_t \in \mathcal{S}_\pi \quad \forall t \ge 0. \tag{3}$$

If $\mathcal{S}_\pi$ is a subset of the safe states (where $(s, \pi(s)) \in \mathcal{C}$), then the policy π guarantees safety for all time steps when initialized within $\mathcal{S}_\pi$, and thus π satisfies the CMDP safety constraint in (1).

Definition 2.3. The **persistent safety set** $S_P \subseteq \mathcal{S}$ is defined as the set of all states for which there exists a policy π such that the agent remains safe indefinitely. Formally:

$$S_P = \{s \in \mathcal{S} \mid \exists \pi, \forall t \ge 0, (s_t, a_t) \in \mathcal{C} \text{ given } s_0 = s\} \tag{4}$$

This set represents the maximal controlled invariant set within the safety constraints. Staying in this set enforces CMDP constraint in (1).

Explicitly computing set S_P requires solving a global reachability problem [Bansal *et al.*, 2017], which is computationally intractable [Mitchell *et al.*, 2005]. CBFs provide a tractable alternative. By satisfying a specific rate-of-change condition, the CBF function estimates the persistent safety set, effectively identifying the boundary of safe operation [Ames *et al.*, 2019].

Specifically, we learn a safety-critic $Q_h(s,a)$ that serves as a certificate for persistent safety. The indicator $h(s,a)$ captures whether an action is *immediately* safe, $Q_h(s,a)$ accounts for the worst-case future safety violations that may occur along subsequent trajectories. Intuitively, if $Q_h(s,a) \leq 0$, the state-action pair is *persistently* safe, meaning there exists a policy that can maintain safety indefinitely. To ensure forward invariance, we define the following condition:

Definition 2.4 (Deterministic Discrete CBF)**.** Consider a deterministic MDP with transition function f (Definition 2.2). A function $Q_h(s,a)$ is a discrete CBF if there exists a policy π such that for all (s,a) with $Q_h(s,a) \leq 0$:

$$Q_h(f(s,a), \pi(f(s,a))) \leq Q_h(s,a) - \alpha(Q_h(s,a)) \quad (5)$$

where $\alpha(\cdot)$ is an extended class $\mathcal{K}$ function [Agrawal and Sreenath, 2017].

An extended class-$\mathcal{K}$ function $\alpha : \mathbb{R} \to \mathbb{R}$ is continuous, strictly increasing, with $\alpha(0) = 0$; it modulates the decay rate in Eq.5, permitting residency in the safe set for $Q_h \leq 0$ while enforcing a corrective decrease toward safety for $Q_h > 0$. Hence $\alpha(q) > 0$ for $q > 0$ and $\alpha(q) < 0$ for $q < 0$, so $\alpha(Q_h)$ has the same sign as Q_h. A canonical choice is the linear form $\alpha(q) = \lambda q$ with $\lambda \in (0, 1]$. Intuitively, this inequality guarantees forward invariance by restricting the evolution of Q_h so that the agent is strictly prevented from crossing the safety boundary. The function $\alpha(\cdot)$ modulates this restriction, effectively dictating how aggressively the policy must steer away from the unsafe region based on the current safety margin. In stochastic environments, relying on a single next state or an expectation is insufficient for strict safety guarantees. We therefore adopt a robust formulation.

Definition 2.5 (Robust Discrete CBF Condition)**.** For a stochastic MDP defined by $\mathcal{T}(\cdot|s,a)$, a function $Q_h(s,a)$ is a robust discrete CBF if there exists a policy π such that for all (s,a) with $Q_h(s,a) \leq 0$:

$$\max_{s' \in \mathrm{supp}(\mathcal{T}(\cdot|s,a))} Q_h(s', \pi(s')) \leq Q_h(s,a) - \alpha(Q_h(s,a)) \quad (6)$$

where supp$(\mathcal{T}(\cdot|s,a))$ denotes the set of possible next states (or support), and $\alpha(\cdot)$ is an extended class $\mathcal{K}$ function.

This ensures that, even under worst-case transitions, the safety value remains decreasing (or sufficiently negative) over time. Small increases in Q_h may occur due to stochasticity, but α ensures forward invariance is maintained without requiring strict decrease.

3 Learning CBFs via Bellman Updates

3.1 Generalized Bellman Operator

Learning control barrier functions from data presents several challenges in the offline reinforcement learning setting. Classical CBF formulations typically assume access to system dynamics or require evaluating all possible successor states [Ames *et al.*, 2014], which is infeasible without an explicit model. Existing reinforcement learning approaches instantiate barrier functions using specific Bellman operators, such as hard maximum-based reachability updates [Yu *et al.*, 2022] or additive cost-style backups [Tan *et al.*, 2023], each inducing different trade-offs between safety and conservatism. Moreover, offline datasets provide limited transition coverage [Fujimoto *et al.*, 2019], requiring safety propagation mechanisms that reason about worst-case future violations rather than expected behavior. These challenges motivate the need for a generalized Bellman backup framework that unifies existing formulations while ensuring convergence and forward-invariant safety properties.

To learn this function, we define a generalized Bellman-like operator that propagates the worst-case safety constraint violation. This effectively enforces a reachability constraint: if a future state is unsafe, the current value must reflect that potential violation.

Assumption 3.1. The function $\Psi : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ satisfies the following properties: it is strictly increasing in its second argument. Furthermore, $\Psi(h, \cdot)$ is a contraction in the second argument, i.e., $|\Psi(h, v_1) - \Psi(h, v_2)| \leq \gamma |v_1 - v_2|$ with $\gamma < 1$ for all h.

Assumption 3.2. The function Ψ satisfies a consistency condition: for any h, v, if $q = \Psi(h, v)$, then $v \leq q - \alpha(q)$ for some extended class $\mathcal{K}$ function α. This ensures that the fixed point induces a valid decay for the safety value.

Intuitively, this condition ensures that any fixed point of the operator $\mathcal{B}$ automatically respects the required safety decay, guaranteeing that the learned value function acts as a valid barrier certificate.

Definition 3.1 (Generalized Reachability Bellman Operator)**.** We define the operator $\mathcal{B}$ acting on the state-action value function:

$$\mathcal{B}Q_h(s,a) = \Psi\left(h(s,a), \max_{s' \in \mathrm{supp}(\mathcal{T}(\cdot|s,a))} \min_{a' \in \mathcal{A}} Q_h(s', a')\right) \quad (7)$$

where Ψ mixes the immediate safety cost with the future cost.

3.2 Theoretical Analysis

Having defined the generalized operator, we now establish its theoretical properties. Specifically, we show that under the imposed assumptions on the mixing function Ψ, the operator defines a contraction mapping in the value function space. Consequently, its iterative application converges to a unique fixed point. Crucially, we prove that this fixed point satisfies the robust discrete CBF condition, thereby providing a valid certificate for the forward-invariant safety set.

