2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

Transfer Your Safety: Learning Transferable Model-Free Safety Filters from a Single Policy to Enhance Safety Across Diverse Tasks

Junjun Xie†, Siru Li†, Shuhao Zhao, Xiaochen Xie and Liang Hu*

***Abstract*— Safety is a fundamental and pervasive requirement in robotics, yet most existing approaches rely on task-specific training or predefined models, necessitating redesign or retraining from scratch when tasks or systems change. In this paper, we propose a novel approach for constructing model-free safety filters that learns perception-based control barrier functions (CBFs) from one initial policy for arbitrary tasks and then derives task-independent safety filters in terms of CBF-based constraints in a model-free manner. The safety filters can be flexibly integrated into policies for diverse tasks and remain robust to mild environmental variations. We further theoretically prove that the safety filters can improve the safety of the initial policy itself, relaxing the safety requirements on initial policies used for CBF construction. The proposed method is systematically evaluated over multiple safety-critical tasks and random environments, validating the effectiveness and generalizability of our method. Notably, starting with an initial LiDAR-only navigation policy, our approach successfully learn LiDAR-visual multimodal CBFs with LiDAR and vision inputs, and applies them to different downstream tasks.**

I. Introduction

As robotic systems undertake increasingly complex tasks in unstructured and cluttered environments, safety becomes a central challenge that must be addressed. Recently, the learning-based methods have been widely explored to increase the capability of robots in such settings [1], [2], [3]. Among them, control barrier functions (CBFs) have gained attention as an effective tool for encoding safety constraints combined with learning-based methods [4], [5], [6], [7]. Existing methods can be roughly categorized into two classes: one employs neural networks to *learn CBF formulation* in complex environments, and the other integrates CBFs into reinforcement or imitation learning pipelines for *safe policy design*.

However, several limitations remain in existing learning-based approaches. Firstly, CBF formulation requires explicit safety labels during data collection, which is cumbersome and costly. Secondly, most CBF-based methods rely on predicting safety in a future horizon to select safe actions, which in turn requires prior models of system dynamics and perception—an unrealistic assumption for robots equipped with high-dimensional sensors such as cameras and LiDARs.

This work was supported in part by the National Natural Science Foundation of China under Grant 62573157, the Science Center Program of National Natural Science Foundation of China under Grant 62188101, Shenzhen Science and Technology Program under Grant SYSPG20241211173609005, under Grant JCYJ20241202123714019, and under Grant JCYJ20250604145333044.

† Equal Contribution, * Corresponding Author.
The authors are with the Department of Automation, School of Intelligence Science and Engineering, Harbin Institute of Technology, Shenzhen, China. For correspondence: l.hu@hit.edu.cn.

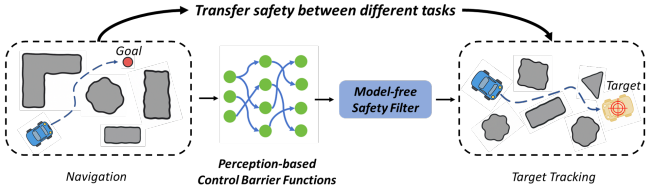


Fig. 1. Illustration of our approach. Perception-based CBFs are learned from an existing policy (such as a navigation policy) to encode underlying general safety properties and embed them into the model-free safety filter, thereby enhancing the safety of policies across different tasks.

Finally, current methods typically learn a safe policy from scratch for each task and environment, without reusing safety knowledge across tasks, resulting in poor data efficiency.

In principle, a robot's safety should depend on environmental configurations (e.g., obstacle locations and shapes) and intrinsic robot characteristics (e.g., dynamics and sensor capabilities), rather than specific tasks. For instance, in an obstacle-dense environment, whether performing navigation, exploration, or target tracking, the fundamental safety requirement remains avoiding collisions with obstacles. Such a requirement is determined by the environment's structure and the robot's properties, rather than the nature of the task itself. The observation suggests that safety properties can be generalized across different tasks within similar scenarios. Motivated by it, we are going to tackle the following problem in this paper: *Given that a robot has a task-specific policy (called the initial policy), not necessarily with guaranteed safety. Can we learn the knowledge of safety from the initial policy that can be transferred to other tasks in similar environments and enhance their safety accordingly?*

To this end, we propose a *model-free* safety filter represented by neural CBFs, which is learned from one initial task and subsequently applied to policies for diverse tasks, as illustrated in Fig. 1. The learned CBFs capture the generalizable safety requirement from robot perception, enabling the safety filter to transfer the safety knowledge flexibly across tasks. The key contributions of this work are summarized as follows:

1) We develop a robust and flexible model-free perception-based safety filter that is transferable to policies for other tasks without requiring models of the system dynamics or perception.
2) We provide a theoretical guarantee that the learned safety filters improve the safety of the initial policy, which relaxes the safety demands on initial policies

used for CBF formulation and thus broadens its practical applicability.

