V-OCBF: Learning Safety Filters from Offline Data via Value-Guided Offline Control Barrier Functions

Anonymous authors
Paper under double-blind review

Abstract

Ensuring safety in autonomous systems requires controllers that satisfy hard, state-wise constraints without relying on online interaction. While existing Safe Offline RL methods typically enforce soft expected-cost constraints, they do not guarantee forward invariance. Conversely, Control Barrier Functions (CBFs) provide rigorous safety guarantees but usually depend on expert-designed barrier functions or full knowledge of the system dynamics. We introduce Value-Guided Offline Control Barrier Functions (V-OCBF), a framework that learns a neural CBF entirely from offline demonstrations. Unlike prior approaches, V-OCBF does not assume access to the dynamics model; instead, it derives a recursive finite-difference barrier update, enabling model-free learning of a barrier that propagates safety information over time. Moreover, V-OCBF incorporates an expectile-based objective that avoids querying the barrier on out-of-distribution actions and restricts updates to the dataset-supported action set. The learned barrier is then used with a Quadratic Program (QP) formulation to synthesize real-time safe control. Across multiple case studies, V-OCBF yields substantially fewer safety violations than baseline methods while maintaining strong task performance, highlighting its scalability for offline synthesis of safety-critical controllers without online interaction or hand-engineered barriers.

1 Introduction

Ensuring the safety of autonomous systems is essential for their reliable and widespread deployment. From household service robots to autonomous vehicles and aerial drones, these systems increasingly operate in complex and unstructured environments where unsafe behavior can lead to irreversible consequences. As autonomy becomes deeply integrated into transportation, manufacturing, and healthcare, guaranteeing that such systems operate within well-defined safety boundaries is critical for reliability, and long-term adoption.

Reinforcement learning (RL) has emerged as a powerful paradigm for enabling autonomous systems to acquire sophisticated control behaviors. However, in safety-critical domains, naïve RL exploration can be hazardous. Although constrained RL (CRL) Achiam et al. (2017); Altman (2021); Alshiekh et al. (2018); Zhao et al. (2023) methods attempt to incorporate safety constraints during learning, they typically require extensive online interaction with the environment. However, most previous studies focus on online RL setting Liu et al. (2024), which suffers from serious safety issues in both training and deployment phases, especially for scenarios that lack high-fidelity simulators and require real system interaction for policy learning. As a result, there is a growing interest in synthesizing safe policies using offline RL or imitation learning Levine et al. (2020); Kumar et al. (2020). Nevertheless, most online and offline safe RL approaches Xu et al. (2022); Ciftci et al. (2024); Stooke et al. (2020) treat safety as a *soft constraint* and regulate only the expected cumulative constraint violations. Such probabilistic constraints are insufficient for applications that demand strict state-wise safety, where even a single violation is unacceptable. Furthermore, jointly optimizing performance and safety from static datasets often leads to unstable training dynamics and overly conservative behavior, particularly when safety-critical transitions are sparsely representedLee et al. (2022).

Control-theoretic tools provide an alternative and more rigorous foundation for safety. In particular, Control Barrier Functions (CBFs) Ames et al. (2014) offer a principled mechanism to enforce hard, instantaneous

safety constraints by guaranteeing the forward invariance of a prescribed safe set. When combined with learning-based controllers, CBFs serve as minimally invasive safety filters that adjust nominal actions only when necessary to prevent constraint violations. Their integration with Quadratic Program (QP) based controllers enables real-time implementation with modern optimization solvers. Consequently, CBF-based controllers have been successfully applied to a wide range of safety-critical tasks, including adaptive cruise control Ames et al. (2014), aerial robotics Wu & Sreenath (2016); Tayal et al. (2024a), and legged locomotion Nguyen & Sreenath (2015). In all of these applications, the performance and safety guarantees fundamentally depend on the quality of the underlying CBF.

Constructing valid CBFs, however, is a challenging problem. Hand-crafting barrier functions requires deep system knowledge and does not scale well to high-dimensional or partially known dynamical systems. This has motivated significant interest in Neural Control Barrier Functions (NCBFs), which leverage the expressive power of neural networks to approximate complex safe sets. A variety of techniques have been proposed for learning NCBFs, including SMT-based synthesis Abate et al. (2021; 2020), mixed-integer programming Zhao et al. (2022), nonlinear optimization Zhang et al. (2023), and loss-based training methods Dawson et al. (2022; 2023); Tayal et al. (2024b; 2025). Other recent approaches learn CBFs from value functions associated with nominal policies So et al. (2024). However, most of these methods rely on online interaction to collect informative samples or refine the barrier, which is often infeasible in safety-critical settings.

Recent work has explored learning Control Barrier Functions (CBFs) from offline demonstrations Robey et al. (2020); Castañeda et al. (2023); Tabbara & Sibai (2025). Existing methods either fit CBFs only on expert trajectories or rely on data-likelihood measures to filter unsafe samples, which limits their ability to generalize beyond the demonstrated states. Uncertainty-aware approaches address distributional mismatch but often become overly conservative. Overall, current offline CBF learning methods are closely tied to the empirical data distribution and do not explicitly reason about future system evolution, resulting in conservative safety guarantees.

This paper introduces Value-Guided Offline Control Barrier Functions (V-OCBF), a novel framework designed to overcome key limitations of existing offline RL and CBF-based approaches. We derive a model-free finite-difference recursion for updating the barrier function, and we show that satisfying this update provides a formal one-step safety guarantee for any control-affine system under the resulting policy. In addition, we propose an expectile-based learning objective that allows the synthesized safe policy to improve over the behavior policy in the dataset while never querying the barrier on out-of-distribution actions, ensuring stable and reliable offline learning. To summarise, the main contributions of this work are as follows:

1. We propose V-OCBF, a framework for learning formally safe controllers entirely from offline demonstrations.
2. We derive a model-free finite-difference barrier recursion and prove that adherence to this update guarantees one-step forward invariance for any control-affine system.
3. We introduce an expectile-based objective that improves upon the behavior policy without evaluating the barrier outside the dataset action support.
4. Across diverse systems, including high-dimensional Safety Gymnasium Ji et al. (2023) tasks, V-OCBF consistently outperforms constrained offline RL and neural CBF baselines in both safety and reward.

2 Related Works

2.1 Safe Offline RL

Safe RL has traditionally been studied in the online setting, where most methods rely on Lagrangian-based constrained optimization Chow et al. (2017); Tessler et al. (2018); Stooke et al. (2020). These approaches regulate safety through cost penalties, treating constraints as soft and therefore allowing non-zero violation risk. CPO Achiam et al. (2017) provides theoretical guarantees during updates through a trust-region

mechanism, but still cannot ensure strict safety throughout online learning. These limitations motivate shifting toward safe offline RL , where policies are synthesized from static data to avoid unsafe exploration. Among early offline-safe RL methods, CPQ Xu et al. (2022) assigns large costs to unsafe or out-of-distribution actions, but this can distort the value function and reduce generalization Li et al. (2023). COptiDICE Lee et al. (2022), building on DICE objectives Lee et al. (2021), inherits the difficulties of residual-gradient learning Baird (1995). More recent work integrates safety considerations into Decision Transformer Chen et al. (2021) or Diffuser models Janner et al. (2022), enabling sequence-modeling-based safety Liu et al. (2023); Lin et al. (2023); Zheng et al. (2024). However, these architectures are computationally expensive and challenging to scale. Overall, while offline RL mitigates unsafe online interaction, existing methods primarily enforce soft constraints and often struggle when unsafe transitions are underrepresented or missing from the dataset.

2.2 Offline Neural CBFs

Complementary to RL-based strategies, several recent methods aim to learn Control Barrier Functions (CBFs) directly from offline demonstrations. The approach in Robey et al. (2020) enforces differentiable barrier conditions on expert trajectories but does not account for out-of-distribution (OOD) states, limiting generalization beyond demonstrated data. The iDBF framework Castañeda et al. (2023) mitigates unsafe generalization by filtering low-likelihood states and actions using a Behavior Cloned policy, although this reliance on BC likelihood restricts coverage when demonstrations do not span the full safe set. Conservative CBFs Tabbara & Sibai (2025) incorporate uncertainty-aware penalties to avoid optimistic predictions on OOD states, but this often yields overly cautious certificates that substantially underestimate the true safe region. While these approaches represent important progress, they fundamentally rely on the empirical demonstration distribution and do not explicitly model how system dynamics may evolve toward unsafe states. Consequently, they tend to produce conservative CBFs that under-approximate the true safe set.

2.3 Positioning of Our Work

V-OCBF differs from safe offline RL methods Lee et al. (2022); Liu et al. (2023); Lin et al. (2023); Zheng et al. (2024), which rely on soft constraints, by directly constructing a hard safety certificate that guarantees one-step forward invariance. Unlike offline neural CBF approaches Robey et al. (2020); Castañeda et al. (2023); Tabbara & Sibai (2025) that remain tied to the demonstration distribution, V-OCBF employs a model-free value recursion that models future safety evolution and expands the safe set beyond observed states. Overall, our method combines the distributional robustness of offline RL with the rigor of CBF-based control to obtain an offline-learned, actuation-aware neural CBF without requiring online rollouts.

3 Background and Problem Setup

We consider a control-affine nonlinear dynamical system defined by the state $x(t) \in \mathcal{X} \subseteq \mathbb{R}^n$, the control input $u(t) \in \mathbb{U} \subseteq \mathbb{R}^m$, and governed by the following dynamics:

$$\dot{x}(t) = f(x(t)) + g(x(t))u(t), \tag{1}$$

where $f : \mathbb{R}^n \to \mathbb{R}^n$ and $g : \mathbb{R}^n \to \mathbb{R}^{n\times m}$ are locally Lipschitz continuous functions. We are given a set $\mathcal{C} \subseteq \mathcal{X}$ that represents the *safe states* for the system and a failure set $\mathcal{F} \subseteq \mathcal{X}$ that represents the set of unsafe states for the system (e.g., obstacles for an autonomous ground robot). Furthermore, the system is controlled by a Lipschitz continuous control policy $\pi : \mathbb{R}^n \to \mathbb{R}^m$. Our focus lies in ensuring the safety of this dynamical system, which is formally defined as follows:

Definition 1 (Safety)**.** *A dynamical system is considered safe if the set, $\mathcal{C} \subseteq \mathcal{X} \subseteq \mathbb{R}^n$, is positively invariant under the control policy, π, i.e, $x(0) \in \mathcal{C}, u(t) = \pi(x(t)) \implies x(t) \in \mathcal{C},\ \forall\ t \geq 0$.*

Since, $\mathcal{F} \subseteq \mathcal{X} \setminus \mathcal{C}$, it can be trivially shown that $x(t) \in \mathcal{C} \implies x(t) \notin \mathcal{F}\ \forall\ t \geq 0$. Using this premise, we define the main objective of this paper:

Objective 1. *Our objective is to synthesize a safe policy $\pi_{\text{safe}} : [t, T) \times \mathcal{X} \to \mathbb{U}$ such that the resulting closed-loop system satisfies the positive invariance property specified in Definition equation 1.*

3.1 Control Barrier Functions

Control Barrier Functions Ames et al. (2014; 2017) are widely used to synthesize control policies with positive invariance guarantees, thereby ensuring system safety. The initial step in constructing a Control Barrier Function (CBF) involves defining a continuously differentiable function $B : \mathcal{X} \to \mathbb{R}$, where the *super-level set* of B corresponds to the safe region $\mathcal{C}$. This leads to the following representation:

$$\mathcal{C} = \{x \in \mathcal{X} : B(x) \geq 0\}, \quad \mathcal{X} \setminus \mathcal{C} = \{x \in \mathcal{X} : B(x) < 0\}. \tag{2}$$

The interior and boundary of $\mathcal{C}$ are further specified as:

$$\text{Int}(\mathcal{C}) = \{x \in \mathcal{X} : B(x) > 0\}, \quad \partial\mathcal{C} = \{x \in \mathcal{X} : B(x) = 0\}. \tag{3}$$

The function h qualifies as a valid Control Barrier Function if it satisfies the following definition:

Definition 2 (Ames et al. (2017))**.** *Given a control-affine system $\dot{x} = f(x) + g(x)u$, the set $\mathcal{C}$ defined by equation 2, with $\frac{\partial B}{\partial x}(x) \neq 0$ for all $x \in \partial\mathcal{C}$, the function B is called the Control Barrier Function (CBF) defined on the set $\mathcal{X}$, if there exists an extended class-$\mathcal{K}$ function κ such that for all $x \in \mathcal{X}$:*

$$\max_{u \in \mathbb{U}} \left[\underbrace{\mathcal{L}_f B(x) + \mathcal{L}_g B(x) u}_{\dot{B}(x,u)} + \kappa\left(B(x)\right) \right] \geq 0, \tag{4}$$

where $\mathcal{L}_f B(x) = \frac{\partial B}{\partial x} f(x)$ and $\mathcal{L}_g B(x) = \frac{\partial B}{\partial x} g(x)$ are the Lie derivatives and n is the dimension of the system.

As established in Ames et al. (2017), any Lipschitz continuous control law $\pi(x)$ that satisfies the condition $\dot{B} + \kappa(B) \geq 0$ guarantees the system's safety when $x(0) \in \mathcal{C}$. Additionally, if the initial state $x(0)$ lies outside $\mathcal{C}$, this condition ensures asymptotic convergence to the safe set $\mathcal{C}$.

While CBFs provide a principled framework to guarantee safety, their practical deployment is hindered by the lack of general methods for constructing valid barrier functions. As a result, practitioners typically resort to handcrafted or domain-specific CBFs, which can yield overly conservative safe sets. Furthermore, in the presence of control bounds, a nominal CBF may conflict with feasibility requirements, causing the corresponding CBF-QP to become infeasible.

3.2 Control Barrier Value Function (CBVF)

To overcome the limitations inherent in classical CBF formulations, Choi et al. (2021) proposed the *Control Barrier Value Function* (CBVF), which integrates Control Barrier Functions with Hamilton–Jacobi (HJ) Reachability Bansal et al. (2017). We begin by encoding the safety specification using a Lipschitz continuous function $\ell : \mathbb{R}^n \to \mathbb{R}$, where the failure set is defined as $\mathcal{F} := \{x \in \mathcal{X} \mid \ell(x) < 0\}$. Under this construction, a CBVF $B : \mathcal{X} \to \mathbb{R}$ is defined as the viscosity solution of the following Hamilton–Jacobi–Bellman Variational Inequality (HJB-VI):

$$\min\left\{ \max_{u \in \mathbb{U}} (\nabla B(x) \cdot (f(x) + g(x)u)),\ \ell(x) - B(x) \right\} = 0, \tag{5}$$

with boundary condition $B(x)|_{t=0} = \ell(x)|_{t=0}$. The resulting value function induces a forward-invariant safe set $\mathcal{C} := \{x \in \mathcal{X} \mid B(x) \geq 0\}$ and ensures that admissible controls $u \in \mathbb{U}$ satisfy the following Lie-derivative condition:

$$\max_{u \in \mathbb{U}} \left[L_f B(x) + L_g B(x) u + \kappa(B(x)) \right] \geq 0. \tag{6}$$

Safe Controller Synthesis using CBVFs: Quite often, we have a reference control policy, $\pi_{ref}(x)$, designed to meet the performance requirements of the system. However, such controllers often lack safety

guarantees. To ensure the system meets its safety requirements while preserving performance, the reference controller must be minimally adjusted to incorporate safety constraints. This adjustment can be accomplished using the Control Barrier Value Function-based Quadratic Program (CBVF-QP), described as follows:

$$\begin{aligned} \pi_{\text{safe}}(x) = \min_{u \in \mathbb{U} \subseteq \mathbb{R}^m} & \left\| u - \pi_{ref} \right\|^2 \\ \text{s.t. } & \mathcal{L}_f B(x) + \mathcal{L}_g B(x) u + \kappa \left(B(x) \right) \geq 0. \end{aligned} \tag{7}$$

The CBVF-QP framework facilitates the synthesis of a provably safe control policy, $\pi_{\text{safe}}(x)$, while staying close to the reference controller to preserve system performance.

Challenges in CBVF Synthesis: Traditional approaches compute Control Barrier Value Functions using grid-based HJ reachability methods Mitchell (2005), which are fundamentally limited by the curse of dimensionality. Recent efforts have attempted to overcome these issues by learning CBVFs through online reinforcement learning So et al. (2024); however, as discussed in Section 1, such approaches require extensive online interaction, rendering them unsuitable for safety-critical systems. Furthermore, existing neural CBF methodologies Abate et al. (2020); Zhang et al. (2023); Tayal et al. (2024b) typically assume access to accurate system dynamics, an assumption that does not hold for many real-world platforms. These limitations motivate a shift toward using offline demonstrations, either sourced from public datasets Liu et al. (2024); Sun et al. (2020) or collected in controlled settings where safety can be guaranteed. Building on this premise, we refine Objective 1 as follows:

Objective 2. *Our objective is to synthesize a Control Barrier Value Function $B : \mathcal{X} \to \mathbb{R}$ directly from an offline dataset of demonstrations $\mathcal{D}$, such that the resulting CBVF-QP controller π_{safe} in equation 7 satisfies the positive invariance property specified in Definition 1.*

4 Methodology

Having introduced the CBVF formulation in the previous section, we now describe a practical methodology for synthesizing a valid Control Barrier Value Function and thereby the safe controller from offline demonstrations $\mathcal{D}$. Our objective is to construct a data-driven approximation of the viscosity solution of the HJB-VI equation 5 without relying on known system dynamics or online interaction. The key idea is to re-interpret it through a finite-difference barrier recursion compatible with demonstration data, and to use this recursion to learn a value-guided barrier function that inherits forward invariance.

4.1 Finite-Difference Barrier Synthesis

To approximate the CBVF using trajectories in $\mathcal{D}$, we consider a finite difference recursive version of equation equation 5. Given a trajectory $\{x_t, u_t\}_{t=0}^{N}$, we define the finite-difference barrier update

$$B(x_t) = \min \left\{ \ell(x_t), \max_{u_t} B(x_{t+1}) \right\}, \qquad \forall t \in \{0, 1, 2, \dots\}, \tag{8}$$

where, $x_t = x(t)$, $u_t = u(t)$ and $x_{t+1} = x(t + \Delta t)$, along with the boundary condition $B(x_0) = \ell(x_0)$. This recursion has two significant advantages. First, it enables us to learn a barrier directly from data without requiring f and g. Second, under mild regularity assumptions, equation 8 preserves the forward invariance property of the CBVF. The derivation of the recursive equation is provided in Appendix A.1. Intuitively, the recursion encodes the principle that a state is safe if and only if its immediate successor is safe or it lies outside the unsafe set as specified by $\ell(x)$.

To represent this barrier function, we parameterize a neural network $B_\psi^{\mathcal{D}}(x)$ with ψ utilizing the universal approximation property. However, directly solving for the recursion equation 8 can lead to degenerate solutions. For instance, setting $B_\psi^{\mathcal{D}}(x) = c$ for a sufficiently small constant c satisfies equation 8 but clearly does not satisfy the CBVF conditions in equation 5. This pathology is analogous to the non-contractive behavior

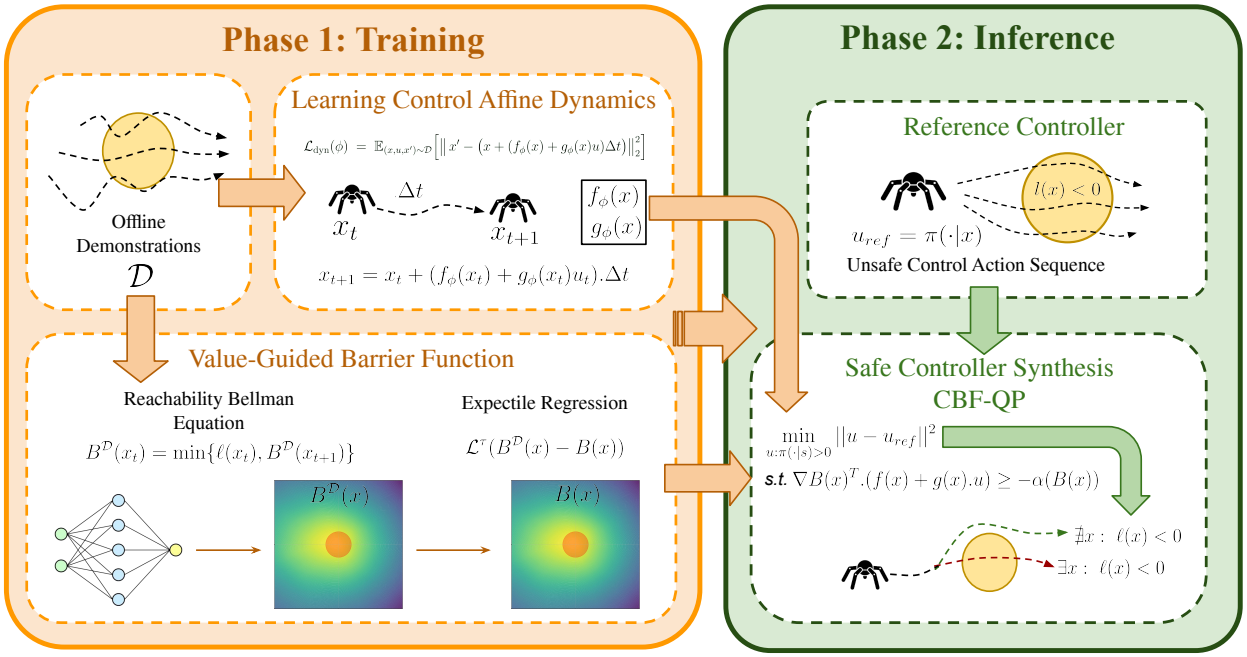

Figure 1: **Framework Overview:** *(Left)*: We learn value guided barrier function with reachability based bellman equation for learning optimal safe region and apply expectile regression for OOD case handling, when learning from Offline Dataset. *(Right)*: Inferencing with CBF-QP using learned barrier function as valid CBF to rollout safe step-wise actions, with any reference controller.

of undiscounted value iteration in MDPs. Following the approach in Fisac et al. (2019), we incorporate a discounted finite-difference loss to avoid such trivial solutions:

$$\mathcal{L}_{B^{\mathcal{D}}}(\psi) = \mathbb{E}_{(x,u,x')\sim\mathcal{D}}\left[\left((1-\gamma)\,\ell(x) + \gamma \min\left\{\ell(x),\, B^{\mathcal{D}}_{\psi}(x')\right\} - B^{\mathcal{D}}_{\psi}(x)\right)^2\right], \tag{9}$$

where $x = x_t$, $x' = x_{t+1}$ and $u = u_t$ and $\gamma \to 1$. This discounted recursion ensures contraction, promotes stable learning, and prevents the network from collapsing to uniformly unsafe or uniformly safe solutions. To avoid evaluating barrier targets at out-of-distribution actions, we remove the maximization over actions from the loss in equation 9 and use only the demonstrated action in each transition. While this prevents unsupported queries, the resulting estimate reflects the safety profile of the behaviour policy that generated the data. Consequently, this produces a behaviour-induced barrier, which is typically sub-optimal because it ignores other admissible actions that could yield larger safe-set estimates.