Theorem 3.1. *Consider a CMDP with continuous safety indicator $h(s,a)$. The iterative application of the operator $\mathcal{B}$ defined in* (7) *converges to a unique fixed point Q_h^*. Furthermore, Q_h^* satisfies the Robust Discrete CBF condition defined in* (6)*, provided Assumptions 3.1 and 3.2 hold.*

The proof of Theorem 3.1 is available in Appendix A.1.

As a consequence of the fixed-point characterization, the boundary of the persistent safety set is given by the zero level-set of the learned barrier value function. In particular, the set

$$\mathcal{C}^* = \{(s,a) \mid Q_h^*(s,a) \leq 0\}$$

defines a forward-invariant safety region, while its boundary is implicitly induced by the condition $Q_h^*(s,a) = 0$. Thus, the safety boundary is not learned explicitly or parameterized a priori, but emerges naturally from the generalized Bellman updates.

Maximal Forward-Invariant Safe Set. As safety in sequential decision-making is inherently global and set-valued, validity requires guarantees over entire trajectories, robustness to worst-case dynamics, and compatibility with value-based policy optimization. Moreover, in the offline setting, safety guarantees must be grounded in dataset-supported transitions rather than extrapolated dynamics. Accordingly, we establish additional properties of the fixed point Q_h^*. These results show that the zero sublevel set of Q_h^* is not merely forward invariant but constitutes the *maximal* set of state–action pairs from which safety can be maintained indefinitely, ruling out unnecessary conservatism. We use the standard pointwise order on value functions, i.e., $Q_1 \preceq Q_2$ if $Q_1(s,a) \leq Q_2(s,a)$ for all (s,a), and define the safe set induced by Q as $\mathcal{C}_Q = \{(s,a) \mid Q(s,a) \leq 0\}$.

Lemma 3.2. A set $\mathcal{C} \subseteq \mathcal{S} \times \mathcal{A}$ is forward invariant if there exists a policy π such that, for all $(s,a) \in \mathcal{C}$ and all $s' \in \text{supp}(\mathcal{T}(\cdot|s,a))$, we have $(s', \pi(s')) \in \mathcal{C}$. A forward-invariant safe set $\mathcal{C}^*$ is maximal if, for any other forward-invariant safe set $\tilde{\mathcal{C}}$, it holds that $\tilde{\mathcal{C}} \subseteq \mathcal{C}^*$. Let Q_h^* be the fixed point of the operator $\mathcal{B}$ defined in (7). Then the set

$$\mathcal{C}^* := \{(s,a) \in \mathcal{S} \times \mathcal{A} \mid Q_h^*(s,a) \leq 0\}$$

is the maximal forward-invariant safe set induced by $\mathcal{B}$.

The proof of Lemma 3.2 is available in Appendix A.2.

4 Generalized Discrete CBF Variants

The theoretical framework established in Section 3 relies on a generalized mixing function Ψ. In practice, the choice of this function determines the nature of the learned safety certificate and its optimization landscape. In this section, we discuss what are the potential instantiations of the framework.

4.1 Deterministic Case

Maximum. The simplest instantiation of our framework aligns with the Reachability Constrained RL (RCRL) approach [Yu *et al.*, 2022]. Here, the operator propagates the maximum of the immediate safety cost and the discounted future safety value:

$$\mathcal{B}Q_h(s,a) = \max\{h(s,a), \gamma V_h(s')\} \tag{8}$$

where, $V_h(s') = \min_{a'} Q_h(s',a')$. This corresponds to $\Psi(h,v) = \max(h, \gamma v)$. The discount factor γ serves to decay the safety value over the horizon; specifically, it causes the safety margin of safe states (negative values) to degrade towards zero over time, while diminishing the magnitude of distant unsafe violations (positive values). While this operator strictly enforces the safety condition, it can suffer from value explosion if the horizon is long and costs are additive.

Smooth. The FISOR [Zheng *et al.*, 2024] algorithm employs a convex combination to smooth the transition between the immediate cost and the future value:

$$\mathcal{B}Q_h(s,a) = (1-\gamma)h(s,a) + \gamma \max\{h(s,a), V_h(s')\} \tag{9}$$

This corresponds to $\Psi(h,v) = (1-\gamma)h + \gamma \max(h,v)$. By incorporating the term $(1-\gamma)h(s,a)$, this operator effectively anchors the value estimate to the immediate safety cost. This anchoring ensures that the safety critic remains responsive to local constraint violations and stabilizes the learning process by balancing the immediate feedback with the bootstrapped lookahead.

Additive. The "Your Value Function is a Barrier Function" framework [Tan *et al.*, 2023] proposes an additive formulation where the Bellman operator aggregates the immediate safety cost and the discounted future value:

$$\mathcal{B}Q_h(s,a) = h(s,a) + \gamma V_h(s') \tag{10}$$

This corresponds to the linear aggregator $\Psi(h,v) = h + \gamma v$. Unlike the maximum-based operators, this formulation resembles the standard Bellman equation for cumulative cost. By interpreting the value function itself (shifted by a constant) as the barrier function, this approach ensures that the safety certificate is learned via standard value iteration techniques, provided the immediate cost $h(s,a)$ is designed to reflect the distance to the safety boundary.

Proposition 4.1. The Maximum, Smooth, and Additive update rules all satisfy Assumption 3.1 (Monotonicity and Contraction) with contraction factor γ. Furthermore, they satisfy the robust CBF condition (Assumption 3.2) with appropriate choice of α.

The proof of Proposition 4.1 is available in Appendix A.5.

4.2 Stochastic Case

The deterministic operators discussed above assume a single possible outcome for each action, which is often violated in real-world systems with aleatoric uncertainty or unmodeled dynamics. In such non-deterministic environments, the agent must satisfy the Robust Discrete CBF condition (Definition 2.5), which requires bounding the maximum violation over all possible next states: $\max_{s'} V_h(s')$.

In the offline setting, we lack an explicit transition model to query the set of reachable states. To overcome this, we propose to employ an *asymmetric aggregation* [Hansen-Estruch *et al.*, 2023] to approximate this worst-case future violation directly from the data distribution. We denote this asymmetric aggregation as E^τ, where the parameter τ can be adjusted to control the level of conservativeness when approximating the maximum. Common choices for such objectives include expectile regression, percentile (pinball) loss, or Gumbel regression [Garg *et al.*, 2023]. By tuning the parameter τ, we can estimate the upper tail of the future safety values observed in the dataset.

