

Deep QP Safety Filter: Model-free Learning for Reachability-based Safety Filter

Byeongjun Kim

Seoul National University

QUDWNS3456@SNU.AC.KR

H. Jin Kim

Seoul National University

HJINKIM@SNU.AC.KR

Editors: G. Sukhatme, L. Lindemann, S. Tu, A. Wierman, N. Atanasov

Abstract

We introduce Deep QP Safety Filter, a fully data-driven safety layer for black-box dynamical systems. Our method learns a Quadratic-Program (QP) safety filter without model knowledge by combining Hamilton–Jacobi (HJ) reachability with model-free learning. We construct contraction-based losses for both the safety value and its derivatives, and train two neural networks accordingly. In the exact setting, the learned critic converges to the viscosity solution (and its derivative), even for non-smooth values. Across diverse dynamical systems – even including a hybrid system – and multiple RL tasks, Deep QP Safety Filter substantially reduces pre-convergence failures while accelerating learning toward higher returns than strong baselines, offering a principled and practical route to safe, model-free control.

Keywords: Safety filter, Hamilton-Jacobi reachability, Model-free learning, Safe reinforcement learning

1. Introduction

Safety-critical control has advanced rapidly in recent years, revealing impressive capabilities across robotics and autonomous systems. Among these, filtering-based approaches that modify an unsafe reference input into a safe one have become popular for their simplicity and real-time feasibility (Hsu et al., 2023a). Within this family, Quadratic-Program (QP) formulations that enforce safety even inside the safe set often produce smoother behavior than switching-based methods, and have been deployed on challenging, unstable platforms such as bipedal robots (Liu et al., 2023).

Despite this progress, current filters depend heavily on prior model knowledge: their reliability deteriorates as model uncertainty grows. Recent learning-based variants attempt to mitigate uncertainty, yet many still presume partial access to the dynamics, which is unavailable for black-box systems. This creates a practical gap between elegant theory and deployment on real platforms with unknown or changing dynamics.

We address this gap by proposing a framework that learns a QP-based safety filter directly from data, requiring no model knowledge. Our method integrates time-discounted Hamilton–Jacobi (HJ) reachability with a model-free learning paradigm to construct a filter that preserves the desirable control characteristics of QP methods while enforcing data-driven safety constraints. Theoretical guarantees follow from the contraction properties of Bellman operators, yielding convergence and enabling a practical, model-free implementation.

We validate the approach in extensive simulations. The learned critic converges to the analytic HJ solution, and, when used as a safety filter, significantly reduces training failures and improves RL policy performance relative to baselines. The filter’s conservatism is reduced via HJ reachability and can be tuned in aggressiveness by the user, allowing reuse across tasks. Finally, we further test the method on a hybrid system that does not meet the theoretical assumptions. Its strong performance indicates that the assumptions are practically valid when approximated by neural networks.

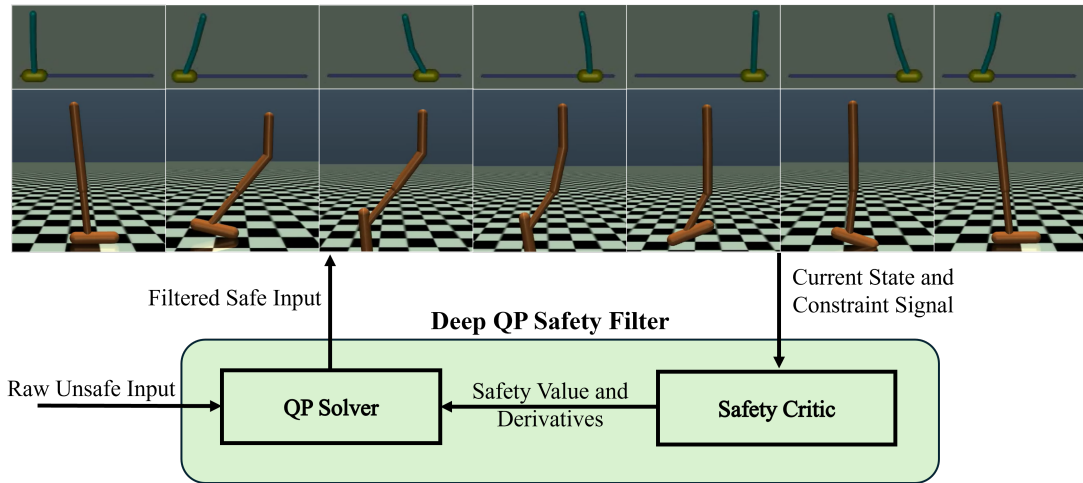


Figure 1: Overview of the Deep QP Safety Filter learned purely from transition data. The Safety Critic maps state and constraints to a safety value and derivatives, used by a QP solver together with a raw reference input, in order to produce a safe control input. The example sequence illustrates an unstable system with bang-bang (± 1) reference commands, where the filtered commands under our method maintain safety.

2. Related Works

Learning with model assumptions. A large body of research uses learning to overcome the limitations of model-based methods, such as the curse of dimensionality in HJ reachability (Bansal and Tomlin, 2021; Hsu et al., 2023b; Li et al., 2025) or the difficulty of handcrafting Control Barrier Functions (CBFs) (Robey et al., 2020; Xiao et al., 2023; Zhang et al., 2025). Other works combine CBF-QPs with RL to compensate for imperfect nominal models (Gangopadhyay et al., 2022; Choi et al., 2020). However, these approaches still rely on partial model knowledge, such as a pre-defined CBF structure (Taylor et al., 2020), properties like Lipschitz constants (Choi et al., 2025), or a precomputed safe policy (So et al., 2024) – assumptions that are rarely available in the black-box settings.

Model-free safety learning. Inspired by RL, model-free safety learning has also been widely explored. Many of these methods rely on discounted cumulative costs (Gu et al., 2024), which do not ensure persistent satisfaction of state constraints. Others aim for model-free state constraint satisfaction but may incur numerous safety violations during training (Yu et al., 2022) or produce switching filters (Fisac et al., 2019). This switching behavior is unsuitable for highly dynamic systems, where such rapidly changing safe control strategies are not preferred. More recent work learns discriminating hyperplanes in the input space (Lavanakul et al., 2024). However, for black-box systems without prior safety knowledge, one must adopt the RL-based training approach of this method. Since this RL approach simply maximizes survival time, the absence of reachability foundations may lead to conservative behavior.