4.2 Avoiding Out-of-Distribution Actions in Offline Learning

The naïve regression objective in equation 9 fits $B^{\mathcal{D}}_{\psi}$ to the mean of the demonstrated next-state targets, but this corresponds to the behavior-induced barrier and yields overly conservative safe sets. Ideally, if we assume unlimited capacity and no sampling error, the optimal parameters should satisfy, $B(x) \approx \mathbb{E}_u\left[\min\left\{\ell(x),\, \max_u B(x')\right\}\right]$. However, such unconstrained maximization can result in actions that are never observed in the dataset.

Since offline data only provides information about those actions selected by the behavior policy, evaluating values (or barrier targets) using unsupported actions can distort learning because the corresponding transitions are not grounded in the dataset. Subsequently, motivated by the insights of Implicit Q-Learning (IQL) Kostrikov et al. (2022), we approximate the maximization over admissible actions by using expectile regression, which enables us to capture the highest admissible barrier values supported by the data while never evaluating B on unseen (x, u) pairs. This allows us to perform a principled value-style backup over the dataset control action support $\mathcal{U}_{\mathcal{D}} = \{\, u \mid (x, u, x') \in \mathcal{D} \,\}$ without extrapolating to unsafe or unobserved

actions. IQL shows that expectile regression produces a value function that reflects the values induced by the behavior policy without requiring an explicit behavior model. This prevents the learning target from being influenced by actions that lie outside the dataset support, while still capturing the highest feasible values supported by the demonstrations.

Following this principle, we estimate a CBVF, B_θ, with θ as the Neural Network parameters, that reflects the safety values implied by demonstrated actions. Formally, we minimize the expectile loss

$$\mathcal{L}_B(\theta) \;=\; \mathbb{E}_{(x,u,x')\sim\mathcal{D}}\big[\mathcal{L}^\tau\big(B^{\mathcal{D}}_\psi(x) - B_\theta(x)\big)\big], \tag{10}$$

where $\mathcal{L}^\tau(y) = |\tau - \mathbf{1}(y<0)|\, y^2$ is the τ-expectile loss used in Kostrikov et al. (2022). Intuitively, a higher expectile level τ places greater weight on underestimation errors than overestimation errors, pushing B_θ toward the upper envelope of safety values supported by the dataset. Thus, τ controls how aggressively the learned barrier emphasizes high, data-supported safety values without extrapolating to unseen actions. The barrier function thus obtained, B_θ, is our proposed *Value-guided Offline Control Barrier Function* (V-OCBF). The complete learning procedure for $B^{\mathcal{D}}_\psi$ and the V-OCBF B_θ is summarized in Algorithm 1 and can be visually seen in Fig. 1.

Algorithm 1 Learning V-OCBF from Offline Demonstrations

Require: Offline dataset $\mathcal{D}$, expectile level τ, batch size m, learning rates η_ϕ, η_θ, discount factor γ

```
Initialize ψ, θ and optimizers
for epoch = 1 to N do
    for minibatch B ⊂ D do
        (A) B^D_ψ update
        L_{B^D} ← (x, u, x', γ, ψ) with (x, u, x') ∼ B          ▷ From Eq. 9
        ψ ← AdamStep(ψ, ∇_ψ L_{B^D})
        (B) form support targets
        For each unique x ∈ D gather support U_D from D
        (C) B-network losses
        L_B ← (x, U_D, τ, B^D_ψ, θ)                             ▷ From Eq. 10
        θ ← AdamStep(θ, ∇_θ L_B)
    end for
end for
Return B_θ
```

4.3 Controller Synthesis via Learned Dynamics

The learned barrier function B is subsequently employed to fulfill the primary goal of synthesizing a safe policy 1 using the CBVF-QP formulation in equation 7. Solving this QP necessitates the evaluation of the Lie derivatives $\mathcal{L}_f B$ and $\mathcal{L}_g B$, both of which rely on the underlying control-affine system dynamics. In our offline-only setting, the true dynamics are unavailable; therefore, we construct a neural network–based surrogate model to approximate the underlying transition dynamics of the form:

$$x_{t+1} = x_t + \big(f_\phi(x_t) + g_\phi(x_t)u_t\big)\,\Delta t, \tag{11}$$

which enables computation of the required derivatives and supports safe policy synthesis. The model parameters ϕ are trained using one-step transitions from the offline dataset $\mathcal{D}$ by minimizing the prediction loss:

$$\mathcal{L}_{\text{dyn}}(\phi) = \mathbb{E}_{(x,u,x')\sim\mathcal{D}}\Big[\,\|\, x' - \big(x + (f_\phi(x) + g_\phi(x)u)\Delta t\big)\,\|^2\Big], \tag{12}$$

implemented as a minibatch MSE objective. The full training procedure is summarized in Algorithm 2.

Importantly, the learned dynamics model is *not* used when learning the barrier function. Incorporating it into the CBVF learning stage would require evaluating terms involving (f_ϕ, g_ϕ) under actions outside the dataset-supported set $\mathcal{U}_\mathcal{D}$, thereby violating the action constraints critical for preventing value underestimation in the

offline regime. Hence, using learned dynamics during CBVF training would allow the network to extrapolate into unsupported regions of the action space, defeating the purpose of the OOD-aware barrier learning objective described earlier.

In contrast, at *inference* time, the learned dynamics serve a different role: they enable the evaluation of Lie derivatives needed to solve the CBVF-QP (equation 7). Specifically, for any query state x, we compute

$$\mathcal{L}_f B(x) = \nabla_x B_\theta(x)^\top f_\phi(x), \qquad \mathcal{L}_g B(x) = \nabla_x B_\theta(x)^\top g_\phi(x). \tag{13}$$

These quantities allow the QP in equation 7 to be solved for the safe control action u_{safe}, completing the pipeline for constructing a safety-certified controller purely from offline demonstrations.

In Section 5.2.2, we provide an analysis demonstrating the limitations of using learned dynamics inside the CBVF learning loop, further reinforcing the necessity of restricting their use to the inference stage only.

Algorithm 2 Controller Synthesis

Require: Offline dataset $\mathcal{D} = \{(x, u, x')\}$, batch size m, epochs N, learning rate, Δt, $u_{ref}(\cdot)$, κ

```
Initialize ϕ and optimizer
for epoch = 1 to N do
    for minibatch B ⊂ D do
        L_dyn(ϕ) ← (x, u, x', Δt, ϕ)                          ▷ From Eq. 12
        ϕ ← AdamStep(ϕ, ∇_ϕ L_dyn)
    end for
end for

Inference at state x̂
L_f B(x̂) ← (∇_x B_θ(x̂), f_ϕ(x̂)) and L_g B(x̂) ← (∇_x B_θ(x̂), g_ϕ(x̂))   ▷ From Eq. 13
u_safe ← QP-Solver(L_f B(x̂), L_g B(x̂), u_ref(x̂), B_θ(x̂), κ)          ▷ From Eq. 7
return safe controller u_safe(x̂)
```

5 Experiments

The experiments are designed to evaluate: (*i*) the safety and performance of V-OCBF relative to constrained offline RL and neural CBF baselines on systems with unknown dynamics, (*ii*) the advantages of value-guided barriers over behavior-policy–induced barriers, (*iii*) the robustness of the resulting QP controller under external disturbances, and (*iv*) the effectiveness of V-OCBF compared to a CBVF synthesized using learned dynamics.

Baselines: We compare V-OCBF against a diverse set of constrained offline learning and CBF-based methods. For constrained offline learning, we include **Behavior Cloning (BC)**, **BEAR-Lag** (Lagrangian constraint version of Kumar et al. (2019)), **COptiDICE** Lee et al. (2022), and **FISOR** Zheng et al. (2024) which enforce safety indirectly via soft constraints on policy optimization or behavior imitation. For CBF-based approaches, we evaluate **Neural Control Barrier Function (NCBF)** Robey et al. (2020), *Conservative Control Barrier Function (CCBF)* Tabbara & Sibai (2025), and *In-Distribution Barrier Function (iDBF)* Castañeda et al. (2023), which synthesize explicit safety filters from offline data but often yield conservative safe sets. In contrast, V-OCBF learns a *value-guided* barrier function from offline demonstrations that accounts for future unsafe interactions, producing a hard, state-wise safety filter with larger safe set coverage.

Evaluation Metrics: We evaluate all methods based on (i) *safety*, measured as the total number of safety violations incurred before episode termination, and (ii) *performance*, measured via the cumulative episode rewards. These metrics allow us to assess the trade-off between strict safety enforcement and task performance across different offline RL and CBF-based approaches.