In practice, we apply the asymmetric aggregation *outside* the aggregator function Ψ to handle stochasticity in the transitions (s,a,s'). The update rule becomes:

$$\mathcal{B}Q_h(s,a) = \mathbb{E}^\tau_{s' \sim \mathcal{D}(s,a)} \left[\Psi\left(h(s,a),\ V_h(s')\right)\right] \tag{11}$$

As $\tau \to 1$, the expectile $\mathbb{E}^\tau$ approaches the maximum operator, it also approaches the worst-case max over the transition support as seen in Fig.2, thereby recovering the Robust Discrete CBF condition (Definition 2.5) which requires bounding the worst-case future violation. Note that Eq.11 adapts Eq.7 to the stochastic case: a direct max over transitions is impractical in offline data (identical (s,a) pairs rarely repeat), so asymmetric regression estimates the upper tail as a

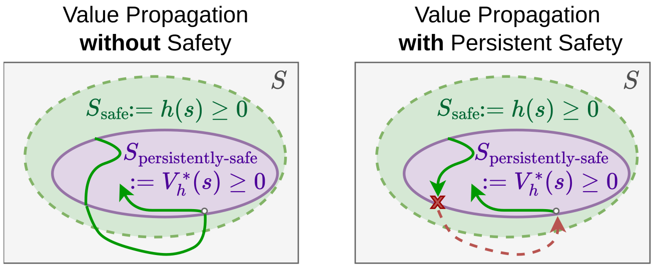

Figure 1: Value propagation with vs. without persistent safety. The figure shows a trajectory that briefly leaves the feasible region. Prior methods allow value propagation through unsafe states by enforcing safety only at policy extraction. Our approach enforces safety within the Bellman update, blocking value propagation through unsafe transitions.

data-efficient surrogate for worst-case estimation under limited coverage. In many practical implementations, Q-learning employs a Huber or L_1 loss [Mnih *et al.*, 2013] to handle outliers and stabilize training. The corresponding asymmetric variant for these robust losses is the percentile loss. Therefore, we utilize the percentile loss to implicitly approximate the robust target $\max_{s'} V_h(s')$ over the transition distribution, thereby enforcing safety against stochasticity in dynamics.

5 Utilizing Learned Persistent Safety Set for the Value Function Estimation

As illustrated in Figure 1, enforcing safety at the value function level prevents reward propagation through unsafe intermediate states—a common issue in prior works such as FISOR [Zheng *et al.*, 2024] that apply safety constraints only during policy extraction. We adopt an Implicit Q-Learning (IQL) [Kostrikov *et al.*, 2022] framework for offline RL to learn the optimal value function from the static dataset $\mathcal{D}$, while enforcing the persistent safety.

Safety-Aware Joint Bellman Target. A core challenge in offline safe RL is ensuring value estimates reflect safety constraints, even when the dataset contains unsafe trajectories. Discarding unsafe transitions is undesirable because they often contain useful information, including safe segments. To address this, we first learn the safety critic Q_h using the chosen CBF instantiation and then use it to define a modified Bellman target. We propose a joint Bellman target for the reward Q-function, Q_R, which explicitly handles safe and unsafe transitions based on learned persistent safety. The target $y_R(s, a, s')$ is defined based on the safety status predicted by our learned CBF Q_h^*:

$$y_R(s,a,s') = \begin{cases} r + \gamma V_R(s') & \text{if } Q_h^*(s,a) \leq 0 \\ \frac{r_{\min}}{1-\gamma} - Q_h^*(s,a) & \text{if } Q_h^*(s,a) > 0 \end{cases} \quad (12)$$

where $r_{\min} = \min_{(s,a,r,c,s') \in \mathcal{D}} r$ is the minimum reward observed across the entire offline dataset (a single dataset-wide constant, not a per-(s,a) minimum). For safe actions ($Q_h^*(s,a) \leq 0$), we update using the standard Bellman backup with the value function V_R. For unsafe actions

---

Algorithm 1 Barrier-Guided Offline Actor–Critic (CBF)

Require: Offline dataset $\mathcal{D} = \{(s,a,r,c,s')\}$, discount γ, expectiles (τ_r, τ_h), temperature β, target rate η, samples N

Initialize actor π_θ, reward critic Q_r^ω, value V_r^ψ, safety critic Q_h^ϕ, safety value V_h^ν, and target networks
for each gradient step **do**
  Sample a minibatch $\mathcal{M} = \{(s,a,r,c,s')\} \sim \mathcal{D}$
  Barrier target: $y_h \leftarrow \Psi(c, V_h(s'))$ for $(s,a,s') \in \mathcal{M}$
  Safety value update:
  $\nu \leftarrow \arg\min_\nu \mathbb{E}_\mathcal{M}\left[\mathcal{L}_{\tau_h}\left(Q_h^\phi(s,a) - V_h^\nu(s)\right)\right]$
  Safety critic update:
  $\phi \leftarrow \arg\min_\phi \mathbb{E}_\mathcal{M}\left[\left|Q_h^\phi(s,a) - y_h\right|\right]$
  Safety-aware reward target (Eq.(12)**):**

$$y_r = \begin{cases} r + \gamma V_r(s') & Q_h^\phi(s,a) \leq 0 \\ \frac{r_{\min}}{1-\gamma} - Q_h^\phi(s,a) & \text{otherwise} \end{cases}$$

  Reward critic update:
  $\omega \leftarrow \arg\min_\omega \mathbb{E}_\mathcal{M}\left[(Q_r^\omega(s,a) - y_r)^2\right]$
  Reward value update:

$$\psi \leftarrow \arg\min_\psi \mathbb{E}_\mathcal{M}\left[\mathcal{L}_{\tau_r}\left(Q_r^\omega(s,a) - V_r^\psi(s)\right)\right]$$

  Policy update: $w \leftarrow \exp\big(\beta(Q_r^\omega(s,a) - V_r^\psi(s))\big)$;
  $\theta \leftarrow \arg\min_\theta \mathbb{E}_\mathcal{M}[-w \log \pi_\theta(a|s)]$
  Update target networks
end for
Evaluation: sample $\{a_i\}_{i=1}^N \sim \pi_\theta(\cdot|s)$,
  return $a^\star = \arg\min_{a_i} Q_h(s, a_i)$

($Q_h^*(s,a) > 0$), we assign a penalty term. This penalty consists of the worst-case discounted return $\frac{r_{\min}}{1-\gamma}$ minus the magnitude of the safety violation $Q_h^*(s,a)$. This ensures that unsafe actions have strictly lower Q-values than safe actions, effectively creating a "soft" barrier in the value landscape.