3. Method

In this section, we describe the overall setting of our problem and introduce the HJ reachability as a preliminary requirement for the proposed method. We then propose a model-free safety filter learning framework grounded on the HJ reachability.

3.1. Problem Setup

Let us denote the state and the control input as $x \in \mathcal{X} \subseteq \mathbb{R}^n$ and $u \in \mathcal{U} \subset \mathbb{R}^m$, respectively, where \mathcal{X} and \mathcal{U} represent the state space and the input space. Since most actuators are subject to hardware limits, we consider the input space \mathcal{U} to be constrained by a known pair of (A, b) , i.e.,

$$\mathcal{U} = \{u \in \mathbb{R}^m \mid Au \leq b\}.$$

This also includes the common box-constrained case in which each actuator independently possesses its own upper and lower bounds.

The system provides a differentiable output signal $c(x) \in \mathbb{R}$ whose functional form is unknown, although its value can be directly measured. We impose the state constraint $c(x) \geq 0$. This constraint can be viewed as an observable signal analogous to a reward signal in RL. Observations are made at a known fixed time interval $\delta t > 0$. Since sensing and control are performed discretely in real systems, we use transition data sampled at the same fixed interval δt . That is, given a particular state x , applied control input u , observed constraint signal c , and the next state and constraint $x_{\delta t}^u$ and $c_{\delta t}^u$ after δt , we form transitions of the type $(x, u, c, x_{\delta t}^u, c_{\delta t}^u)$.

We assume the control-affine dynamics of the given black-box system:

$$\dot{x} = f(x) + g(x)u.$$

This assumption is well justified, as many physical systems are controlled by force or torque commands through actuators. In such cases, the Euler–Lagrange equation naturally leads to a control-affine form. Even if a system is not strictly control-affine, the structure can still be represented by introducing a virtual input v with $\dot{u} = v$. Moreover, as will be discussed later, even a highly dynamic hybrid system exhibiting discontinuous state transitions can be effectively learned in a smooth manner through discrete transition data. The proposed algorithm has been empirically verified to handle such hybrid cases successfully.

Lastly, we define the notion of safety considered throughout this work.

Definition 1 (Safe state and safe set) *A state x is said to be safe if there exists a control trajectory such that the state at time t , $\mathbf{x}(t)$, driven by that control from initial condition $\mathbf{x}(0) = x$, satisfies $c(\mathbf{x}(t)) \geq 0$ for all $t \geq 0$. A set S is said to be a safe set if every element $x \in S$ is a safe state.*

Our goal is to design a QP-based safety filter for systems with black-box dynamics. Unlike bang–bang safety filters that intervene only when the safety value falls below a threshold, we aim for a formulation that consistently filters the control input. This approach provides smoother control commands and more stable system behavior while preserving safety guarantees.

3.2. Hamilton-Jacobi Reachability

Hamilton–Jacobi (HJ) reachability has been one of the cornerstones in the field of safety-critical control. It is an optimal-control-based framework that handles input bounds and relative degrees between constraint and control, enabling safe control with minimal conservatism. Let us consider the safety value function and its corresponding Partial Differential Equation (PDE) (Fisac et al., 2019, 2015):

$$V(x) = \sup_{\pi} \inf_{t \geq 0} c(x_t^{\pi}), \quad \min \left\{ c(x) - V(x), \max_{u \in \mathcal{U}} \frac{\partial V}{\partial x}(x)(f(x) + g(x)u) \right\} = 0.$$

where π denotes any control policy and x_t^{π} is the state at time t driven by policy π starting from x . The function $V(x)$ represents the smallest achievable constraint value when the system is controlled

optimally from the current state x . By definition, the current state x is considered *safe* if and only if $V(x) \geq 0$, and the 0-superlevel set of $V(x)$ renders the largest control invariant safe set.

To enable model-free learning, we rewrite the definition of $V(x)$ in a recursive form:

$$V(x) = \sup_{\pi} \min \left\{ \min_{0 \leq t_1 \leq s} c(x_{t_1}^{\pi}), \inf_{t_2 \geq s} c(x_{t_2}^{\pi}) \right\} = \sup_{\pi} \min \left\{ \min_{0 \leq t \leq s} c(x_t^{\pi}), V(x_s^{\pi}) \right\} \quad (1)$$

However, (1) requires transition data to be collected under an optimally safe policy. Since such a policy is not known in advance, this formulation is not directly suitable for model-free learning. To relax this limitation and allow learning from arbitrary transition data, we introduce a state-action safety value function analogous to the Q -function in RL (Watkins and Dayan, 1992):

$$Q_s(x, u) = \min \left\{ \min_{0 \leq t_1 \leq s} c(x_{t_1}^u), \sup_{\pi} \inf_{t_2 \geq 0} c((x_s^u)^{\pi}_{t_2}) \right\} = \min \left\{ \min_{0 \leq t \leq s} c(x_t^u), V(x_s^u) \right\} \quad (2)$$

Here, $Q_s(x, u)$ represents the smallest constraint value that can be achieved when applying a constant control u for a short horizon $s \geq 0$ and then switching to the optimally safe policy thereafter. This formulation serves as the foundation for a model-free learning algorithm that can learn directly from arbitrary transition data.