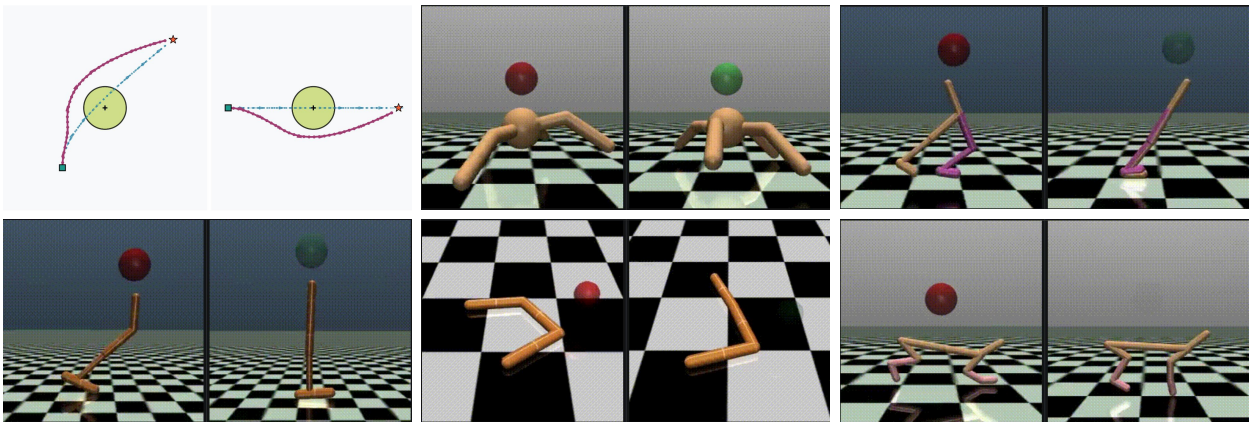

Figure 2: **Illustrations of the test environments:** (*Top-Left*) environment illustrates Autonomous Ground Vehicle with an obstacle and a goal point. (*MuJoCo Env*) **Red** sphere denotes unsafe state.

5.1 Experimental Case Studies

To perform a holistic performance analysis of our proposed approach, we apply V-OCBF in conjunction with Behavior Cloning (BC) as the nominal (reference) controller for all the different environments which are supposed to assess varying objectives. Below we list all the environments that we use:

- **Autonomous Ground Vehicle (AGV) Collision Avoidance:** In our first experiment, we examine a 3-dimensional collision avoidance problem involving an autonomous ground vehicle governed by Dubins' car dynamics (Dubins, 1957). The objective is to ensure safety by avoiding a static obstacle while navigating through a bounded environment. Further details on the system dynamics, state space bounds, and experimental setup are provided in Appendix B.1.

- **MuJoCo Safety Gymnasium:** We next evaluate our framework on Safety Gymnasium environments Ji et al. (2023). Specifically, we evaluate the V-OCBF-based QP (equation 7) on high-dimensional MuJoCo tasks like Hopper, Swimmer, Half Cheetah, Walker2D and Ant. The objective in each environment is to maximize reward while keeping the agent velocity below the velocity thresholds. We keep the reward and safety-violation metrics identical to the Safety-Gymnasium definitions and use the standard DSRL dataset for safe offline RL Liu et al. (2024). To evaluate our method against baselines, we randomly sampled 500 initial states for each environment, respectively, the results for which can be referred to from Figure 3.

5.2 Results

5.2.1 Effectiveness in Co-optimizing Safety and Performance

We begin by evaluating all methods on the AGV Collision Avoidance task, which provides a clear setting to study how different approaches balance safety and performance. The results in Table 1 highlight notable differences in how offline RL and CBF-based methods handle this trade-off.

Offline RL baselines such as BC, BEAR-Lag, and COptiDICE achieve relatively low safety rates. BC tends to reproduce unsafe behaviors from the dataset, while BEAR-Lag and COptiDICE try to account for safety but remain limited because they operate with soft constraint formulations. Their lower reward and safety scores indicate that they struggle to balance both safety and performance objectives.

In contrast, methods that incorporate a CBF-QP layer, such as BC+NCBF, BC+iDBF, and BC+CCBF, achieve much higher safety rates. The QP ensures that unsafe actions are filtered out, even if the nominal controller is imperfect. However, the performance of these approaches still depends heavily on the quality of the learned barrier. FISOR performs better than the other offline RL baselines because it explicitly

Table 1: AGV Collision Avoidance Experiment: Percentage Safe Episodes, Mean Episode Reward and Safe Set Volume (refer Appendix B.1.1) across different methods. Evaluated over 500 episodes and 5 seed values.

Method	**Safe Episodes (%)**	**Episode Reward**	**Safe Set Volume (%)**
BC	48.92 ± 1.69	20.45 ± 1.84	42.51
BEAR-Lag Kumar et al. (2019)	65.12 ± 0.24	13.85 ± 0.81	58.21
COptiDICE Lee et al. (2022)	68.91 ± 0.32	15.33 ± 0.67	62.32
BC+NCBF Robey et al. (2020)	92.48 ± 0.60	44.61 ± 2.58	81.92
BC+iDBF Castañeda et al. (2023)	92.87 ± 0.73	48.23 ± 2.01	83.32
BC+CCBF Tabbara & Sibai (2025)	93.56 ± 0.56	49.66 ± 2.34	90.94
FISOR Zheng et al. (2024)	95.78 ± 0.2	52.33 ± 0.93	90.14
BC+**V-OCBF** (Ours)	**98.28 ± 0.54**	**54.93 ± 0.46**	**92.57**

expands the feasible safe region before optimizing for performance. However, due to lack of explicit safety filtering, it leads to lesser safety rates than our proposed method, due to the impending learning errors in the computation of feasible region. This also highlights the importance of QP based safety filtering scheme for achieving better safety. Overall, *V-OCBF achieves the strongest results across all metrics.*

To further analyze scalability, we extend the evaluation to MuJoCo Safety Gymnasium environments (Hopper, Half-Cheetah, Ant, Swimmer, and Walker2D), with unknown dynamic models. The results in Figure 3 demonstrate that V-OCBF again achieves the lowest safety violation rates across all tasks. Notably, the method maintains near-zero violations on while preserving satisfactory reward levels compared to BC and outperforming iDBF, NCBF, and CCBF. These neural CBF baselines degrade sharply in higher dimensions: NCBF suffers from optimization difficulties, while iDBF often enforces overly restrictive boundaries that suppress task performance. FISOR again remains competitive but does not match the safety consistency of V-OCBF.

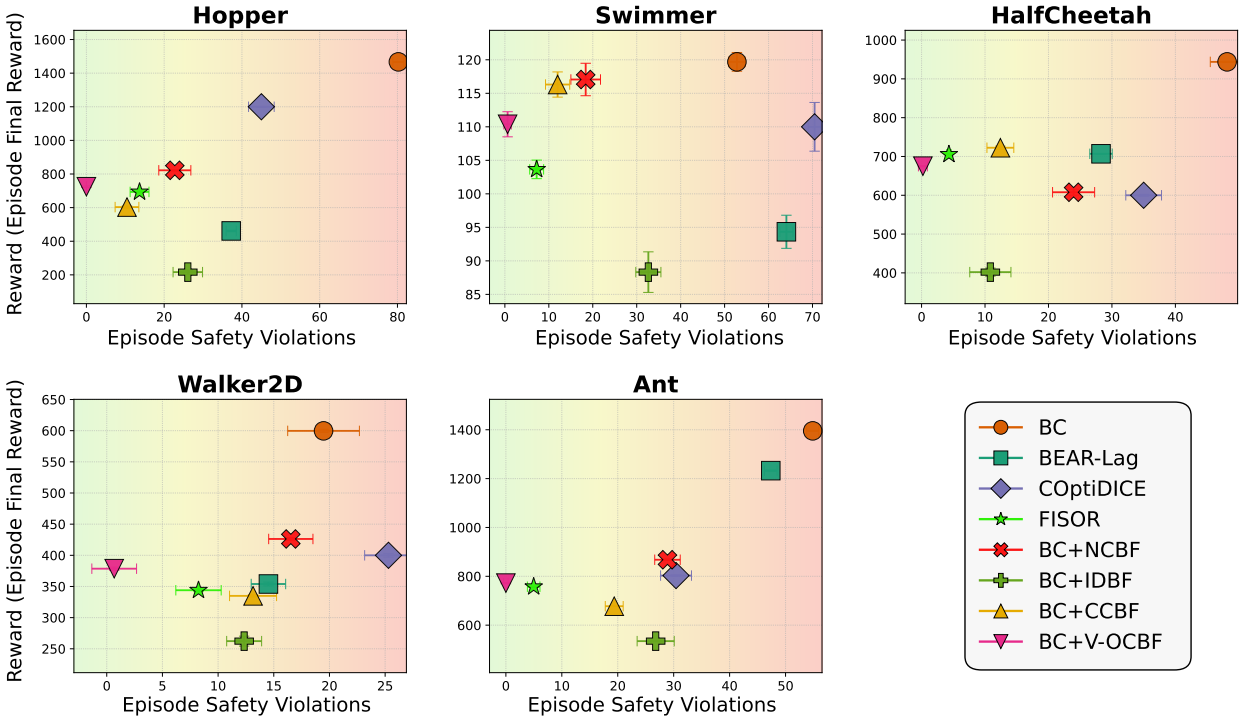

Figure 3: Results plot for all the Mujoco based Safety Gymnasium Ji et al. (2023) environments. Points towards left (←) are more safe than those on right (→). Evaluated over 500 episodes and 5 seed values.

Overall, the experiments provide strong empirical evidence that V-OCBF effectively co-optimizes safety and performance, scaling from low-dimensional AGV dynamics to complex MuJoCo systems. The method consistently outperforms existing offline RL and neural CBF baselines in terms of safety while maintaining competitive reward, highlighting its suitability for offline settings where both strict safety and reliable performance are required.

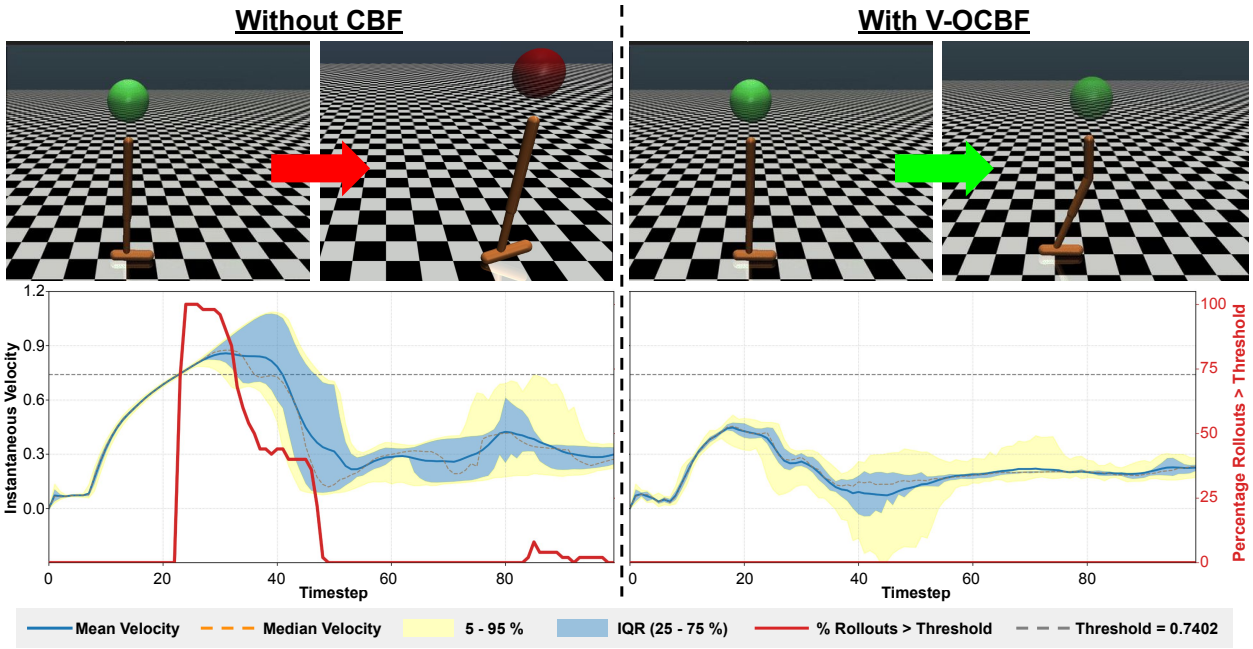

Figure 4: **Qualitative comparison of Hopper rollouts with and without V-OCBF.** *(Top):* When using V-OCBF, the agent adapts its gait to keep its forward velocity within the allowable safety threshold, demonstrating how the learned barrier actively regulates unsafe accelerations. *(Bottom):* Instantaneous velocities across episodes for both methods. The left axis reports the velocity profile, while the right axis shows the fraction of episodes that violate the safety threshold at each timestep. V-OCBF substantially reduces violation frequency, illustrating its ability to enforce safety constraints during deployment. Additional qualitative results for other environments are provided in Appendix B.2.1.