We define *reward leakage* as the propagation of reward through unsafe transitions during Bellman updates, causing the reward critic to overestimate values for policies that violate safety. Our safety-aware target (Eq. 12) directly penalizes such transitions during training, preventing leakage. In contrast, FISOR [Zheng *et al.*, 2024] enforces safety only at policy extraction, allowing value propagation through unsafe regions. Enforcing safety at the value level ensures the Bellman operator converges to a safety-consistent fixed point as shown in figure 3 and the toy example in Appendix D. The reward Q-function is updated by minimizing the Mean Squared Error (MSE) loss:

$$L_Q(\theta) = \mathbb{E}_{(s,a,r,s') \sim \mathcal{D}}\left[\left(y_R(s,a,s') - Q_R(s,a)\right)^2\right] \quad (13)$$

Value Function Estimation via Expectile Regression. We then learn the state-value function $V_R(s)$ via expectile regression over the learned Q_R values:

$$L_V(\psi) = \mathbb{E}_{(s,a)\sim\mathcal{D}}\left[L_2^\tau(Q_R(s,a) - V_R(s))\right] \quad (14)$$

By utilizing the safety-aware Q_R, the expectile regression (with $\tau > 0.5$) naturally approximates the value of the best *safe* actions in the dataset distribution. In states where both