3.3. Time-discounted reachability

The recursive definitions in (1) and (2) do not inherently possess contraction properties, which makes them unsuitable for stable critic learning. To overcome this limitation, several works (Fisac et al., 2019; Li et al., 2025) introduced a discount factor in the discrete-time domain. In this section, we extend this concept to the continuous-time setting and establish the theoretical foundations required to learn the corresponding discounted value functions and their derivatives. First, for a non-negative scalar $\lambda \geq 0$, we define the discounted value functions $V^{\lambda}(x)$ and $Q_s^{\lambda}(x, u)$ as

$$V^{\lambda}(x) := \sup_{\pi} \inf_{t \geq 0} \left\{ \int_0^t \lambda e^{-\lambda \tau} c(x_{\tau}^{\pi}) d\tau + e^{-\lambda t} c(x_t^{\pi}) \right\}, \quad (3)$$

$$Q_s^{\lambda}(x, u) := \min \left\{ \min_{0 \leq t \leq s} \int_0^t \lambda e^{-\lambda \tau} c(x_{\tau}^u) d\tau + e^{-\lambda t} c(x_t^u), \int_0^s \lambda e^{-\lambda \tau} c(x_{\tau}^u) d\tau + e^{-\lambda s} V^{\lambda}(x_s^u) \right\} \quad (4)$$

We list several corollaries relating the original and discounted value functions.

Corollary 1 *For all $(x, u) \in \mathcal{X} \times \mathcal{U}$, the following inequalities hold:*

$$Q_s(x, u) \leq V(x) \leq c(x), \quad Q_s^{\lambda}(x, u) \leq V^{\lambda}(x) \leq c(x).$$

Corollary 2 *As the discount factor approaches zero, the discounted functions converge to their undiscounted counterparts:*

$$\lim_{\lambda \downarrow 0} V^{\lambda}(x) = V(x), \quad \lim_{\lambda \downarrow 0} Q_s^{\lambda}(x, u) = Q_s(x, u), \quad \forall (x, u) \in \mathcal{X} \times \mathcal{U}.$$

Corollary 3 *As the duration s tends to zero, the state-action safety values converge to their corresponding safety values:*

$$\lim_{s \downarrow 0} Q_s(x, u) = V(x), \quad \lim_{s \downarrow 0} Q_s^{\lambda}(x, u) = V^{\lambda}(x), \quad \forall (x, u) \in \mathcal{X} \times \mathcal{U}.$$

Proof It directly follows from the definitions in (1) and (2) that $V(x) \leq c(x)$ and $V^{\lambda}(x) \leq c(x)$ for all $x \in \mathcal{X}$. Because $V^{\lambda}(x)$ is defined as the supremum over all admissible control, $Q_s(x, u) \leq V(x)$

and $Q_s^\lambda(x, u) \leq V^\lambda(x)$ for all $(x, u) \in \mathcal{X} \times \mathcal{U}$. These prove the first corollary. The second and third corollaries are obtained by substituting $\lambda = 0$ and $s = 0$ into (3)-(4) and (2)-(4). \blacksquare

These results serve as the starting point for constructing our model-free safety filter learning algorithm. For further analysis, we consider the following HJ PDE for $V^\lambda(x)$.

Lemma 1 (PDE for the discounted safety value function) *The value function in (3) satisfies the following PDE in the viscosity sense (So et al., 2024), (Altarovici et al., 2013):*

$$\min \left\{ c(x) - V^\lambda(x), \max_{u \in \mathcal{U}} \frac{\partial V}{\partial x}(x)(f(x) + g(x)u) + \lambda \left(c(x) - V^\lambda(x) \right) \right\} = 0 \quad (5)$$

The PDE in **Lemma 1** characterizes the local optimality condition of the discounted safety value function. To make this relation explicit in terms of the control input u , we next introduce an auxiliary quantity that measures the instantaneous change of $V^\lambda(x)$ along the system dynamics. This quantity, referred to as the advantage of discounted safety value, plays a key role in defining the model-free Bellman operator used in our learning framework. To facilitate further analysis, we assume that $V^\lambda(x)$ is differentiable. This assumption not only simplifies the subsequent derivations but is also not restrictive in practice, as most neural network approximators employ smooth activation functions such as \tanh or the Exponential Linear Unit (ELU) (Clevert et al., 2015).

Theorem 1 (Advantage in discounted safety value) *Let us define the advantage $q^\lambda(x, u) := \lim_{s \downarrow 0} \frac{Q_s^\lambda(x, u) - V^\lambda(x)}{s}$. Then, q^λ satisfies the following:*

$$q^\lambda(x, u) \leq 0 \quad \forall (x, u) \in \mathcal{X} \times \mathcal{U}, \quad \max_{u \in \mathcal{U}} q^\lambda(x, u) = 0 \quad \forall x \in \mathcal{X},$$

$$q^\lambda(x, u) = \min \left\{ c(x) - V^\lambda(x), \frac{\partial V^\lambda}{\partial x}(x)(f(x) + g(x)u) + \lambda \left(c(x) - V^\lambda(x) \right) \right\}.$$

Proof Expanding each term inside the min of (4) with a sufficiently small s ,

$$\begin{aligned} \min_{0 \leq t \leq s} \int_0^t \lambda e^{-\lambda \tau} c(x_\tau^u) d\tau + e^{-\lambda t} c(x_t^u) &= c(x) + s \min \left\{ 0, \frac{\partial c}{\partial x}(x)(f(x) + g(x)u) \right\} + O(s^2) \\ \int_0^s \lambda e^{-\lambda \tau} c(x_\tau^u) d\tau + e^{-\lambda s} V^\lambda(x_s^u) &= V^\lambda(x) + s\lambda \left(c(x) - V^\lambda(x) \right) \\ &\quad + s \frac{\partial V^\lambda}{\partial x}(x)(f(x) + g(x)u) + O(s^2). \end{aligned}$$

Now, we subtract $V^\lambda(x)$, divide s , and take $s \downarrow 0$. For the ease of further analysis, we separately consider the two cases: 1) $V^\lambda(x) = c(x)$ and 2) $V^\lambda(x) < c(x)$. For the case of 1), we have the following:

$$q^\lambda(x, u) = \min \left\{ 0, \frac{\partial V^\lambda}{\partial x}(x)(f(x) + g(x)u) \right\},$$

where we used the fact that, from **Corollary 1** and the continuity of both $c(x)$ and $V^\lambda(x)$,

$$\frac{\partial V^\lambda}{\partial x}(x)(f(x) + g(x)u) \leq \frac{\partial c}{\partial x}(x)(f(x) + g(x)u) \quad \forall u \in \mathcal{U} \quad \text{if} \quad V^\lambda(x) = c(x).$$