5.2.2 Effect of learned dynamics

We additionally examine how using the learned dynamics influences both CBVF learning and QP-based control on the AGV environment, where ground-truth dynamics are available for comparison. Table 2 reports safety and reward for three configurations: (i) V-OCBF with a QP evaluated using learned dynamics, (ii) V-OCBF with a QP evaluated using true dynamics, and (iii) a neural CBVF trained using learned dynamics and QP-based controller evaluated on the same dynamics.

Across all metrics, using learned dynamics at inference time yields performance that is close to the known-dynamics case, indicating that the controller synthesis using QP is robust to moderate model error. In contrast, using learned dynamics during CBVF training leads to a noticeable drop in both safety and reward. This aligns with our earlier discussion in section 4.3: learning the barrier through a model introduces distributional mismatch because the learned dynamics may generate actions outside the dataset-supported set $\mathcal{U}_{\mathcal{D}}$, yielding suboptimal targets for B. These findings reinforce our design choice, dynamics should be used only at inference to compute the lie derivatives in the QP, while barrier learning itself should avoid dependence on a learned model.

5.3 Ablation Studies

To further evaluate the robustness of V-OCBF, we conduct ablation studies that examine: (i) the sensitivity of the learned safe set to the size of the barrier network, (ii) the effect of disturbances on the safety of

Table 2: Use of Learned Dynamics at the time of Training v/s Inference (Mean over 500 episode rollouts).

Method	**Safe Episodes (%)**	**Episode Reward**
BC+**V-OCBF** (QP with Learned Dynamics)	98.28 ± 0.54	54.93 ± 0.46
BC+**V-OCBF** (QP with Known Dynamics)	99.52 ± 0.58	55.91 ± 0.39
BC+**CBVF from Learned Dynamics**	94.64 ± 0.65	46.40 ± 0.02

QP based controller, and (iii) the dependence of the learned barrier on the expectile parameter τ. Across these experiments, V-OCBF exhibits stable safety performance over a broad range of architectural and hyperparameter choices, remaining resilient even under perturbations to the control actions. These findings suggest that the framework generalises reliably beyond the specific tasks considered and is well-suited for settings where safety certification must be carried out under imperfect system knowledge. The full results and visualisations for these studies are presented in Appendix C.

6 Conclusion, Limitations and Future works

This work presented V-OCBF, a value-guided framework for learning control barrier functions directly from offline demonstrations. By combining a finite-difference CBVF recursion with an expectile-based objective, V-OCBF propagates safety information in a principled manner while avoiding evaluations on unsupported actions, enabling reliable CBF-QP control even when dynamics are unknown and action authority is limited. Experiments across AGV and high-dimensional Safety Gymnasium environments show that V-OCBF consistently reduces safety violations while maintaining strong task performance, outperforming both constrained offline RL and neural CBF baselines.

A current limitation is that the framework is not explicitly trained to handle adversarial disturbances or worst-case model errors. Future work will explore robust extensions of V-OCBF that incorporate adversarial or uncertainty-aware training objectives, enabling safety guarantees under stronger disturbance models and broader distribution shifts.

References

Alessandro Abate, Daniele Ahmed, Mirco Giacobbe, and Andrea Peruffo. Formal synthesis of lyapunov neural networks. *IEEE Control Systems Letters*, 5(3):773–778, 2020.

Alessandro Abate, Daniele Ahmed, Alec Edwards, Mirco Giacobbe, and Andrea Peruffo. Fossil: A software tool for the formal synthesis of Lyapunov functions and barrier certificates using neural networks. In *Proceedings of the 24th International Conference on Hybrid Systems: Computation and Control*, pp. 1–11, 2021.

Joshua Achiam, David Held, Aviv Tamar, and Pieter Abbeel. Constrained policy optimization. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, pp. 22–31. JMLR.org, 2017.

Mohammed Alshiekh, Roderick Bloem, Rüdiger Ehlers, Bettina Könighofer, Scott Niekum, and Ufuk Topcu. Safe reinforcement learning via shielding. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32 (1), Apr. 2018. doi: 10.1609/aaai.v32i1.11797. URL `https://ojs.aaai.org/index.php/AAAI/article/view/11797`.

Eitan Altman. *Constrained Markov decision processes*. Routledge, 2021.

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*, pp. 6271–6278. IEEE, 2014.

Aaron D. Ames, Xiangru Xu, Jessy W. Grizzle, and Paulo Tabuada. Control barrier function based quadratic programs for safety critical systems. *IEEE Transactions on Automatic Control*, 62(8):3861–3876, 2017. doi: 10.1109/tac.2016.2638961.

Leemon Baird. Residual algorithms: Reinforcement learning with function approximation. In *Machine Learning Proceedings 1995*, pp. 30–37. Elsevier, 1995.

Somil Bansal, Mo Chen, Sylvia Herbert, and Claire J Tomlin. Hamilton-jacobi reachability: A brief overview and recent advances. In *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, pp. 2242–2253. IEEE, 2017.

Fernando Castañeda, Haruki Nishimura, Rowan Thomas McAllister, Koushil Sreenath, and Adrien Gaidon. In-distribution barrier functions: Self-supervised policy filters that avoid out-of-distribution states. In Nikolai Matni, Manfred Morari, and George J. Pappas (eds.), *Proceedings of The 5th Annual Learning for Dynamics and Control Conference*, volume 211 of *Proceedings of Machine Learning Research*, pp. 286–299. PMLR, 15–16 Jun 2023. URL `https://proceedings.mlr.press/v211/castaneda23a.html`.

Lili Chen, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Misha Laskin, Pieter Abbeel, Aravind Srinivas, and Igor Mordatch. Decision transformer: Reinforcement learning via sequence modeling. *Advances in neural information processing systems*, 34:15084–15097, 2021.

Jason J Choi, Donggun Lee, Koushil Sreenath, Claire J Tomlin, and Sylvia L Herbert. Robust control barrier–value functions for safety-critical control. In *2021 60th IEEE Conference on Decision and Control (CDC)*, pp. 6814–6821. IEEE, 2021.

Yinlam Chow, Mohammad Ghavamzadeh, Lucas Janson, and Marco Pavone. Risk-constrained reinforcement learning with percentile risk criteria. *The Journal of Machine Learning Research*, 18(1):6070–6120, 2017.

Yusuf Umut Ciftci, Darren Chiu, Zeyuan Feng, Gaurav S Sukhatme, and Somil Bansal. Safe-gil: Safety guided imitation learning for robotic systems. *arXiv preprint arXiv:2404.05249*, 2024.

Charles Dawson, Zengyi Qin, Sicun Gao, and Chuchu Fan. Safe nonlinear control using robust neural Lyapunov-barrier functions. In *Conference on Robot Learning*, pp. 1724–1735. PMLR, 2022.

Charles Dawson, Sicun Gao, and Chuchu Fan. Safe control with learned certificates: A survey of neural Lyapunov, barrier, and contraction methods for robotics and control. *IEEE Transactions on Robotics*, 2023.

Lester E Dubins. On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents. *American Journal of Mathematics*, 79(3):497–516, 1957.

Jaime F. Fisac, Neil F. Lugovoy, Vicenç Rubies-Royo, Shromona Ghosh, and Claire J. Tomlin. Bridging hamilton-jacobi safety analysis and reinforcement learning. In *2019 International Conference on Robotics and Automation (ICRA)*, pp. 8550–8556, 2019. doi: 10.1109/ICRA.2019.8794107.

Michael Janner, Yilun Du, Joshua Tenenbaum, and Sergey Levine. Planning with diffusion for flexible behavior synthesis. In *International Conference on Machine Learning*, pp. 9902–9915. PMLR, 2022.

Jiaming Ji, Borong Zhang, Jiayi Zhou, Xuehai Pan, Weidong Huang, Ruiyang Sun, Yiran Geng, Yifan Zhong, Josef Dai, and Yaodong Yang. Safety gymnasium: A unified safe reinforcement learning benchmark. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2023. URL `https://openreview.net/forum?id=WZmlxIuIGR`.

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

Aviral Kumar, Justin Fu, Matthew Soh, George Tucker, and Sergey Levine. Stabilizing off-policy q-learning via bootstrapping error reduction. *Advances in neural information processing systems*, 32, 2019.

Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative q-learning for offline reinforcement learning. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 1179–1191. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/0d2b2061826a5df3221116a5085a6052-Paper.pdf.

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

Jongmin Lee, Cosmin Paduraru, Daniel J Mankowitz, Nicolas Heess, Doina Precup, Kee-Eung Kim, and Arthur Guez. COptiDICE: Offline constrained reinforcement learning via stationary distribution correction estimation. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=FLA55mBee6Q.

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.

Jianxiong Li, Xianyuan Zhan, Haoran Xu, Xiangyu Zhu, Jingjing Liu, and Ya-Qin Zhang. When data geometry meets deep function: Generalizing offline reinforcement learning. In *The Eleventh International Conference on Learning Representations*, 2023.

Qian Lin, Bo Tang, Zifan Wu, Chao Yu, Shangqin Mao, Qianlong Xie, Xingxing Wang, and Dong Wang. Safe offline reinforcement learning with real-time budget constraints. In *International Conference on Machine Learning*, 2023.

Zuxin Liu, Zijian Guo, Yihang Yao, Zhepeng Cen, Wenhao Yu, Tingnan Zhang, and Ding Zhao. Constrained decision transformer for offline safe reinforcement learning. In *International Conference on Machine Learning*, 2023.