	CAPS		CCAC		LSPC		FISOR		CBF$_{smooth}$	
	reward ↑	cost↓	reward ↑	cost↓	reward ↑	cost↓	reward↑	cost↓	reward ↑	cost↓
CarButton1	**-0.04**$_{\pm 0.04}$	**0.98**$_{\pm 0.53}$	0.45$_{\pm 0.06}$	11.31$_{\pm 0.06}$	**-0.03**$_{\pm 0.03}$	**0.59**$_{\pm 0.11}$	**-0.02**$_{\pm 0.05}$	**0.26**$_{\pm 0.24}$	**0.12**$_{\pm 0.04}$	**0.47**$_{\pm 0.42}$
CarButton2	**-0.15**$_{\pm 0.02}$	**0.56**$_{\pm 0.01}$	0.53$_{\pm 0.01}$	9.82$_{\pm 0.01}$	**-0.13**$_{\pm 0.06}$	**0.93**$_{\pm 0.09}$	**0.01**$_{\pm 0.00}$	**0.58**$_{\pm 0.33}$	**0.05**$_{\pm 0.06}$	**0.13**$_{\pm 0.15}$
CarGoal1	**0.23**$_{\pm 0.04}$	**1.09**$_{\pm 0.57}$	0.84$_{\pm 0.13}$	1.81$_{\pm 0.14}$	**0.27**$_{\pm 0.22}$	**0.24**$_{\pm 0.47}$	**0.49**$_{\pm 0.03}$	**0.83**$_{\pm 0.16}$	**0.55**$_{\pm 0.33}$	**0.27**$_{\pm 0.41}$
CarGoal2	**0.14**$_{\pm 0.08}$	**0.52**$_{\pm 0.01}$	0.90$_{\pm 0.05}$	5.53$_{\pm 0.05}$	**0.16**$_{\pm 0.41}$	**0.41**$_{\pm 0.36}$	**0.06**$_{\pm 0.03}$	**0.33**$_{\pm 0.31}$	**0.18**$_{\pm 0.12}$	**0.20**$_{\pm 0.35}$
CarPush1	**0.20**$_{\pm 0.00}$	**0.32**$_{\pm 0.00}$	**-0.12**$_{\pm 0.00}$	**0.59**$_{\pm 0.09}$	**0.18**$_{\pm 0.07}$	**0.41**$_{\pm 0.12}$	**0.28**$_{\pm 0.04}$	**0.28**$_{\pm 0.15}$	**0.30**$_{\pm 0.16}$	**0.13**$_{\pm 0.28}$
CarPush2	0.05$_{\pm 0.03}$	2.31$_{\pm 1.03}$	0.12$_{\pm 0.03}$	6.06$_{\pm 0.03}$	**0.07**$_{\pm 0.05}$	**0.80**$_{\pm 0.02}$	**0.14**$_{\pm 0.07}$	**0.89**$_{\pm 0.51}$	**0.20**$_{\pm 0.12}$	**0.23**$_{\pm 0.25}$
AntVelocity	**0.87**$_{\pm 0.00}$	**0.22**$_{\pm 0.01}$	**0.51**$_{\pm 0.09}$	**0.59**$_{\pm 0.11}$	**0.97**$_{\pm 0.28}$	**0.14**$_{\pm 0.04}$	**0.89**$_{\pm 0.00}$	**0.00**$_{\pm 0.00}$	**0.88**$_{\pm 0.02}$	**0.17**$_{\pm 0.30}$
HalfCheetahVelocity	**0.87**$_{\pm 0.00}$	**0.29**$_{\pm 0.01}$	**0.94**$_{\pm 0.15}$	**0.94**$_{\pm 0.02}$	**0.97**$_{\pm 0.37}$	**1.00**$_{\pm 0.81}$	**0.89**$_{\pm 0.01}$	**0.00**$_{\pm 0.00}$	**0.93**$_{\pm 0.01}$	**0.00**$_{\pm 0.00}$
SwimmerVelocity	0.41$_{\pm 0.10}$	2.64$_{\pm 2.85}$	**-0.08**$_{\pm 0.01}$	**0.00**$_{\pm 0.00}$	0.46$_{\pm 0.19}$	1.28$_{\pm 0.15}$	**-0.04**$_{\pm 0.03}$	**0.00**$_{\pm 0.00}$	**0.56**$_{\pm 0.03}$	**0.21**$_{\pm 0.36}$
Average (9)	**0.29**$_{\pm 0.37}$	**0.99**$_{0.89}$	0.41$_{\pm 0.33}$	3.18$_{\pm 3.06}$	0.34$_{\pm 0.31}$	1.10$_{\pm 1.08}$	**0.30**$_{\pm 0.37}$	**0.35**$_{\pm 0.35}$	**0.42**$_{\pm 0.33}$	**0.24**$_{\pm 0.33}$
AntCircle	**0.32**$_{\pm 0.00}$	**0.00**$_{\pm 0.00}$	**0.54**$_{\pm 0.05}$	**0.00**$_{\pm 0.00}$	0.35$_{\pm 0.14}$	1.66$_{\pm 0.18}$	**0.20**$_{\pm 0.00}$	**0.00**$_{\pm 0.00}$	**0.43**$_{\pm 0.13}$	**0.12**$_{\pm 0.21}$
AntRun	0.53$_{\pm 0.00}$	1.83$_{\pm 0.00}$	**0.02**$_{\pm 0.00}$	**0.00**$_{\pm 0.00}$	0.75$_{\pm 0.03}$	15.58$_{\pm 0.04}$	**0.45**$_{\pm 0.08}$	**0.03**$_{\pm 0.01}$	**0.63**$_{\pm 0.09}$	**0.53**$_{\pm 0.31}$
BallCircle	**0.33**$_{\pm 0.00}$	**0.00**$_{\pm 0.00}$	**0.67**$_{\pm 0.14}$	**0.00**$_{\pm 0.00}$	**0.50**$_{\pm 0.31}$	**0.14**$_{\pm 0.07}$	**0.34**$_{\pm 0.01}$	**0.00**$_{\pm 0.00}$	**0.44**$_{\pm 0.02}$	**0.22**$_{\pm 0.21}$
BallRun	**0.10**$_{\pm 0.02}$	**0.00**$_{\pm 0.00}$	**0.31**$_{\pm 0.04}$	**0.00**$_{\pm 0.00}$	**0.17**$_{\pm 0.09}$	**0.00**$_{\pm 0.00}$	**0.18**$_{\pm 0.01}$	**0.00**$_{\pm 0.00}$	**0.27**$_{\pm 0.02}$	**0.00**$_{\pm 0.00}$
CarCircle	**0.42**$_{\pm 0.02}$	**0.01**$_{\pm 0.01}$	**0.68**$_{\pm 0.08}$	**0.74**$_{\pm 0.06}$	**0.70**$_{\pm 0.47}$	**0.36**$_{\pm 0.11}$	**0.40**$_{\pm 0.02}$	**0.11**$_{\pm 0.20}$	**0.58**$_{\pm 0.03}$	**0.12**$_{\pm 0.21}$
CarRun	**0.96**$_{\pm 0.00}$	**0.12**$_{\pm 0.00}$	**0.92**$_{\pm 0.09}$	**0.29**$_{\pm 0.02}$	0.97$_{\pm 0.42}$	1.52$_{\pm 0.39}$	**0.73**$_{\pm 0.02}$	**0.14**$_{\pm 0.22}$	**0.84**$_{\pm 0.07}$	**0.23**$_{\pm 0.34}$
DroneCircle	**0.34**$_{\pm 0.01}$	**0.00**$_{\pm 0.00}$	**0.14**$_{\pm 0.01}$	**0.17**$_{\pm 0.06}$	0.57$_{\pm 0.04}$	2.74$_{\pm 0.06}$	**0.48**$_{\pm 0.00}$	**0.00**$_{\pm 0.00}$	**0.37**$_{\pm 0.02}$	**0.38**$_{\pm 0.44}$
DroneRun	0.42$_{\pm 0.04}$	17.69$_{\pm 8.48}$	0.40$_{\pm 0.13}$	15.86$_{\pm 0.16}$	**0.56**$_{\pm 0.16}$	**0.00**$_{\pm 0.00}$	**0.30**$_{\pm 0.03}$	**0.55**$_{\pm 0.55}$	**0.55**$_{\pm 0.01}$	**0.00**$_{\pm 0.00}$
Average (8)	0.43$_{\pm 0.25}$	2.46$_{\pm 6.19}$	0.46$_{\pm 0.30}$	2.13$_{\pm 5.55}$	0.57$_{\pm 0.25}$	2.75$_{\pm 5.28}$	**0.39**$_{\pm 0.18}$	**0.10**$_{\pm 0.19}$	**0.59**$_{\pm 0.19}$	**0.18**$_{\pm 0.16}$
easysparse	**0.20**$_{\pm 0.15}$	**1.05**$_{\pm 1.07}$	**-0.05**$_{\pm 0.00}$	**0.32**$_{\pm 0.00}$	0.85$_{\pm 0.16}$	5.76$_{\pm 3.9}$	**0.38**$_{\pm 0.06}$	**0.53**$_{\pm 0.30}$	**0.52**$_{\pm 0.01}$	**0.46**$_{\pm 0.15}$
easymean	**0.15**$_{\pm 0.10}$	**0.61**$_{\pm 0.34}$	**-0.06**$_{\pm 0.01}$	**0.14**$_{\pm 0.10}$	0.78$_{\pm 0.05}$	3.59$_{\pm 1.27}$	**0.38**$_{\pm 0.05}$	**0.25**$_{\pm 0.16}$	**0.38**$_{\pm 0.07}$	**0.36**$_{\pm 0.37}$
easydense	**0.10**$_{\pm 0.01}$	**0.57**$_{\pm 0.31}$	**-0.05**$_{\pm 0.00}$	**0.24**$_{\pm 0.06}$	0.61$_{\pm 0.39}$	5.20$_{\pm 3.26}$	**0.36**$_{\pm 0.03}$	**0.25**$_{\pm 0.13}$	**0.55**$_{\pm 0.21}$	**0.61**$_{\pm 0.18}$
mediumsparse	**0.46**$_{\pm 0.12}$	**0.40**$_{\pm 0.63}$	**-0.08**$_{\pm 0.00}$	**0.24**$_{\pm 0.06}$	0.95$_{\pm 0.07}$	4.24$_{\pm 1.11}$	**0.42**$_{\pm 0.05}$	**0.22**$_{\pm 0.36}$	**0.54**$_{\pm 0.04}$	**0.23**$_{\pm 0.46}$
mediummean	**0.39**$_{\pm 0.03}$	**0.33**$_{\pm 0.25}$	**-0.06**$_{\pm 0.01}$	**0.06**$_{\pm 0.09}$	0.97$_{\pm 0.03}$	4.14$_{\pm 0.93}$	**0.39**$_{\pm 0.04}$	**0.08**$_{\pm 0.07}$	**0.54**$_{\pm 0.15}$	**0.30**$_{\pm 0.36}$
mediumdense	**0.50**$_{\pm 0.13}$	**0.36**$_{\pm 0.20}$	**-0.07**$_{\pm 0.00}$	**0.20**$_{\pm 0.00}$	0.70$_{\pm 0.39}$	2.77$_{\pm 1.73}$	**0.49**$_{\pm 0.05}$	**0.44**$_{\pm 0.32}$	**0.53**$_{\pm 0.16}$	**0.26**$_{\pm 0.15}$
hardsparse	0.32$_{\pm 0.04}$	1.76$_{\pm 0.32}$	**-0.05**$_{\pm 0.00}$	**0.24**$_{\pm 0.06}$	0.51$_{\pm 0.11}$	1.99$_{\pm 0.66}$	**0.30**$_{\pm 0.02}$	**0.01**$_{\pm 0.01}$	**0.42**$_{\pm 0.02}$	**0.77**$_{\pm 0.07}$
hardmean	**0.18**$_{\pm 0.01}$	**0.17**$_{\pm 0.15}$	**-0.05**$_{\pm 0.00}$	**0.20**$_{\pm 0.00}$	0.53$_{\pm 0.12}$	4.48$_{\pm 1.63}$	**0.26**$_{\pm 0.02}$	**0.09**$_{\pm 0.07}$	**0.36**$_{\pm 0.06}$	**0.28**$_{\pm 0.16}$
harddense	**0.25**$_{\pm 0.02}$	**1.03**$_{\pm 0.35}$	**-0.03**$_{\pm 0.01}$	**0.34**$_{\pm 0.12}$	0.49$_{\pm 0.12}$	3.56$_{\pm 1.87}$	**0.30**$_{\pm 0.03}$	**0.34**$_{\pm 0.31}$	**0.36**$_{\pm 0.09}$	**0.87**$_{\pm 0.42}$
Average (9)	**0.28**$_{\pm 0.14}$	**0.70**$_{\pm 0.50}$	**-0.06**$_{\pm 0.02}$	**0.22**$_{\pm 0.10}$	0.71$_{\pm 0.16}$	3.97$_{\pm 1.82}$	**0.36**$_{\pm 0.07}$	**0.25**$_{\pm 0.17}$	**0.47**$_{\pm 0.08}$	**0.46**$_{\pm 0.24}$