When $V^\lambda(x) < c(x)$, the min operator activates the second expression. In this case, we have $q^\lambda(x, u) = \frac{\partial V^\lambda}{\partial x}(x)(f(x) + g(x)u) + \lambda(c(x) - V^\lambda(x))$. Therefore, by the **Corollary 1** and **Lemma 1**, we can concisely write $q^\lambda(x, u)$ as the following:

$$q^\lambda(x, u) = \min \left\{ c(x) - V^\lambda(x), \frac{\partial V^\lambda}{\partial x}(x)(f(x) + g(x)u) + \lambda(c(x) - V^\lambda(x)) \right\}.$$

Thus, $\max_{u \in \mathcal{U}} q^\lambda(x, u) = 0 \forall x \in \mathcal{X}$ by **Lemma 1**, which means $q^\lambda(x, u) \leq 0 \forall (x, u) \in \mathcal{X} \times \mathcal{U}$. ■

3.4. Model-free safety filter learning

In this subsection, we construct a model-free safety filter using the time-discounted reachability formulation developed in the previous subsection. For a sampled transition $(x, u, c, x_{\delta t}^u, c_{\delta t}^u)$ observed at a fixed interval δt , the first term in (4) can be treated as monotonic over $0 \leq t \leq \delta t$. By leveraging **Corollary 1** and **Theorem 1**, we can rewrite (4) as the following:

$$Q_{\delta t}^\lambda(x, u) \simeq V^\lambda(x) + \delta t \cdot q^\lambda(x, u) \simeq \min \left\{ c, \text{int}(c, \lambda, \delta t) + e^{-\lambda \cdot \delta t} V^\lambda(x_{\delta t}^u) \right\} \quad (6)$$

where, $\text{int}(c, \lambda, \delta t)$ denotes a numerically approximated integral term $\int_0^{\delta t} \lambda e^{-\lambda \tau} c(x_\tau^u) d\tau$. From **Theorem 1**, $\max_{u \in \mathcal{U}} q^\lambda(x, u) = 0$ for all $x \in \mathcal{X}$. Using this fact and $\frac{\partial V^\lambda}{\partial x}(x)(f(x) + g(x)u) \simeq (V^\lambda(x_{\delta t}^u) - V^\lambda(x))/\delta t$, we can write

$$\begin{aligned} \frac{\partial V^\lambda}{\partial x}(x)(f(x) + g(x)u) \simeq \frac{1}{\delta t} \cdot \left(\min \left\{ c_{\delta t}^u - V^\lambda(x), e^{-\lambda \cdot \delta t} V^\lambda(x_{\delta t}^u) - V^\lambda(x) + \right. \right. \\ \left. \left. \text{int}(c_{\delta t}^u, \lambda, \delta t) + \delta t \cdot e^{-\lambda \cdot \delta t} \max_{u' \in \mathcal{U}} \frac{\partial V^\lambda}{\partial x}(x_{\delta t}^u)(f(x_{\delta t}^u) + g(x_{\delta t}^u)u') \right\} \right) \end{aligned} \quad (7)$$

We denote $\partial V^\lambda(x, u) := \frac{\partial V^\lambda}{\partial x}(x)(f(x) + g(x)u)$ and define two Bellman operators:

$$\mathcal{T}_v[V^\lambda](x, u) := \min \left\{ c, \text{int}(c, \lambda, \delta t) + e^{-\lambda \cdot \delta t} V^\lambda(x_{\delta t}^u) \right\} - \delta t \cdot q^\lambda(x, u) \quad (8)$$

$$\begin{aligned} \mathcal{T}_{dv}[\partial V^\lambda](x, u) := \left\{ \min \left\{ c_{\delta t}^u, \text{int}(c_{\delta t}^u, \lambda, \delta t) + e^{-\lambda \cdot \delta t} V^\lambda(x_{\delta t}^u) + \right. \right. \\ \left. \left. \delta t \cdot e^{-\lambda \cdot \delta t} \cdot \max_{u' \in \mathcal{U}} \partial V^\lambda(x_{\delta t}^u, u') \right\} - V^\lambda(x) \right\} / \delta t \end{aligned} \quad (9)$$

When q^λ in (8) or V^λ in (9) is treated as a fixed term independent of the function to which the Bellman operator is applied, each operator becomes an $e^{-\lambda \cdot \delta t}$ -contraction in the supremum norm. This leads to the following results.

Theorem 2 ($e^{-\lambda \cdot \delta t}$ -contraction on the discounted safety value) *Let two different value functions be V_1^λ, V_2^λ . For a fixed q^λ , $\|\mathcal{T}_v[V_1^\lambda] - \mathcal{T}_v[V_2^\lambda]\|_\infty \leq e^{-\lambda \cdot \delta t} \|V_1^\lambda - V_2^\lambda\|_\infty$.*

Theorem 3 ($e^{-\lambda \cdot \delta t}$ -contraction on the derivatives) *Let two different pairs of the derivatives be $\partial V_1^\lambda, \partial V_2^\lambda$. For a fixed safety value V^λ , $\|\mathcal{T}_{dv}[\partial V_1^\lambda] - \mathcal{T}_{dv}[\partial V_2^\lambda]\|_\infty \leq e^{-\lambda \cdot \delta t} \|\partial V_1^\lambda - \partial V_2^\lambda\|_\infty$.*

Proof Since $q^\lambda(x, u)$ in (8) for both V_1^λ and V_2^λ are fixed and identical, the effect of this term vanishes when $\mathcal{T}_v[V_1^\lambda](x, u) - \mathcal{T}_v[V_2^\lambda](x, u)$ for all pair of (x, u) . Using $|\min\{a, b\} - \min\{a, c\}| \leq$

$|b - c|$, we have the following inequality:

$$\left| \mathcal{T}_v[V_1^\lambda(x, u)] - \mathcal{T}_v[V_2^\lambda(x, u)] \right| \leq e^{-\lambda \cdot \delta t} |V_1^\lambda(x_{\delta t}^u) - V_2^\lambda(x_{\delta t}^u)|.$$

Taking the supremum norm on x and u for both sides concludes the proof of **Theorem 2**.