3) Given a LiDAR-only navigation policy, we learn multimodal CBFs (LiDAR and RGB-camera), and transfer them to policies for three distinct safety-critical tasks in random environments: cruising, target tracking, and navigation, validating both flexibility and effectiveness of our approach.

II. Related Work

A. Learning-based Safe Policies

Safety is indispensable in robotics and has been increasingly embedded into learning-based methods to handle complex tasks and environments, with reinforcement learning (RL) and imitation learning (IL) being the most representative approaches. In RL, safety is typically promoted through reward shaping with safety-oriented terms [8], [9], while these designs are largely heuristic and provide limited theoretical assurance. A more principled framework is safe reinforcement learning (safe RL), which incorporates constraints during training to obtain policies that satisfy safety requirements. These methods have also been broadly demonstrated across various robots [10], [11], [12]. However, precise task objectives and constraints are demanded before training, leading to policies that are confined to predefined tasks and often accompanied by an unstable training process. Imitation learning mainly uses expert demonstrations for training, but the performance depends on the quality of the data [13], [14], [15], and is susceptible to fragile corner cases, leading to task-specific solutions with poor generalization.

Owing to the strong theoretical foundation, CBFs have been widely incorporated into learning-based methods to enhance interpretability and enable theoretical analysis on safety. There are two main approaches: the first implicitly incorporates CBFs into the training process to enhance policy safety [16], [17], [18], but typically lacks formal guarantees similar to adding safety rewards in RL; the second applies CBFs as a safety filter that explicitly modifies the original policy to meet safety requirements which is highly flexible and offers strong theoretical guarantees [19], [20], [21], but it relies on system models to predict future safety, making it incompatible with the model-free learning-based methods.

B. Neural Control Barrier Functions

To represent safety requirements in complex tasks and scenarios, neural control barrier functions (NCBFs) are typically learned by training neural networks on collected data. A prevailing approach involves designing loss functions that enforce fundamental CBF properties such as separating safe from unsafe sets [22], [23], [24], as well as additional criteria like feasibility and optimality [25].

While NCBFs provide mappings from states or observations [26] to safety-related function values, they do not account for the effect of actions on safety directly, but rather require a prior model to predict subsequent states or observations, evaluating future CBF values to assess action feasibility. This dependency complicates implementation and restricts the approach to model-based scenarios.

A further limitation is that data labeling relies on manual quantification, which is challenging to assess since it depends on the robot's control performance and must account for both the present state and subsequent long-term behavior. For example, in obstacle avoidance scenarios, states or observations are typically labeled as safe if no collision occurs or at a safe distance, based solely on the relative position between the robot and obstacles. However, for a robot with limited braking capability, even if it is currently in a collision-free state at a certain velocity, it may be unable to avoid a collision in the future. Therefore, labeling such a state as safe is unreasonable for this robot.

Recent research has explored deriving CBFs from policies to address these problems. Specific policies can be used to obtain data for training the CBF, implicitly accounting for input constraints [27]. The relationship between value functions and CBFs has been established [28], and a specially designed reward structure in RL has been leveraged to derive CBFs [29]. But these CBFs are built upon specific or optimal policies and often suffer from issues such as reward hacking during training, making it difficult to achieve the desired results.

III. Preliminaries and Problem Formulation

Consider a robot system modeled as a Markov decision process (MDP) [30]

$$s_{t+1} = \mathcal{F}(s_t, a_t) \tag{1}$$

which can be defined concisely as a tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{F}, R, \gamma \rangle$, where $\mathcal{S}$ is a set of states satisfied $s_t \in \mathcal{S} \subseteq \mathbb{R}^n$, $\mathcal{A}$ is a set of actions satisfied $a_t \in \mathcal{A} \subseteq \mathbb{R}^m$, $\mathcal{F} : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$ is a possibly dynamics, $R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is a reward function, and $\gamma \in [0, 1]$ is a discount factor.