Table 1: Results for normalized cost and normalized reward where ↑ indicates that a higher value is better, while ↓ indicates that a lower value is preferable. **Bold** text denotes safe policies, Red indicates unsafe policies, and **Blue** highlights the highest reward among the safe policies for each task. Each result is obtained as an average by running over three random seeds and evaluating over 20 episodes.

safe and unsafe actions are present, the high penalty on unsafe actions ensures they fall below the expectile, causing $V_R(s)$ to converge towards the value of the safe actions. In states where only unsafe actions are available, $V_R(s)$ correctly reflects the low, penalized value, signaling the agent to avoid such states. In practice, we can either learn the CBF as a pre-training step and then learn the reward critic Q_R, or learn both in parallel. The latter approach aligns more closely with standard offline RL, which typically extracts the policy while learning the Q-value. Following this approach, we learn all three components—the CBF Q_h, the reward critic Q_R, and the policy π—concurrently, as presented in the algorithm pseudocode.

Pseudocode and Computational cost. Algorithm 1 summarizes our method where action-value functions (Q_h, Q_r) are parameterized to support Bellman updates and policy improvement, while value functions (V_h, V_r) are also learned via function approximation but optimized using expectile regression to provide stable targets. At each step, we fit the safety critic Q_h using a Bellman-like update, then use it to filter transitions for the reward critic Q_r: safe transitions use standard bootstrapped targets, while unsafe ones receive a pessimistic penalty based on $r_{\min}$. The policy is improved via advantage-weighted regression. This approach prevents reward propagation through unsafe states and ensures efficient, stable offline training. On a single RTX 2080 Ti GPU, 500k gradient steps require about 20 minutes.

6 Experiments

Goal of our experimental evaluation is:

1. Can a learned discrete control barrier function (CBF) effectively enforce hard safety constraints in the offline setting while obtaining high rewards ?
2. Assess how different deterministic CBFs—CBF$_{\text{smooth}}$ Eq.(9), CBF$_{\text{additive}}$ Eq.(10), and CBF$_{\text{maximum}}$ Eq.(8)—perform in the offline setting.
3. Does the robust smooth barrier update (section 4.2), utilizing an asymmetric loss, provides a safety advantage over the standard version in the presence of uncertainty?
4. Ablation studies to analyze the algorithm's sensitivity to hyperparameters (see Appendix B.2).

Evaluation Setup. We evaluate on diverse safety-critical continuous control tasks (Safety Gym, Bullet Gym, MetaDrive) using the same experimental setup as FISOR [Zheng *et al.*, 2024]. Rewards are normalized by the dataset's min and max, and costs by a predefined limit. A policy is considered safe if its normalized cost does not exceed one. Further details are in Appendix C.

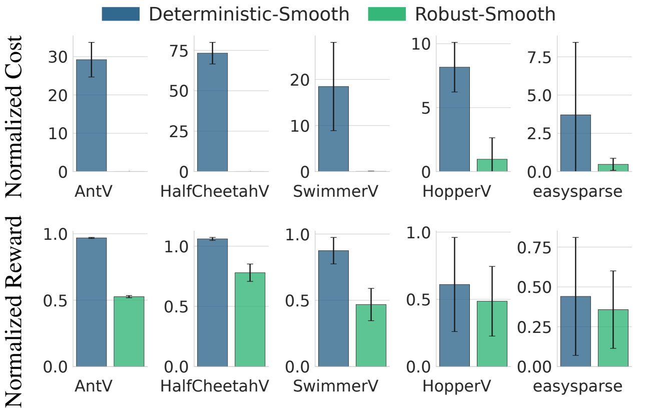

Figure 2: Effect of asymmetric aggregation on safety and reward propagation. This experiment evaluates whether the robust CBF variant in Sec. 4.2 improves performance under uncertainty. We add outliers to DSRL and compare Standard-Smooth with Robust-Smooth, which uses an asymmetric loss to approximate the maximum cost over the transition function. Across tasks, Robust-Smooth achieves lower normalized costs with comparable rewards.

Comparison with Baselines. We compare our method to safe offline RL baselines: CAPS [Chemingui *et al.*, 2025b], LSPC [Koirala *et al.*, 2025] (soft constraints), and CCAC [Guo *et al.*, 2025b], FISOR [Zheng *et al.*, 2024] (hard constraints). As shown in Table 1, methods enforcing safety only at policy extraction (e.g., CAPS, CCAC, LSPC) often achieve high rewards but violate constraints. In contrast, our approach (Algorithm 1) maintains near-zero cost across tasks, enforcing hard safety while achieving top rewards among safe policies. This is especially clear in recovery-heavy tasks, where restricting value backups to the barrier-certified set preserves high returns and prevents unsafe transitions. Thus, integrating a learned discrete CBF into value critic learning (Section 5) enables strict safety without sacrificing performance. FISOR and our method both satisfy the cost criterion, but our safety-aware Bellman target yields more accurate value estimates, supporting both high reward, safety.

Comparison of CBFs. A comparison of different barrier formulations is available in Appendix B.1, which reveals distinct trade-offs. For instance, $\text{CBF}_{\text{maximum}}$ tends to optimize for reward, while $\text{CBF}_{\text{smooth}}$ yields the most stable results. This distinction highlights that the propagation of safety information is a decisive factor in offline safe RL.