Zuxin Liu, Zijian Guo, Haohong Lin, Yihang Yao, Jiacheng Zhu, Zhepeng Cen, Hanjiang Hu, Wenhao Yu, Tingnan Zhang, Jie Tan, and Ding Zhao. Datasets and benchmarks for offline safe reinforcement learning. *Journal of Data-centric Machine Learning Research*, 2024.

Ian M. Mitchell. A toolbox of level set methods. In *A Toolbox of Level Set Methods*, 2005. URL https://api.semanticscholar.org/CorpusID:59892255.

Quan Nguyen and Koushil Sreenath. Safety-critical control for dynamical bipedal walking with precise footstep placement. *IFAC-PapersOnLine*, 48(27):147–154, 2015.

Alexander Robey, Haimin Hu, Lars Lindemann, Hanwen Zhang, Dimos V. Dimarogonas, Stephen Tu, and Nikolai Matni. Learning control barrier functions from expert demonstrations. In *2020 59th IEEE Conference on Decision and Control (CDC)*, pp. 3717–3724, 2020. doi: 10.1109/CDC42340.2020.9303785.

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 *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 11532–11539. IEEE, 2024.

Adam Stooke, Joshua Achiam, and Pieter Abbeel. Responsive safety in reinforcement learning by PID lagrangian methods. In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 9133–9143. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/stooke20a.html.

Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, Vijay Vasudevan, Wei Han, Jiquan Ngiam, Hang Zhao, Aleksei Timofeev, Scott Ettinger, Maxim Krivokon, Amy Gao, Aditya Joshi, Yu Zhang, Jonathon Shlens, Zhifeng Chen, and Dragomir Anguelov. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

Ihab Tabbara and Hussein Sibai. Learning neural control barrier functions from offline data with conservatism, 2025. URL https://arxiv.org/abs/2505.00908.

Manan Tayal, Rajpal Singh, Jishnu Keshavan, and Shishir Kolathaya. Control barrier functions in dynamic uavs for kinematic obstacle avoidance: A collision cone approach. In *2024 American Control Conference (ACC)*, pp. 3722–3727. IEEE, 2024a.

Manan Tayal, Hongchao Zhang, Pushpak Jagtap, Andrew Clark, and Shishir Kolathaya. Learning a formally verified control barrier function in stochastic environment. In *2024 IEEE 63rd Conference on Decision and Control (CDC)*, pp. 4098–4104, 2024b. doi: 10.1109/CDC56724.2024.10886052.

Manan Tayal, Aditya Singh, Shishir Kolathaya, and Somil Bansal. A physics-informed machine learning framework for safe and optimal control of autonomous systems. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=SrfwiloGQF.

Chen Tessler, Daniel J Mankowitz, and Shie Mannor. Reward constrained policy optimization. In *International Conference on Learning Representations*, 2018.

Guofan Wu and Koushil Sreenath. Safety-critical control of a planar quadrotor. In *2016 American Control Conference (ACC)*, pp. 2252–2258, 2016. doi: 10.1109/ACC.2016.7525253.

Haoran Xu, Xianyuan Zhan, and Xiangyu Zhu. Constraints penalized q-learning for safe offline reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 8753–8760, 2022.

Hongchao Zhang, Junlin Wu, Yevgeniy Vorobeychik, and Andrew Clark. Exact verification of relu neural control barrier functions. In A. Oh, T. Neumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 5685–5705. Curran Associates, Inc., 2023.

Qingye Zhao, Xin Chen, Zhuoyu Zhao, Yifan Zhang, Enyi Tang, and Xuandong Li. Verifying neural network controlled systems using neural networks. In *25th ACM International Conference on Hybrid Systems: Computation and Control*, pp. 1–11, 2022.

Weiye Zhao, Tairan He, Rui Chen, Tianhao Wei, and Changliu Liu. Safe reinforcement learning: A survey. *arXiv preprint arXiv:2302.03122*, 2023.

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 *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=cM95sT3gM3.

A Theoretical Insights

A.1 Finite Difference Barrier Condition

Lemma 1 (Finite-Difference Approximation of the Barrier Condition)**.** *Consider a control-affine system $\dot{x} = f(x) + g(x)u$ with a continuously differentiable barrier function $B : \mathcal{X} \to \mathbb{R}$. The continuous-time barrier condition*

$$\min\{\, \max_{u} \nabla B(x)^\top (f(x) + g(x)u),\ \ell(x) - B(x)\,\} = 0 \tag{14}$$

admits the finite-difference approximation

$$\min\{\, \max_{u_t} B(x_{t+1}),\ \ell(x_t)\,\} = B(x_t), \tag{15}$$

up to a positive scaling by Δt.

Proof. Starting from the continuous-time condition in equation 14, note that

$$\nabla B(x)^\top (f(x) + g(x)u) = \nabla B(x)^\top \dot{x} = \frac{d}{dt} B(x(t)). \tag{16}$$

Using a forward Euler approximation, the time derivative satisfies

$$\frac{d}{dt} B(x(t)) \approx \frac{B(x_{t+1}) - B(x_t)}{\Delta t}, \tag{17}$$

where, $x_t = x(t)$, $u_t = u(t)$ and $x_{t+1} = x(t + \Delta t)$. Substituting equation 17 into equation 14 yields

$$\min\left\{ \max_{u_t} \frac{B(x_{t+1}) - B(x_t)}{\Delta t},\ \ell(x_t) - B(x_t) \right\} = 0. \tag{18}$$

Let $a = B(x_{t+1}) - B(x_t)$ and $b = \ell(x_t) - B(x_t)$. Since $\Delta t > 0$, the function

$$m(a, b) = \min\{\, a/\Delta t,\ b\,\}$$

satisfies $m(a, b) = 0$ if and only if $\min\{a, b\} = 0$. This follows from the fact that scaling one argument of the minimum by a positive constant does not change which argument is smaller, nor the value at which the minimum equals zero. Thus, equation 18 is equivalent to

$$\min\{\, a,\ b\,\} = 0, \tag{19}$$

which gives the finite-difference barrier condition in equation 15. This establishes the claim. □

A.2 Expectile Regression

Expectile regression is a classical tool in statistics and econometrics for estimating asymmetric statistics of a random variable. For a random variable X, the τ-expectile is defined as the minimizer of the asymmetric least-squares problem

$$m_\tau = \arg\min_{m} \mathbb{E}_{x \sim X} \left[L^\tau (x - m) \right], \tag{20}$$

where $L_\tau(y) = |\tau - \mathbf{1}(y < 0)|, y^2$.

When $\tau > 0.5$, this loss assigns more weight to samples above the estimate m_τ and less weight to samples below it. Conversely, $\tau < 0.5$ emphasizes lower values. Thus, expectiles interpolate smoothly between the mean ($\tau = 0.5$) and a "high-value–seeking" statistic as $\tau \to 1$.

Expectile regression can also be extended to learning conditional expectiles:

$$m_\tau(x) = \arg\min_{m(\cdot)} \mathbb{E}_{(x,y) \sim \mathcal{D}} \left[L^\tau (y - m(x)) \right], \tag{21}$$

which can be optimized efficiently via stochastic gradient descent. This makes expectiles easy to implement in modern machine-learning pipelines, unlike alternative high-order statistics that require specialized solvers.

A.2.1 Why Expectile Regression in V-OCBF?

In V-OCBF, the barrier function ideally requires computing $\max_{u' \in \mathcal{U}} B(x')$, but performing this maximization directly is problematic in the offline setting because the dataset supports only a restricted action set $\mathcal{U}_{\mathcal{D}} = \{ u \mid (x, u, x') \in \mathcal{D} \}$, and querying $B(x')$ for unseen actions $u' \notin \mathcal{U}_{\mathcal{D}}$ introduces unsupported targets that can distort or destabilize learning. Moreover, even restricting the maximization to $\mathcal{U}_{\mathcal{D}}$ leads to a brittle update since a hard maximum is extremely sensitive to noise or rare outlier transitions, often producing poor barrier estimates. Expectile regression provides a smooth and data-supported alternative by emphasizing higher (safer) barrier values when $\tau > 0.5$, effectively approximating the maximal barrier value over $\mathcal{U}_{\mathcal{D}}$ while never querying out-of-distribution actions, thereby yielding a stable and principled mechanism for synthesising offline CBVF.

Expectile regression resolves these issues. For $\tau > 0.5$, the expectile estimator concentrates on the largest barrier values consistent with the dataset. This yields a smooth approximation to $\max_{u' \in \mathcal{U}_{\mathcal{D}}} B(x')$, which is equivalent to performing a constrained maximization over only the dataset-supported actions, without ever querying B on out-of-distribution actions.

B Description of the Experiments

B.1 Autonomous Ground Vehicle Collision Avoidance

We consider a collision-avoidance task for an autonomous ground vehicle modeled using the Dubins dynamics:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{\Phi} \end{bmatrix} = \begin{bmatrix} v \cos \Phi \\ v \sin \Phi \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u, \tag{22}$$

where the state $x = [x_1, x_2, \phi]^T \in \mathcal{X} \subseteq \mathbb{R}^3$ encodes the vehicle's position and heading. The forward velocity is fixed at $v = 0.6$ units/s, and the admissible state space is given by $\mathcal{X} = [-1, 1]^2 \times [-\pi, \pi]$. The control input $u \in [-1, 1]$ denotes the angular velocity ω.

The reward function $r(x)$ is defined as

$$r(x) = \frac{C}{\left(\left\| \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} - \begin{bmatrix} x_{g1} \\ x_{g2} \end{bmatrix} \right\| + \epsilon \right)} \tag{23}$$

where $C = 0.1$ is the scaling factor and $(x_{g1}, x_{g2})^T$ denotes the goal location. Rewards are accumulated only while the agent remains collision-free. The constant $\epsilon > 0$ is included to prevent numerical instability when the vehicle approaches the goal.

Safety is encoded through the function

$$\ell(x) = \left\| \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} - \begin{bmatrix} x_{o1} \\ x_{o2} \end{bmatrix} \right\| - R, \tag{24}$$

where $(x_{o1}, x_{o2})^T = (0, 0)^T$ denotes the center of the obstacle and $R = 0.2$ units is the radius of the obstacle.

Offline Data Generation. Since this is a custom environment, we construct an offline dataset for training both the barrier function and the safe policy. We sample 1500 initial states uniformly from $\mathcal{X}$ and simulate each trajectory for 500 discrete timesteps with step size $\Delta t = 0.01s$. During data collection, control inputs are drawn uniformly at random from the admissible range, ensuring diverse system trajectories for learning-based safety certification.

B.1.1 Safe Set Volume Analysis

Backward Reachable Tube is the set of initial states for which the agent acting optimally, will eventually reach the target set $\mathcal{T}$ within the time horizon $[0, t]$:

$$SSV(x) = \{x : \exists u(\cdot), \exists t \in [0, T], \xi^u_{x,0}(t) \in \mathcal{T}\} \tag{25}$$

where $\xi^u_{x,0}(t)$ represents system state at time $t \in [0, T]$, and u defined over time horizon $[0, T]$, represents sequence of control actions over the time horizon.