A. Reinforcement Learning

Reinforcement learning aims to learn a policy via interaction between the environment and the agent with the dynamic model (1). The cumulative reward is defined as $G_t = \sum_{k=0}^{\infty} \gamma^k R_{t+k}$. With a policy π, state value function $V^\pi(s)$ and action value function $Q^\pi(s, a)$ are defined as

$$\begin{aligned} V^\pi(s) &= \mathbb{E}^\pi[G_t | S_t = s], \\ Q^\pi(s, a) &= \mathbb{E}^\pi[G_t | S_t = s, A_t = a]. \end{aligned} \tag{2}$$

The objective is to find an optimal policy π^* to make $V^{\pi^*}(s) \geq V^\pi(s), \forall \pi, s \in \mathcal{S}$.

B. Discrete-time Control Barrier Functions

Consider the discrete-time system (1), we have the following definitions and a lemma:

Definition 1. *(Discrete-time Control Barrier Function [31]) The function $h : \mathbb{R}^n \to \mathbb{R}$ is called a* ***Discrete-time Control Barrier Function*** *for the system* (1) *if there exists λ such that*

$$\sup_{a_t \in \mathcal{A}} [h(\mathcal{F}(s_t, a_t)) + (\lambda - 1)h(s_t)] \geq 0, 0 \leq \lambda \leq 1. \tag{3}$$

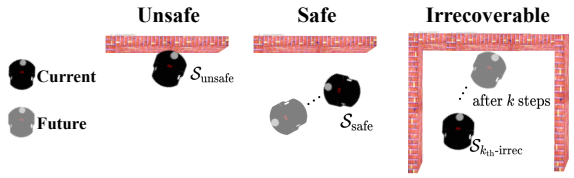


Fig. 2. An example for unsafe/safe/irrecoverable states. The unsafe set includes collided states. The safe set contains states from which it is possible to proceed indefinitely without collision. The k_{th} irrecoverable set encompasses states from which there is a collision within k steps.

Lemma 1. (Forward Invariant [32], [33], [34]) The set $\mathcal{C}$ is defined as the zero-superlevel set of a smooth function $h : \mathbb{R}^n \to \mathbb{R}$, i.e.,

$$\begin{aligned} \mathcal{C} &= \{s \in \mathbb{R}^n : h(s) \geq 0\}, \\ \partial\mathcal{C} &= \{s \in \mathbb{R}^n : h(s) = 0\}, \\ \text{Int}(\mathcal{C}) &= \{s \in \mathbb{R}^n : h(s) > 0\}, \end{aligned} \tag{4}$$

then the system (1) is **forward invariant** with respect to the set $\mathcal{C}$, i.e.,

$$\forall s_t \in \mathcal{C} \Rightarrow s_{t+1} \in \mathcal{C} \tag{5}$$

if

$$h(s_{t+1}) + (\lambda - 1)h(s_t) \geq 0, 0 \leq \lambda \leq 1. \tag{6}$$

Proof. Since $s_t \in \mathcal{C}$, from (4) we have $h(s_t) \geq 0$. If (6) holds on, then $h(s_{t+1}) \geq (1-\lambda)h(s_t) \geq 0$, which means $s_{t+1} \in \mathcal{C}$. ■

C. State Space Partitioning

Following the approach in [29], we categorize the entire state space $\mathcal{S}$ into three classes: unsafe set $\mathcal{S}_{\text{unsafe}}$, safe set $\mathcal{S}_{\text{safe}}$ and k-steps irrecoverable set $\mathcal{S}_{k_{\text{th}}\text{-irrec}}$, respectively. Once the unsafe set $\mathcal{S}_{\text{unsafe}}$ is defined, the relationships are described as follows:

$$\begin{aligned} s_0 \in \mathcal{S}_{\text{safe}} &\Leftrightarrow \exists\pi, \forall t > 0, s_t \notin \mathcal{S}_{\text{unsafe}} \\ s_0 \in \mathcal{S}_{k_{\text{th}}\text{-irrec}} &\Leftrightarrow \forall\pi, \max_t \{s_t \notin \mathcal{S}_{\text{unsafe}}\} = k \end{aligned} \tag{7}$$

As illustrated in Fig. 2, taking obstacle avoidance as an example, $\mathcal{S}_{\text{unsafe}}$ is typically defined as the region occupied by obstacles, then $\mathcal{S}_{\text{safe}}$ represents states where there exist actions from the action space such that collisions can always be avoided. $\mathcal{S}_{k_{\text{th}}\text{-irrec}}$ includes states that are definitely collisions within k steps (for example, due to high-speed motion). To ensure safety, our goal is to guarantee that the robot's state never enters the unsafe set, i.e., $s_t \notin \mathcal{S}_{\text{unsafe}}, \forall t > 0$.