Robustness via Asymmetric Losses. Figure 2 compares the standard smooth (deterministic) barrier update (Eq. (9)) with its robust variant (Eq. (11)). To test robustness, we modify the DSRL dataset by mislabeling 70% of high-cost trajectories as safe using the DSRL's built-in functionality. We evaluate both deterministic and robust versions of Smooth CBF across velocity control and MetaDrive tasks. The robust version utilizes an asymmetric loss (percentile) with $\tau = 0.6$, while the standard version uses the standard L1 loss. Robust-Smooth consistently reduces normalized cost compared to the Deterministic-Smooth version. Particularly in velocity tasks, Robust-Smooth achieves nearly zero cost whereas the Deterministic-Smooth version incurs high costs. This improvement is achieved with only a modest reduction in normalized reward, demonstrating that asymmetric losses effectively estimate a persistent safety set under uncertainty. Importantly, these trends are consistent across all tasks, suggesting that improvements arise from stochasticity-aware safety estimation rather than task-specific tuning. Although, the robust variant uses asymmetric losses to approximate worst-case transitions, reducing costs on average in stochastic environments, but for *hardense*/*hardsparse*, limited data coverage causes $\mathbb{E}^\tau$ to underestimate extreme violations, leading to less conservative behavior and slightly higher costs than the deterministic formulation. Complete results across all velocity control, MetaDrive environments are in Appendix B.3.

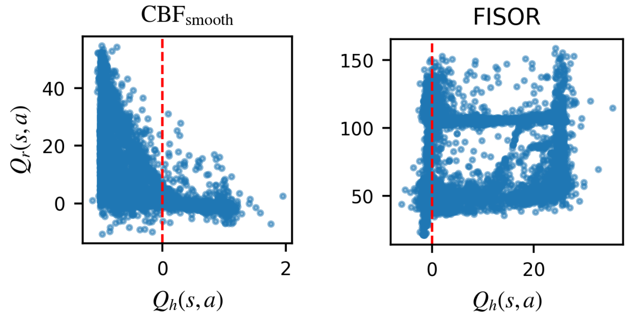

Figure 3: Reward critic $Q_r(s, a)$ vs. safety critic $Q_h(s, a)$ distributions for Swimmer Velocity. The red dashed line at $Q_h(s, a) = 0$ marks the safety boundary: points to the left are safe, and those to the right are unsafe. **Left:** $\text{CBF}_{\text{smooth}}$ prevents reward leakage as $Q_r(s, a)$ is low when $Q_h(s, a) > 0$ while **Right:** FISOR shows reward leakage as $Q_r(s, a)$ is high even for unsafe pairs.

Reward Leakage Analysis. To empirically evaluate reward leakage, we save the fully trained model and randomly sample 10K state-action pairs from the offline dataset. For each pair, we compute the safety critic $Q_h(s, a)$ and reward critic $Q_r(s, a)$ using the saved model, and then plot $Q_r(s, a)$ versus $Q_h(s, a)$ to verify that, for $Q_h(s, a) > 0$, the reward critic remains low, indicating that reward does not propagate through unsafe transitions. Figure 3 confirms that $\text{CBF}_{\text{smooth}}$ approach effectively blocks reward propagation through unsafe regions, ensuring that high rewards are only assigned to safe state-action pairs and providing strong safety guarantees during value learning.

7 Conclusions

We propose an offline safe RL framework that learns discrete CBFs via a generalized Bellman backup, unifying prior CBF methods with convergence guarantees. We further introduce a robust asymmetric variant for worst-case transitions and a safety-aware Bellman target to prevent reward leakage, together with a value-based offline RL method for safe policy extraction. Across continuous control and autonomous driving benchmarks, our method satisfies constraints while achieving competitive or superior reward.

Ethical Statement

There are no ethical issues.

Contribution Statement

Ayan Choudhury and Janaka Brahmanage contributed equally to this work.

References

[Achiam *et al.*, 2017] Joshua Achiam, David Held, Aviv Tamar, and Pieter Abbeel. Constrained policy optimization. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, page 22–31, 2017.

[Agrawal and Sreenath, 2017] Ayush Agrawal and Koushil Sreenath. Discrete Control Barrier Functions for Safety-Critical Control of Discrete Systems with Application to Bipedal Robot Navigation. In *Robotics: Science and Systems XIII*. Robotics: Science and Systems Foundation, 2017.

[Altman, 2021] Eitan Altman. *Constrained Markov Decision Processes: Stochastic Modeling*. Routledge, Boca Raton, 2021.

[Ames *et al.*, 2014] Aaron D. Ames, Jessy W. Grizzle, and Paulo Tabuada. Control barrier function based quadratic programs with application to adaptive cruise control. In *53rd IEEE Conference on Decision and Control*, pages 6271–6278, 2014.

[Ames *et al.*, 2019] Aaron D. Ames, Samuel Coogan, Magnus Egerstedt, Gennaro Notomista, Koushil Sreenath, and Paulo Tabuada. Control Barrier Functions: Theory and Applications. In *European Control Conference*, pages 3420–3431, 2019.

[Bansal *et al.*, 2017] Somil Bansal, Mo Chen, Sylvia Herbert, and Claire J. Tomlin. Hamilton-Jacobi Reachability: A Brief Overview and Recent Advances, September 2017. arXiv:1709.07523 [cs, math].

[Brahmanage and Kumar, 2026] Janaka Brahmanage and Akshat Kumar. Beyond hard constraints: Budget-conditioned reachability for safe offline reinforcement learning. In *International Conference on Automated Planning and Scheduling (ICAPS)*, 2026.

[Chemingui *et al.*, 2025a] Yassine Chemingui, Aryan Deshwal, Alan Fern, Thanh Nguyen-Tang, and Jana Doppa. Online optimization for offline safe reinforcement learning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[Chemingui *et al.*, 2025b] Yassine Chemingui, Aryan Deshwal, Honghao Wei, Alan Fern, and Janardhan Rao Doppa. Constraint-Adaptive Policy Switching for Offline Safe Reinforcement Learning. In *Proceedings of the Thirty-Ninth AAAI Conference on Artificial Intelligence and Thirty-Seventh Conference on Innovative Applications of Artificial Intelligence and Fifteenth Symposium on Educational Advances in Artificial Intelligence*, pages 15722–15730, 2025.

[Fang *et al.*, 2024] Nan Fang, Guiliang Liu, and Wei Gong. Offline Inverse Constrained Reinforcement Learning for Safe-Critical Decision Making in Healthcare, October 2024. arXiv:2410.07525 [cs].

[Fujimoto *et al.*, 2019] Scott Fujimoto, David Meger, and Doina Precup. Off-Policy Deep Reinforcement Learning without Exploration, August 2019. arXiv:1812.02900 [cs, stat].

[Ganai *et al.*, 2024] Milan Ganai, Sicun Gao, and Sylvia L. Herbert. Hamilton-Jacobi Reachability in Reinforcement Learning: A Survey. *IEEE Open Journal of Control Systems*, 3:310–324, 2024.

[Garg *et al.*, 2023] Divyansh Garg, Joey Hejna, Matthieu Geist, and Stefano Ermon. Extreme Q-Learning: MaxEnt RL without Entropy. In *The Eleventh International Conference on Learning Representations*, February 2023.