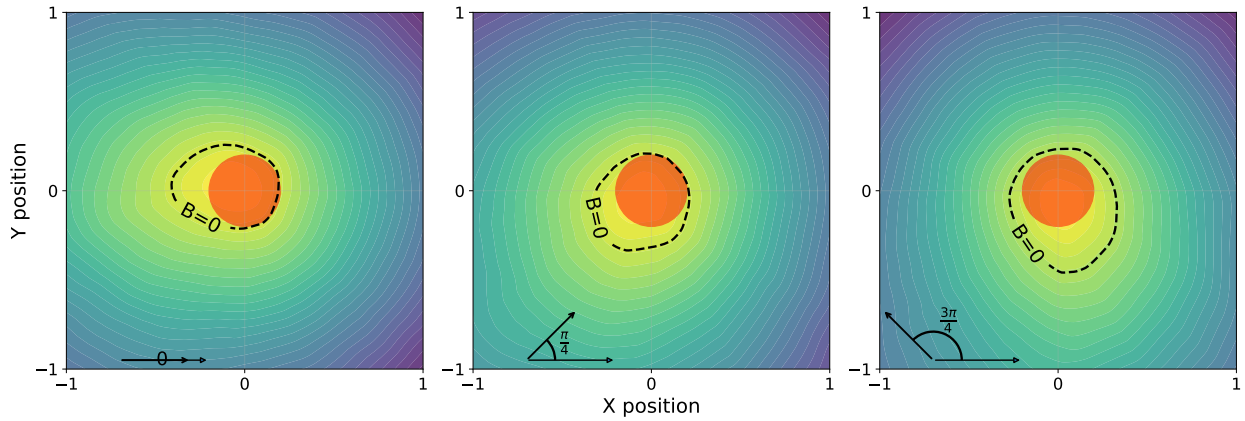

Figure 5: **Safe Set Volume:** Figure shows a depiction of feasible region (i.e., anything outside B=0 enclosure) for V-OCBF for 3 different heading angles. Note that calculation of the actual Safe Set volume is a cumulative of feasible region over countless such frames along the continuous heading angle (ϕ) dimension.

We performed Safe Set Volume (SSV) analysis to inspect the effectiveness and conservatism of various approaches. Specifically, we aim to evaluate the volume of the state space from which the agent can safely initialize. Effectively, the larger this Safe Set volume is, bigger the feasible region gets, hence, making it easy to navigate in the environment, and smaller the Safe Set volume becomes, lesser the feasible region gets, thus, overall making it less safe for the agent to navigate in the environment.

To evaluate the safe initialization region that each of the method casts, we uniformly sample different initial states across the state space and check whether they belong to the above-defined Safe Set Volume by rolling out trajectories in the online environment for over 100k times (as we started to observe convergence in the Safe Set Volume after the 100k mark). We investigate the results for this analysis in the main paper (refer Table 1) where we show that V-OCBF enables the largest safe feasible region among all the various approaches.

B.2 Safety MuJoCo Environments:

To evaluate our framework on higher-dimensional systems, we use MuJoCo Safety Gymnasium environments Ji et al. (2023), where safe velocity tasks introduce velocity constraints for Gymnasium's MuJoCo-v4 agents.

The agent receives a scalar cost signal on the basis of its velocity at each step:

$$cost = bool(V_{current} > V_{threshold}) \tag{26}$$

This constraint can be framed as a safety function

$$\ell(x) = V_{threshold} - V_{current} \tag{27}$$

since $\ell(x) \geq 0$ corresponds to safe state. This safety value function provides us a continuous signal instead of sparse values of 0 or 1. For agent specific threshold velocities and time-step Δt, we refer the official documentation, the official values for which have been compiled and included in Table 3.

Table 3: Threshold velocity $V_{threshold}$, time step size Δt and action space $\mathbb{U}$ for each MuJoCo based Safety Gymnasium Ji et al. (2023) environment as per official documentation.

Environment	**Hopper**	**Half-Cheetah**	**Ant**	**Walker2D**	**Swimmer**
$\boldsymbol{V_{threshold}}$	0.7402	3.2096	2.6222	2.3415	0.2282
$\boldsymbol{\Delta t}$	0.008	0.05	0.05	0.008	0.04
$\mathbb{U}$	$[-1.0, 1.0]^3$	$[-1.0, 1.0]^6$	$[-1.0, 1.0]^8$	$[-1.0, 1.0]^6$	$[-1.0, 1.0]^2$

B.2.1 Qualitative Analysis

In continuation to the Hopper plot in the main paper (Fig. 4), we include plots for all the remaining MuJoCo safety gymnasium environments covered in this paper with the visualization of the episode rollouts, both with and without V-OCBF. Given the safety rates of V-OCBF compared to the other baselines (refer Fig. 3), it becomes important to visually realize the real-time effect on performance (Figures 6, 7, 8).

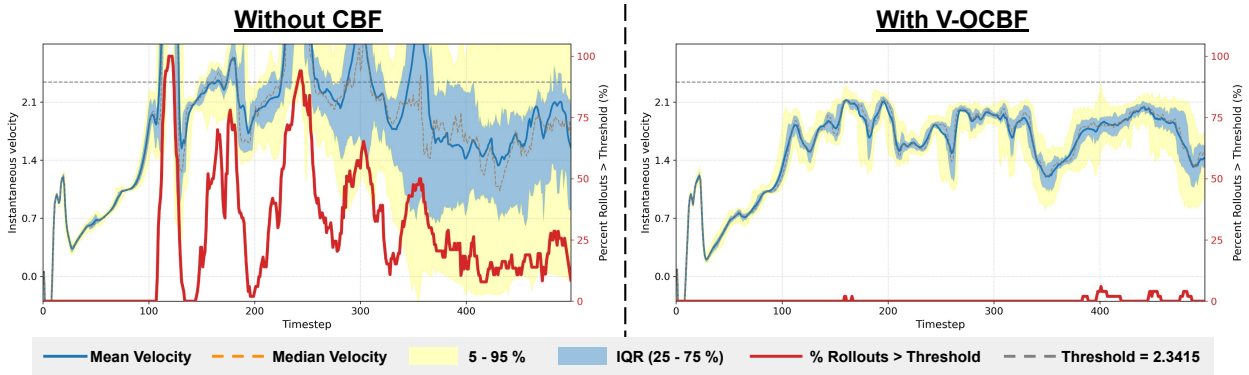

Figure 6: **Walker-2D.** Instantaneous velocity (left Y-axis) across episodes. Right Y-axis depicts percentage of episodes when the agent violates the velocity threshold at a particular timestep.

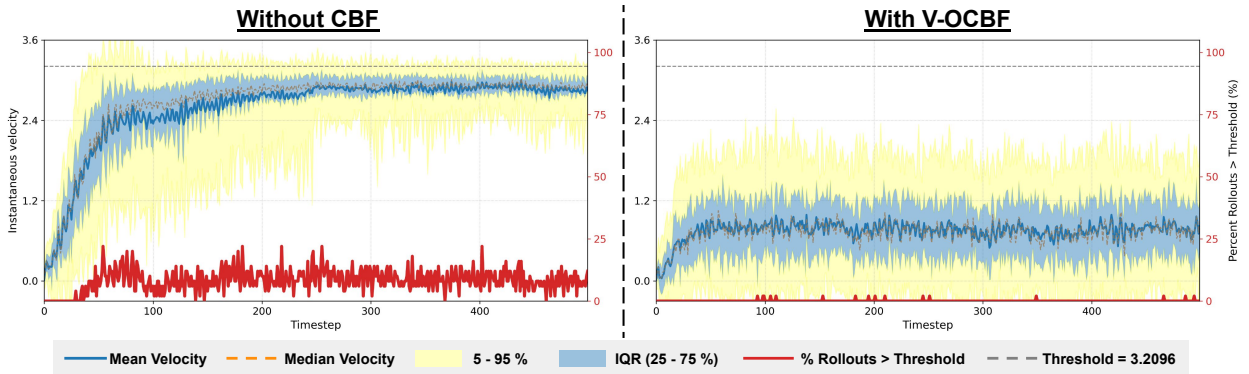

Figure 7: **Half-Cheetah.** Instantaneous velocity (left Y-axis) across episodes. Right Y-axis depicts percentage of episodes when the agent violates the velocity threshold at a particular timestep.

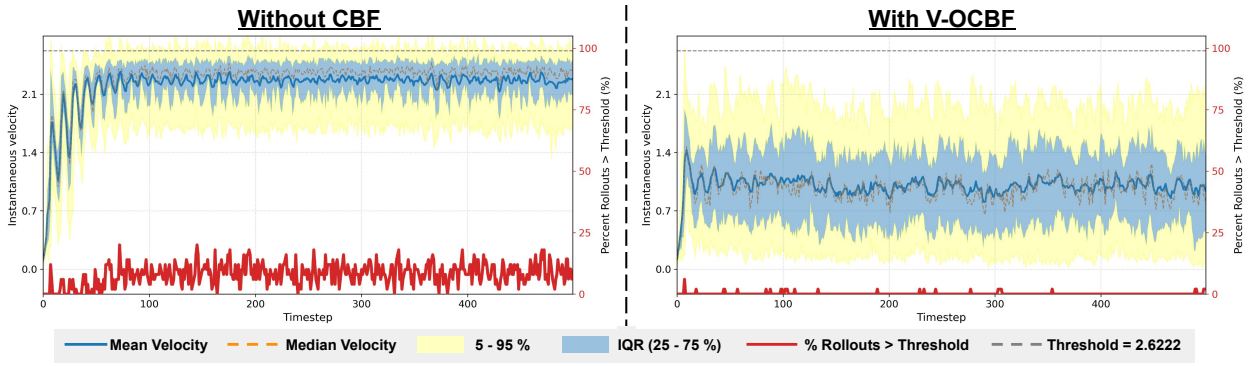

Figure 8: **Ant.** Instantaneous velocity (left Y-axis) across episodes. Right Y-axis depicts percentage of episodes when the agent violates the velocity threshold at a particular timestep.

C Ablation Studies

We have performed a set of ablation studies on our primary testing environment of Autonomous Ground Vehicle Collision Avoidance to further strengthen our claims of the paper. AGV acts as an ideal testing bed for such ablations as it is easy to visualize and infer from the results in a 2D space. Besides, the environment properly encapsulates our requirements of providing a safety critical control environment. Following subsections cover the ablations we have performed.

C.1 Robustness to size of the NN

In this section we show our framework's robustness against varying model sizes, thus, demonstrating the versatility on different hyperparameter choices. Table 4 shows the safety rate for different model dimensions.

Table 4: Safety Rate (%) reported for various model dimensions. Columns represent Width: neurons in each hidden layer, and Rows represent number of layers.