Similarly, terms related to V^λ are canceled out in $\mathcal{T}_{dv}[\partial V_1^\lambda](x, u) - \mathcal{T}_{dv}[\partial V_2^\lambda](x, u)$. Thus,

$$\begin{aligned} \left| \mathcal{T}_{dv}[\partial V_1^\lambda](x, u) - \mathcal{T}_{dv}[\partial V_2^\lambda](x, u) \right| &\leq e^{-\lambda \cdot \delta t} \left| \max_{u_1 \in \mathcal{U}} \partial V_1^\lambda(x_{\delta t}^u, u_1) - \max_{u_2 \in \mathcal{U}} \partial V_2^\lambda(x_{\delta t}^u, u_2) \right| \\ &\leq e^{-\lambda \cdot \delta t} \max_{u' \in \mathcal{U}} \left| \partial V_1^\lambda(x_{\delta t}^u, u') - \partial V_2^\lambda(x_{\delta t}^u, u') \right| \end{aligned}$$

Taking the supremum norm on x and u for both sides concludes the proof of **Theorem 3**. \blacksquare

These theorems imply that, for each fixed q^λ or V^λ , repeated application of the operators converges to the corresponding unique fixed points. In practice, following the standard approach in off-policy RL (Haarnoja et al., 2018; Lillicrap et al., 2015), we adopt slowly updated target networks so that the q^λ and V^λ terms can evolve gradually during training, and the learned value and derivative functions progressively converge toward better fixed points. We approximate $V^\lambda \approx v_\theta^\lambda$ and $\partial V^\lambda \approx \partial v_\phi^\lambda$, with parameters (θ, ϕ) . We denote parameters of target networks as θ' and ϕ' , which are used to compute $q^\lambda \approx q_{\theta', \phi'}^\lambda$ in (8) and $V^\lambda \approx v_{\theta'}^\lambda$ in (9). Given a mini-batch of N transitions $(x_i, u_i, c_i, x_{\delta t, i}^u, c_{\delta t, i}^u)_{i=1}^N$, we minimize the following losses:

$$\mathcal{L}_v = \frac{1}{N} \sum_{i=1}^N \left\{ v_{\theta'}^\lambda(x_i) - \min \left\{ c_i, \text{int}(c_i, \lambda, \delta t) + e^{-\lambda \cdot \delta t} v_{\theta'}^\lambda(x_{\delta t, i}^u) \right\} + \delta t \cdot q_{\theta', \phi'}^\lambda(x_i, u_i) \right\}^2 \quad (10)$$

$$\begin{aligned} \mathcal{L}_{\partial v} = \frac{1}{N} \sum_{i=1}^N \left\{ \partial v_\phi^\lambda(x_i, u_i) - \left\{ \min \left\{ c_{\delta t, i}^u, \text{int}(c_{\delta t, i}^u, \lambda, \delta t) + e^{-\lambda \cdot \delta t} v_{\theta'}^\lambda(x_{\delta t, i}^u) + \right. \right. \right. & (11) \\ \left. \left. \left. \delta t \cdot e^{-\lambda \cdot \delta t} \cdot \max_{u' \in \mathcal{U}} \partial v_{\phi'}^\lambda(x_{\delta t, i}^u, u') \right\} - v_{\theta'}^\lambda(x_i) \right\} / \delta t \right\}^2 \end{aligned}$$

The terms related to $x_{\delta t, i}^u$ are computed using the target networks for numerical stability. We further reparameterize $\partial v_\phi^\lambda(x, u) = a_\phi^\lambda(x)u - \max_{u \in \mathcal{U}} a_\phi^\lambda(x)u + b_\phi^\lambda(x)$, so that $\max_{u \in \mathcal{U}} \partial v_\phi^\lambda(x, u) = b_\phi^\lambda(x)$. This representation allows the condition from **Lemma 1** to be imposed directly through the term $b_\phi^\lambda(\cdot)$. In the sampled mini-batch, we enforce this condition by setting $b_{\phi'}^\lambda(x_{\delta t, i}^u) = -\lambda(c(x_{\delta t, i}^u) - v_{\theta'}^\lambda(x_{\delta t, i}^u))$ whenever $v_{\theta'}^\lambda(x_{\delta t, i}^u) < c(x_{\delta t, i}^u)$.

For data collection, we first generate a potentially unsafe raw command $u_{\text{raw}} \in \mathcal{U}$ and apply the filtered control u_{filtered} by solving the following QP:

$$\begin{aligned} u_{\text{filtered}} &= \arg \min_{u \in \mathcal{U}} \|u - u_{\text{raw}}\|_2^2 & (12) \\ \text{s.t. } & \underbrace{a_\phi^\lambda(x)u - \max_{u' \in \mathcal{U}} a_\phi^\lambda(x)u' + b_\phi^\lambda(x)}_{\partial v_\phi^\lambda(x, u)} + \alpha(v_\theta^\lambda(x)) \geq 0 \end{aligned}$$

Here, $\alpha(\cdot)$ is a user-defined extended class- \mathcal{K} function that regulates the aggressiveness of the filter. For simplicity, it is designed as $\alpha(z) = \alpha z$ with positive gain $\alpha > 0$. A larger α permits inputs that reduce the estimated safety value more rapidly. Before solving the QP, we solve a Linear Program

Algorithm 1 Online Model-free Safety Filter Learning

Input: System with constraint signal, observation interval δt , raw input generator $\pi_{\text{raw}}(x, t)$, $\lambda_{\text{init}} > 0$, λ -scheduler, mini-batch size N , learning rates $\gamma_\theta, \gamma_\phi$, target update ratio τ