[Gu *et al.*, 2023] Shangding Gu, Jakub Grudzien Kuba, Yuanpei Chen, Yali Du, Long Yang, Alois Knoll, and Yaodong Yang. Safe multi-agent reinforcement learning for multi-robot control. *Artificial Intelligence*, 319:103905, 2023.

[Gu *et al.*, 2024] Shangding Gu, Long Yang, Yali Du, Guang Chen, Florian Walter, Jun Wang, and Alois Knoll. A Review of Safe Reinforcement Learning: Methods, Theory and Applications. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

[Gulino *et al.*, 2023] Cole Gulino, Justin Fu, Wenjie Luo, George Tucker, Eli Bronstein, Yiren Lu, Jean Harb, Xinlei Pan, Yan Wang, Xiangyu Chen, John D Co-Reyes, Rishabh Agarwal, Rebecca Roelofs, Yao Lu, Nico Montali, Paul Mougin, Zoey Yang, Brandyn White, Aleksandra Faust, Rowan McAllister, Dragomir Anguelov, and Benjamin Sapp. Waymax: An Accelerated, Data-Driven Simulator for Large-Scale Autonomous Driving Research. In *Proceedings of Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[Guo *et al.*, 2025a] Junyu Guo, Zhi Zheng, Donghao Ying, Ming Jin, Shangding Gu, Costas Spanos, and Javad Lavaei. Don't trade off safety: Diffusion regularization for constrained offline RL. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[Guo *et al.*, 2025b] Zijian Guo, Weichao Zhou, Shengao Wang, and Wenchao Li. Constraint-Conditioned Actor-Critic for Offline Safe Reinforcement Learning. In *The Thirteenth International Conference on Learning Representations*, 2025.

[Hansen-Estruch *et al.*, 2023] Philippe Hansen-Estruch, Ilya Kostrikov, Michael Janner, Jakub Grudzien Kuba, and Sergey Levine. IDQL: Implicit Q-Learning as an Actor-Critic Method with Diffusion Policies. *Artificial Intelligence (cs.AI), FOS: Computer and information sciences, Machine Learning (cs.LG)*, 2023. arxiv:2304.10573.

[Harms *et al.*, 2024] Marvin Harms, Mihir Kulkarni, Nikhil Khedekar, Martin Jacquet, and Kostas Alexis. Neural control barrier functions for safe navigation. In *IEEE/RSJ In-*

ternational Conference on Intelligent Robots and Systems*, pages 10415–10422, 2024.

[Koirala *et al.*, 2025] Prajwal Koirala, Zhanhong Jiang, Soumik Sarkar, and Cody Fleming. Latent Safety-Constrained Policy Approach for Safe Offline Reinforcement Learning. In *The Thirteenth International Conference on Learning Representations*, 2025.

[Kostrikov *et al.*, 2022] Ilya Kostrikov, Ashvin Nair, and Sergey Levine. Offline Reinforcement Learning with Implicit Q-Learning. In *The Tenth International Conference on Learning Representations*, 2022.

[Lindemann *et al.*, 2024] Lars Lindemann, Alexander Robey, Lejun Jiang, Satyajeet Das, Stephen Tu, and Nikolai Matni. Learning robust output control barrier functions from safe expert demonstrations. *IEEE Open Journal of Control Systems*, 3:158–172, 2024.

[Littman and Szepesvári, 1996] Michael L Littman and Csaba Szepesvári. A generalized reinforcement-learning model: Convergence and applications. In *Proceedings of the Thirteenth International Conference on International Conference on Machine Learning*, pages 310–318, 1996.

[Mitchell *et al.*, 2005] I.M. Mitchell, A.M. Bayen, and C.J. Tomlin. A time-dependent Hamilton-Jacobi formulation of reachable sets for continuous dynamic games. *IEEE Transactions on Automatic Control*, 50(7):947–957, 2005.

[Mnih *et al.*, 2013] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing Atari with Deep Reinforcement Learning, 2013.

[Osinski *et al.*, 2019] Blazej Osinski, Adam Jakubowski, Piotr Milos, Pawel Ziecina, Christopher Galias, Silviu Homoceanu, and Henryk Michalewski. Simulation-based reinforcement learning for real-world autonomous driving. *CoRR*, abs/1911.12905, 2019.

[So *et al.*, 2024] Oswin So, Zachary Serlin, Makai Mann, Jake Gonzales, Kwesi Rutledge, Nicholas Roy, and Chuchu Fan. How to train your neural control barrier function: Learning safety filters for complex input-constrained systems. In *IEEE International Conference on Robotics and Automation*, pages 11532–11539, 2024.

[Su *et al.*, 2025] Huikang Su, Dengyun Peng, Zifeng Zhuang, YuHan Liu, Qiguang Chen, Donglin Wang, and Qinghe Liu. Boundary-to-region supervision for offline safe reinforcement learning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[Sutton and Barto, 2020] Richard S. Sutton and Andrew Barto. *Reinforcement learning: an introduction*. Adaptive computation and machine learning. The MIT Press, Cambridge, Massachusetts London, England, second edition edition, 2020.

[Tabbara and Sibai, 2025] Ihab Tabbara and Hussein Sibai. Learning conservative neural control barrier functions from offline data, 2025. arXiv:2505.00908 [cs].

[Tan *et al.*, 2023] Daniel C H Tan, Fernando Acero, Robert McCarthy, Dimitrios Kanoulas, and Zhibin Li. Your Value Function is a Control Barrier Function. *International Conference on Machine Learning: 2nd Workshop on Formal Verification of Machine Learning (WFVML)*, 2023.

[Tessler *et al.*, 2018] Chen Tessler, Daniel J. Mankowitz, and Shie Mannor. Reward Constrained Policy Optimization. In *Proceedings of The Seventh International Conference on Learning Representations*, 2018.

[Wachi *et al.*, 2024] Akifumi Wachi, Xun Shen, and Yanan Sui. A Survey of Constraint Formulations in Safe Reinforcement Learning, May 2024. arXiv:2402.02025 [cs].

[Yu *et al.*, 2022] Dongjie Yu, Haitong Ma, Shengbo Eben Li, and Jianyu Chen. Reachability Constrained Reinforcement Learning. In *Proceedings of the 39th International Conference on Machine Learning*, 2022.

[Yu *et al.*, 2025] Hongzhan Yu, Seth Farrell, Ryo Yoshimitsu, Zhizhen Qin, Henrik I. Christensen, and Sicun Gao. Estimating control barriers from offline data, 2025. arxiv:2503.10641.

[Zheng *et al.*, 2024] Yinan Zheng, Jianxiong Li, Dongjie Yu, Yujie Yang, Shengbo Eben Li, Xianyuan Zhan, and Jingjing Liu. Safe Offline Reinforcement Learning with Feasibility-Guided Diffusion Model. In *The Twelfth International Conference on Learning Representations*, 2024.