Width → **Layers ↓**	**256** **Neurons**	**96** **Neurons**	**64** **Neurons**	**48** **Neurons**
2	**98.28 ± 0.54**	97.18 ± 0.41	96.96 ± 1.21	–
3	98.04 ± 0.30	97.36 ± 0.22	97.12 ± 0.23	96.54 ± 1.62
4	–	–	–	96.82 ± 1.24

As can be seen from the Table 4, the models don't affect the safety performance significantly, until we drop the size by too much (as in the cases of **(48,3)**, **(48,4)** and **(64,2)**). We consider the subtle drop is safety rate for the lower dimensional networks to be within considerable limits.

C.2 Robustness to errors in Learned Dynamics

To evaluate the robustness of our pipeline to errors in the learned dynamics, we measure how additive perturbations affect the safety performance of the CBF-based controllers. Concretely, we perturb the computed safe action ω by adding zero-mean Gaussian noise $\mathcal{N}(0, \sigma^2)$ with a range of standard deviations σ (refer 5). For every noise level we run 500 independent trials (with 5 different random seeds each) and report the safety rate defined as the fraction of trials that remain within the safe set over the evaluation horizon. Table 5 summarizes the results across several CBF variants, for each method we report mean safety rate ± standard error. Across noise levels spanning small to large perturbations, the safety rates degrade only mildly, indicating that the learned CBFs tolerate moderate inaccuracies in the dynamics / control signal. These results suggest that, while precise dynamics models improve performance, our approach provides substantially safety-aware control even when the dynamics model is imperfect.

Table 5: Safety Rate (%) reported for Robustness Testing of **learned Dynamics** for Autonomous Ground Vehicle by adding upto 5%, 10% and 20% of the possible control action as noise.

Method	**No Noise**	**+ Small Noise (5%)**	**+ Medium Noise (10%)**	**+ Large Noise (20%)**
BC+NCBF Robey et al. (2020)	92.48 ± 0.60	92.48 ± 0.62	92.52 ± 0.41	92.32 ± 0.56
BC+iDBF Castañeda et al. (2023)	92.87 ± 0.73	92.84 ± 0.64	92.40 ± 0.84	92.76 ± 0.28
BC+CCBF Tabbara & Sibai (2025)	93.56 ± 0.56	93.72 ± 0.58	93.68 ± 0.80	93.76 ± 0.38
BC+**V-OCBF** (Ours)	**98.28 ± 0.54**	**98.08 ± 0.61**	**97.84 ± 0.68**	**97.94 ± 0.48**

This empirical test shows that even if we haven't learned the most accurate dynamics model, we can expect a considerably precise safety-aware control for the system with our approach.

C.2.1 Sensitivity to Expectile parameter (τ)

As discussed in Section 4.2, we use a τ-expectile loss to obtain the final CBVF. When $\tau = 0.5$, the objective reduces to a symmetric MSE fit, causing the learned barrier to reflect the average safety value induced by the behavior policy. This reproduces the *behavior-induced* barrier and typically yields overly conservative safe sets that do not generalize well beyond the demonstrated trajectories.

To investigate this effect, we conduct a study by comparing $\tau = 0.5$ against higher expectile levels that emphasize the upper tail of the safety value distribution. Table 6 summarises the safety rates achieved on the AGV environment for different τ values:

Table 6: Safety Rate (%) reported for Autonomous Ground Vehicle by varying τ in the expectile regression.

τ	**0.5 (Behavior Policy)**	**0.7**	**0.8**	**0.9**	**0.99**
Safety Rate (%)	96.53 ± 0.23	96.87 ± 0.54	97.16 ± 1.01	97.72 ± 0.88	**98.28 ± 0.54**

The results reveal two key patterns. First, the behavior-induced barrier ($\tau = 0.5$) yields the lowest safety rate, consistent with its tendency to underrepresent safety-critical regions that are not frequently explored by the behavior policy. Second, as τ increases, the barrier becomes progressively more conservative in safety-critical areas and better aligned with the upper end of the demonstrated safety values, resulting in consistently higher safety rates.

The best performance is achieved near $\tau \to 1$, indicating that prioritizing the high-value portion of the demonstration distribution leads to safer barrier estimates and improved closed-loop performance.

D Experimental Details

D.1 Experimental Hardware

To keep the evaluation fair and avoid any discriminatory added advantage to any specific experiment, all experiments were conducted on a single system equipped with an 14th Gen Intel Core i9-14900KS CPU, 128GB RAM, and an NVIDIA GeForce RTX 5090 GPU for training and experiment evaluations.

D.2 Hyperparameters for the Proposed Algorithm

We have compiled and listed down all the hyperparameters that we used to perform our experiments and report the results. These training settings for all the environments are detailed in the Table 7. For the MuJoCo environments, we use the widely accepted DSRL Liu et al. (2024) dataset. We also use the *Class-K* function $\kappa = \alpha \times B(\cdot)$ where $\alpha = 1$.

Table 7: Hyperparameters for the Proposed (V-OCBF) Network, and Learned Dynamics Model.

Hyperparameter	**Value**
Network Architecture	Multi-Layer Perceptron (MLP)
Activation Function	ReLU
Optimizer	Adam optimizer
Learning Rate	3×10^{-5}
Discount Factor (γ)	0.99
Autonomous Ground Vehicle Collision Avoidance	
Number of Hidden Layers	2
Hidden Layer Size	256 neurons per layer
Dataset Size	75K
Dynamics Model Hidden Layers	3
Dynamics Model Hidden Layer Size	64
Safe Velocity Hopper	
Number of Hidden Layers	3
Hidden Layer Size	256 neurons per layer
Dataset Size	1.32M
Dynamics Model Hidden Layers	4
Dynamics Model Hidden Layer Size	64
Safe Velocity Half-Cheetah	
Number of Hidden Layers	3
Hidden Layer Size	128 neurons per layer
Dataset Size	249K
Dynamics Model Hidden Layers	4
Dynamics Model Hidden Layer Size	64
Safe Velocity Ant	
Number of Hidden Layers	3
Hidden Layer Size	256 neurons per layer
Dataset Size	2.09M
Dynamics Model Hidden Layers	4
Dynamics Model Hidden Layer Size	64
Safe Velocity Walker2D	
Number of Hidden Layers	3
Hidden Layer Size	256 neurons per layer
Dataset Size	2.12M
Dynamics Model Hidden Layers	4
Dynamics Model Hidden Layer Size	64
Safe Velocity Swimmer	
Number of Hidden Layers	3
Hidden Layer Size	256 neurons per layer
Dataset Size	1.68M
Dynamics Model Hidden Layers	4
Dynamics Model Hidden Layer Size	64

D.3 Hyperparameters for the Baselines

For the CBF-based baselines (NCBF, iDBF, CCBF), we use the same network architectures and learned dynamics models described above to ensure a fair comparison. Hyperparameters for the remaining safe offline RL baselines (COptiDICE, BEAR-Lag, BC, FISOR) are provided in Table 8. We use the official implementations of BEAR-Lag and COptiDICE from Liu et al. (2024) and of FISOR from Zheng et al. (2024), and train all methods on the same DSRL datasets.

Table 8: Hyperparameters for all the Baselines network. Optimizer: Adam, LR: 1×10^{-3}, Activation: ReLU (applies to all rows). For COptiDICE and BEAR-Lag refer OSRL, Liu et al. (2024), for the official implementations. For FISOR refer Zheng et al. (2024) for official implementation.

Autonomous Ground Vehicle (AGV)	
BC - Actor	MLPActor, (128,2)
BEAR-Lag - Actor	SquashedGaussianMLPActor, (128,2)
BEAR-Lag - Critic	EnsembleDoubleQCritic, (256,2)
BEAR-Lag - Cost critic	MLP, (256,2)
COptiDICE - Actor (extraction)	MLPActor, (128,2)
COptiDICE - Dual / ν network	DualNet (value-like), (256,2)
FISOR - Actor (diffusion denoiser backbone)	DiffusionDenoiserMLP, (128,2)
FISOR - Feasibility classifier	FeasibilityClassifier, (256,2)
Safe Velocity Hopper	
BC - Actor	MLPActor, (256,2)
BEAR-Lag - Actor	SquashedGaussianMLPActor, (256,2)
BEAR-Lag - Critic	EnsembleDoubleQCritic, (256,2)
BEAR-Lag - Cost critic	MLP, (256,2)
COptiDICE - Actor (extraction)	MLPActor, (256,2)
COptiDICE - Dual / ν network	DualNet (value-like), (256,2)
FISOR - Actor (diffusion denoiser backbone)	DiffusionDenoiserMLP, (256,2)
FISOR - Feasibility classifier	FeasibilityClassifier, (256,2)
Safe Velocity Half-Cheetah	
BC - Actor	MLPActor, (256,2)
BEAR-Lag - Actor	SquashedGaussianMLPActor, (256,2)
BEAR-Lag - Critic	EnsembleDoubleQCritic, (256,2)
BEAR-Lag - Cost critic	MLP, (256,2)
COptiDICE - Actor (extraction)	MLPActor, (256,2)
COptiDICE - Dual / ν network	DualNet (value-like), (256,2)
FISOR - Actor (diffusion denoiser backbone)	DiffusionDenoiserMLP, (256,2)
FISOR - Feasibility classifier	FeasibilityClassifier, (256,2)
Safe Velocity Ant	
BC - Actor	MLPActor, (256,2)
BEAR-Lag - Actor	SquashedGaussianMLPActor, (256,2)
BEAR-Lag - Critic	EnsembleDoubleQCritic, (256,2)
BEAR-Lag - Cost critic	MLP, (256,2)
COptiDICE - Actor (extraction)	MLPActor, (256,2)
COptiDICE - Dual / ν network	DualNet (value-like), (256,2)
FISOR - Actor (diffusion denoiser backbone)	DiffusionDenoiserMLP, (256,2)
FISOR - Feasibility classifier	FeasibilityClassifier, (256,2)
Safe Velocity Walker2D	
BC - Actor	MLPActor, (256,2)
BEAR-Lag - Actor	SquashedGaussianMLPActor, (256,2)
BEAR-Lag - Critic	EnsembleDoubleQCritic, (256,2)
BEAR-Lag - Cost critic	MLP, (256,2)
COptiDICE - Actor (extraction)	MLPActor, (256,2)
COptiDICE - Dual / ν network	DualNet (value-like), (256,2)
FISOR - Actor (diffusion denoiser backbone)	DiffusionDenoiserMLP, (256,2)
FISOR - Feasibility classifier	FeasibilityClassifier, (256,2)
Safe Velocity Swimmer	
BC - Actor	MLPActor, (256,2)
BEAR-Lag - Actor	SquashedGaussianMLPActor, (256,2)
BEAR-Lag - Critic	EnsembleDoubleQCritic, (256,2)
BEAR-Lag - Cost critic	MLP, (256,2)
COptiDICE - Actor (extraction)	MLPActor, (256,2)
COptiDICE - Dual / ν network	DualNet (value-like), (256,2)
FISOR - Actor (diffusion denoiser backbone)	DiffusionDenoiserMLP, (256,2)
FISOR - Feasibility classifier	FeasibilityClassifier, (256,2)