Initialize Replay Buffer $\mathcal{B} = \{\}$, Networks $\theta, \phi, \theta' \leftarrow \theta, \phi' \leftarrow \phi$, Discount factor $\lambda \leftarrow \lambda_{\text{init}}$

for episode $n = 1, 2, \dots$ **do**

Observe the state x and the constraint c from the system

for $t = \delta t, 2\delta t, \dots$ **do**

Generate a raw input $u_{\text{raw}} \leftarrow \pi_{\text{raw}}(x, t)$

Infer the current safety value and derivatives $v_\theta^\lambda(x), a_\phi^\lambda(x), b_\phi^\lambda(x)$

$u \leftarrow$ Solve QP (12) **if** QP is feasible **else** $\arg \max_{u \in \mathcal{U}} a_\phi^\lambda(x)u$

Apply u to the system and observe transition $x_{\delta t}^u, c_{\delta t}^u$

Store transition $\mathcal{B} \leftarrow \mathcal{B} \cup (x, u, c, x_{\delta t}^u, c_{\delta t}^u)$

Sample a mini-batch of N transitions from \mathcal{B}

Compute loss $\mathcal{L}_v, \mathcal{L}_{\partial v}$ using (10), (11)

Update network parameters $\theta \leftarrow \theta - \gamma_\theta \cdot \nabla_\theta \mathcal{L}_v, \phi \leftarrow \phi - \gamma_\phi \cdot \nabla_\phi \mathcal{L}_{\partial v}$

Update target networks $\theta' \leftarrow (1 - \tau) \cdot \theta' + \tau \cdot \theta, \phi' \leftarrow (1 - \tau) \cdot \phi' + \tau \cdot \phi$

Decrease λ by the λ -scheduler

$x \leftarrow x_{\delta t}^u; c \leftarrow c_{\delta t}^u$ **if** $c_{\delta t}^u \geq 0$ **else** break

end

end

(LP) to compute $\max_{u \in \mathcal{U}} a_\phi^\lambda(x)u$. When the networks are not yet sufficiently converged during training, the QP (12) can occasionally become infeasible. In such cases, we apply the control input that achieves the LP optimum instead. Since $a_\phi^\lambda(x) \approx \frac{\partial V^\lambda}{\partial x}(x)g(x)$, the LP fallback takes a greedy ascent step on the critic’s estimated safety value. Finally, following **Corollary 2**, the discount factor λ is gradually decreased during training, so that the learned discounted value v_θ^λ converges to the undiscounted value function V . The proposed online model-free safety filter learning algorithm is summarized in **Algorithm 1**. Although designed for online learning, it can also leverage offline data since the learning is off-policy.

4. Experiments

Our method is validated through various dynamical systems, including Gymnasium environments (Towers et al., 2024) popular in model-free RL¹. We highlight two key results: (1) consistent convergence to analytic solutions regardless of δt , and (2) the least conservative behavior among baselines due to the reachability-based formulation. We also validated the method on Hopper, a contact-driven hybrid system, confirming the practical relevance of the underlying assumptions. As a diagnostic, after training, we measured QP infeasibility over 10^6 filtering calls: 0 occurrences on Double Integrator, Inverted Pendulum, and Inverted Double Pendulum, and 0.2343% on Hopper.

All QPs in (12) were solved using the `proxqp` solver (Bambade et al., 2022) with the `qp solvers` package (Caron et al., 2025). The discount factor λ was initialized at $0.1/\delta t$ and gradually decayed to $0.0001/\delta t$, while the learning rate decreased from 3×10^{-4} to 10^{-6} . Both schedules followed a polynomial decay with a power of 5 and decay intervals of 10^6 . We constructed neural networks using ELU activations (Clevert et al., 2015) and layer normalization (Ba et al., 2016). It

1. Source code: <https://github.com/snu-larr/Deep-QP-Safety-Filter>.

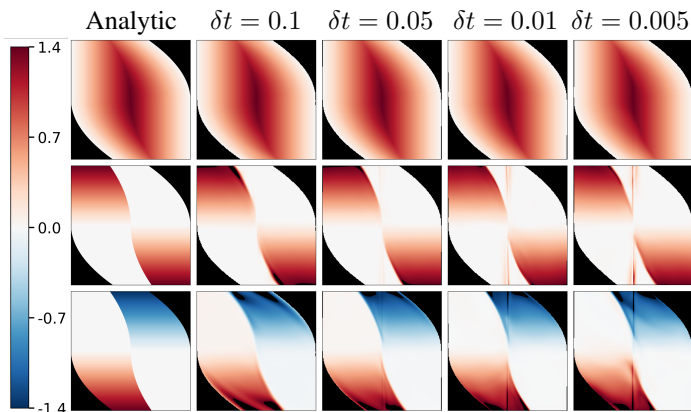


Figure 2: Comparison with analytic solutions. For each plot, the horizontal and vertical axes represent the position and velocity of the Double Integrator. From top to bottom, the first column shows the analytic solutions $V(x)$, $\max_{u \in \mathcal{U}} \partial V(x, u)$, and $\frac{\partial V}{\partial x}(x)g(x)$, while the remaining columns show the learned counterparts $v_\theta^\lambda(x)$, $b_\phi^\lambda(x)$, and $a_\phi^\lambda(x)$ under different δt .

consisted of two hidden layers with a width of 256, except for the Hopper environment, which used three layers. We trained the models using the Adam optimizer (Kingma and Ba, 2017) with $N = 256$ and $\tau = 0.005$. The raw input was generated by a random process $du_t = \kappa(\mu - u_t)dt + \sigma dW_t$, with κ , μ , and σ randomly sampled from uniform distributions for every episode. Since the safe set rendered by the safety value is scale-invariant², we normalized each newly observed constraint value by its running maximum c_{\max} to ensure that the constraint signal remained bounded by 1. We also employed two value functions, $v_{\theta,1}^\lambda$ and $v_{\theta,2}^\lambda$ to balance over- and underestimation of the safety value. When computing the next target value $v_{\theta'}^\lambda(x_{\delta t}^u)$, we used the minimum of the two estimates, whereas the current target value $v_{\theta'}^\lambda(x)$ was computed using only one of them.

4.1. Convergence to the analytic solution

We conducted experiments on the Double Integrator system. The constraint was set such that the position remains within the interval of $[-1.4, 1.4]$. We compare the learned critic against the analytic solution. The results are presented in Figure 2, where only the safety domain of $V(x) \geq 0$ or $v_\theta^\lambda(x) \geq 0$ is plotted. The critic consistently converges across different δt and captures the discontinuities in the analytic solution.

4.2. Adjusting the aggressiveness via the free parameter α

The parameter α in (12) is not learned but selected by the user to freely control the aggressiveness of the filter. Figure 3 shows the effect of different α values for the Inverted Pendulum environment with the constraint $c(x) = \min\{1 - |x_{\text{base}}|, 0.2 - |\theta_{\text{tip}}|\}$. As α increases, the controller behaves more aggressively while preserving safety. In practice, α can be set considering the sustainable limits of the hardware. Furthermore, although the constraint is defined as the minimum of two functions with different scales, the learned filter maintains safe operation.

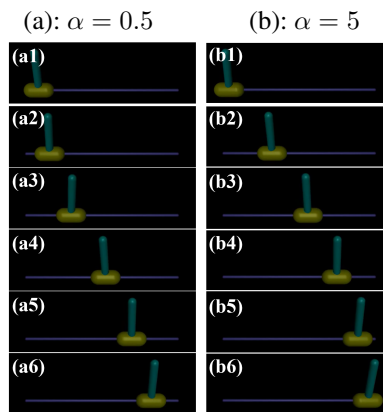


Figure 3: Effects of α on the aggressiveness. In both cases, u_{raw} is switched from -1 to 1, and the two images in each row are captured at identical time steps. Larger α produces more aggressive behavior with faster acceleration and deceleration.

2. Multiplying the safety value by any positive constant does not change the safe set, i.e., $\mathcal{S} = \{x \mid V(x) \geq 0\} = \{x \mid kV(x) \geq 0\} \forall k > 0$.

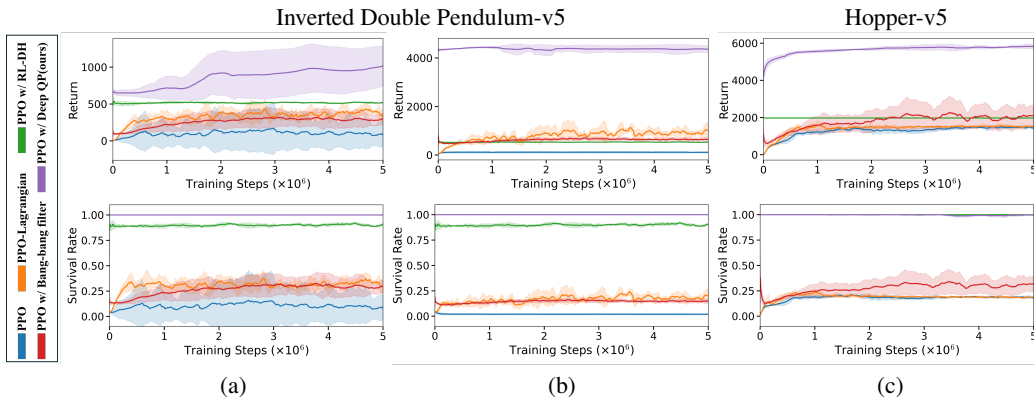


Figure 4: Comparison Results of RL tasks with the learned safety filter against other baselines. Inverted Double Pendulum environment with modified rewards (a): $|x|_{\text{base}}$ and (b): $|v|_{\text{base}}$. Hopper with default reward (c). All parameters of PPO used across baselines are identical.

4.3. Safe reinforcement learning

Our safety filter is learned task-agnostically: its training objective depends only on the safety signal $c(x)$, not on any task reward or performance objective. Thus, the same filter can be paired with any RL controller. Therefore, in this last subsection, we demonstrate the synergy between our learned safety filter and RL by showing that it can drastically reduce failures before convergence. The systems used are the Inverted Double Pendulum with $c(x) = \min\{0.95 - |x_{\text{base}}|, y_{\text{tip}} - 1\}$ and Hopper with $c(x) = \min\{z_{\text{torso}} - 0.7, 0.2 - |\theta_{\text{torso}}|\}$. Both systems are highly unstable and require sophisticated control strategies to satisfy the safety constraints. For the Inverted Double Pendulum, we considered two tasks with conflicting rewards: $|x_{\text{base}}|$ and $|v_{\text{base}}|$. The Hopper task inherently conflicts with safety because moving forward requires riskier motions. Baselines include PPO (Schulman et al., 2017), PPO-Lagrangian (Ray et al., 2019), PPO with RL-DH (Lavanakul et al., 2024), and PPO with our own variant implemented as a bang-bang safety filter ($\alpha \rightarrow \infty$) with a switching criterion of $v_{\theta}^{\lambda}(x) < 0.2$. Figure 4 shows that, even though the system is highly unstable and rewards are conflicting with the safety, the proposed method achieves faster and higher rewards with significantly fewer failures. The bang-bang variant further demonstrates that switching-based filters are unsuitable for such systems.

5. Conclusion

We presented Deep QP Safety Filter, a fully data-driven, model-free safety layer for a black-box system that provides no prior information about its dynamics. By casting safety as a time-discounted reachability problem and proving contraction of the associated Bellman operators, our method stably learns both the safety value and its derivatives. In the exact setting, the critic converges to the viscosity solution, and empirically it remains robust even when the true value is non-smooth. Experiments show that our filter preserves safety while being markedly less conservative than prior methods, achieving a practical balance between safety and task performance. We further demonstrate task-agnostic deployment: a single trained filter adapts across tasks via a scalar gain that controls aggressiveness. When combined with RL, the filter substantially reduces failure rates during training without degrading returns, and continues to perform beyond idealized assumptions when realized with neural networks. Overall, Deep QP Safety Filter delivers a principled, scalable safety mechanism for model-free control that is both theoretically grounded and practically effective.

Acknowledgments

This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT)(RS-2024-00436984).

References

- Albert Altarovici, Olivier Bokanowski, and Hasnaa Zidani. A general hamilton-jacobi framework for non-linear state-constrained control problems. *ESAIM: Control, Optimisation and Calculus of Variations*, 19(2):337–357, 2013.
- Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.
- Antoine Bambade, Sarah El-Kazdadi, Adrien Taylor, and Justin Carpentier. Prox-qp: Yet another quadratic programming solver for robotics and beyond. In *RSS 2022-Robotics: Science and Systems*, 2022.
- Somil Bansal and Claire J Tomlin. Deepreach: A deep learning approach to high-dimensional reachability. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1817–1824. IEEE, 2021.
- Stéphane Caron, Daniel Arnström, Suraj Bonagiri, Antoine Dechaume, Nikolai Flowers, Adam Heins, Takuma Ishikawa, Dustin Kenefake, Giacomo Mazzamuto, Donato Meoli, Brendan O’Donoghue, Adam A. Oppenheimer, Abhishek Pandala, Juan José Quiroz Omaña, Nikitas Rontsis, Paarth Shah, Samuel St-Jean, Nicola Vitucci, Soeren Wolfers, Fengyu Yang, @bdelhaisse, @MeindertHH, @rimaddo, @urob, @shaoanlu, Ahmed Khalil, Lev Kozlov, Antoine Groudiev, João Sousa Pinto, Roland Schwan, and Rohan Budhiraja. qpsolvers: Quadratic Programming Solvers in Python, 2025. URL <https://github.com/qpsolvers/qpsolvers>.
- Jason Choi, Fernando Castañeda, Claire J Tomlin, and Koushil Sreenath. Reinforcement learning for safety-critical control under model uncertainty, using control lyapunov functions and control barrier functions. In *Robotics: Science and Systems (RSS)*, 2020.
- Jason J Choi, Christopher A Strong, Koushil Sreenath, Namhoon Cho, and Claire J Tomlin. Data-driven hamiltonian for direct construction of safe set from trajectory data. In *2025 IEEE 64th Conference on Decision and Control (CDC)*, pages 2951–2958. IEEE, 2025.
- Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. Fast and accurate deep network learning by exponential linear units (elus). *arXiv preprint arXiv:1511.07289*, 4(5):11, 2015.
- Jaime F Fisac, Mo Chen, Claire J Tomlin, and S Shankar Sastry. Reach-avoid problems with time-varying dynamics, targets and constraints. In *Proceedings of the 18th international conference on hybrid systems: computation and control*, pages 11–20, 2015.
- 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)*, pages 8550–8556. IEEE, 2019.

- Briti Gangopadhyay, Pallab Dasgupta, and Soumyajit Dey. Safe and stable rl (s 2 rl) driving policies using control barrier and control lyapunov functions. *IEEE Transactions on Intelligent Vehicles*, 8(2):1889–1899, 2022.
- Shangding Gu, Long Yang, Yali Du, Guang Chen, Florian Walter, Jun Wang, and Alois Knoll. A review of safe reinforcement learning: Methods, theories and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.
- Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pages 1861–1870. Pmlr, 2018.
- Kai-Chieh Hsu, Haimin Hu, and Jaime F Fisac. The safety filter: A unified view of safety-critical control in autonomous systems. *Annual Review of Control, Robotics, and Autonomous Systems*, 7, 2023a.
- Kai-Chieh Hsu, Duy Phuong Nguyen, and Jaime Fernandez Fisac. Isaacs: Iterative soft adversarial actor-critic for safety. In *Learning for Dynamics and Control Conference*, pages 90–103. PMLR, 2023b.
- Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017. URL <https://arxiv.org/abs/1412.6980>.
- Will Lavanakul, Jason Choi, Koushil Sreenath, and Claire Tomlin. Safety filters for black-box dynamical systems by learning discriminating hyperplanes. In *6th Annual Learning for Dynamics & Control Conference*, pages 1278–1291. PMLR, 2024.
- Jingqi Li, Donggun Lee, Jaewon Lee, Kris Shengjun Dong, Somayeh Sojoudi, and Claire Tomlin. Certifiable reachability learning using a new lipschitz continuous value function. *IEEE Robotics and Automation Letters*, 10(4):3582–3589, 2025. doi: 10.1109/LRA.2025.3535183.
- Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.
- Jinze Liu, Minzhe Li, Jessy W Grizzle, and Jiunn-Kai Huang. Clf-cbf constraints for real-time avoidance of multiple obstacles in bipedal locomotion and navigation. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 10497–10504. IEEE, 2023.
- Alex Ray, Joshua Achiam, and Dario Amodei. Benchmarking Safe Exploration in Deep Reinforcement Learning. 2019.
- 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)*, pages 3717–3724. Ieee, 2020.
- John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

- 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)*, pages 11532–11539. IEEE, 2024.
- Andrew Taylor, Andrew Singletary, Yisong Yue, and Aaron Ames. Learning for safety-critical control with control barrier functions. In *Learning for dynamics and control*, pages 708–717. PMLR, 2020.
- Mark Towers, Ariel Kwiatkowski, Jordan Terry, John U Balis, Gianluca De Cola, Tristan Deleu, Manuel Goulão, Andreas Kallinteris, Markus Krimmel, Arjun KG, et al. Gymnasium: A standard interface for reinforcement learning environments. *arXiv preprint arXiv:2407.17032*, 2024.
- Christopher JCH Watkins and Peter Dayan. Q-learning. *Machine learning*, 8(3):279–292, 1992.
- Wei Xiao, Tsun-Hsuan Wang, Ramin Hasani, Makram Chahine, Alexander Amini, Xiao Li, and Daniela Rus. Barriernet: Differentiable control barrier functions for learning of safe robot control. *IEEE Transactions on Robotics*, 39(3):2289–2307, 2023.
- Dongjie Yu, Haitong Ma, Shengbo Li, and Jianyu Chen. Reachability constrained reinforcement learning. In *International conference on machine learning*, pages 25636–25655. PMLR, 2022.
- Songyuan Zhang, Oswin So, Kunal Garg, and Chuchu Fan. Gcbf+: A neural graph control barrier function framework for distributed safe multi-agent control. *IEEE Transactions on Robotics*, 2025.