IV. Methodology

In this section, we first establish a theoretical connection between the value function and CBF, and then develop a corresponding safety filter with a formal analysis. Based on these results, we further propose a method to learn perception-based CBFs from existing policies. Finally, we highlight the flexibility and self-improvement ability of our approach. The algorithm framework is depicted in Fig. 3.

A. Value Function as Control Barrier Function

There is a zero-one reward shaping mechanism proposed in [29] that naturally transfers the optimal value function to a CBF, which is formalized as *Lemma* 2:

Lemma 2. **(Optimal Value Function as Control Barrier Function [29])** Assume that there is an immediate termination if $s \in \mathcal{S}_{\text{unsafe}}$, and define reward function $r(s,a) = 0$ if $s \in \mathcal{S}_{\text{unsafe}}$ and $r(s,a) = 1$ otherwise, then a control barrier function can be constructed based on the optimal value function $V^*(s)$ as $h(s) = V^*(s) - \varepsilon$, where $\varepsilon \in (0, \frac{1}{1-\gamma}]$, i.e.,

- $h^\pi(s) \geq 0$ if $s \in \mathcal{S}_{\text{safe}}$;
- $h^\pi(s) < 0$ if $s \in \mathcal{S}_{\text{unsafe}}$.

Proof. By definition (2):

- for $s \in \mathcal{S}_{\text{unsafe}}$, we have $V^*(s) = 0$ due to immediate termination;
- for $s \in \mathcal{S}_{\text{safe}}$, there exists a policy π that guarantees the safety, so that $V^*(s) = \Sigma_{i=0}^{\infty}\gamma^i = \frac{1}{1-\gamma}$;
- for $s \in \mathcal{S}_{k_{\text{th}}\text{-irrec}}$, the agent falls in $\mathcal{S}_{\text{unsafe}}$ within k steps, hence $V^*(s) = \Sigma_{i=0}^{k-1}\gamma^i = \frac{1-\gamma^k}{1-\gamma}$.

Then any bias $\varepsilon \in (0, \frac{1}{1-\gamma}]$ makes $h(s) = V^*(s) - \varepsilon$ is a control barrier function satisfying $h(s) \geq 0$ if $s \in \mathcal{S}_{\text{safe}}$ and $h(s) < 0$ if $s \in \mathcal{S}_{\text{unsafe}}$. ■

This algorithm has been verified on several benchmarks [35], but it exclusively examines the relationship between the *optimal* value function and CBF. We would like to extend it by a further step, that is the relationship between the value function of an arbitrary policy and the CBF, as stated in *Corollary* 1:

Corollary 1. **(Value Function as Control Barrier Function)** Assume there is an immediate termination if $s \in \mathcal{S}_{\text{unsafe}}$, and define reward function $r(s,a) = 0$ if $s \in \mathcal{S}_{\text{unsafe}}$ and $r(s,a) = 1$ otherwise, then the value function $V^\pi(s)$ of any policy π can be transferred to a (conservative) control barrier function $h^\pi(s) = V^\pi(s) - \varepsilon$ where $\varepsilon \in (0, \frac{1}{1-\gamma}]$, i.e.,

- $h^\pi(s) < 0$ if $s \in \mathcal{S}_{\text{unsafe}}$.

Proof. Since π is an arbitrary policy, $V^\pi(s) \leq V^*(s)$, then for $s \in \mathcal{S}_{\text{unsafe}}$, $V^\pi(s) \leq V^*(s) = 0$, and hence $h^\pi(s) = V^\pi(s) - \varepsilon < 0$. Different from *Lemma* 2, for $s \in \mathcal{S}_{\text{safe}}$ or $s \in \mathcal{S}_{k_{\text{th}}\text{-irrec}}$, $h^\pi(s) \leq 0$ might also hold, indicating that it might regard some safe states as unsafe, and thus it is a conservative CBF. ■

Corollary 1 shows that the value function of any policy can be transferred into a CBF. Therefore, there is no need to train an RL algorithm completely to obtain an optimal policy; instead, any available (non-optimal) policy can be leveraged for CBF learning via policy evaluation.

In practical applications, since the accurate value function is usually unavailable and instead approximated using a neural network, the approximation error may cause ambiguity in determining system safety. While [29] analyses the case where the optimal value function has bounded approximation errors, we further investigate the relationship between the

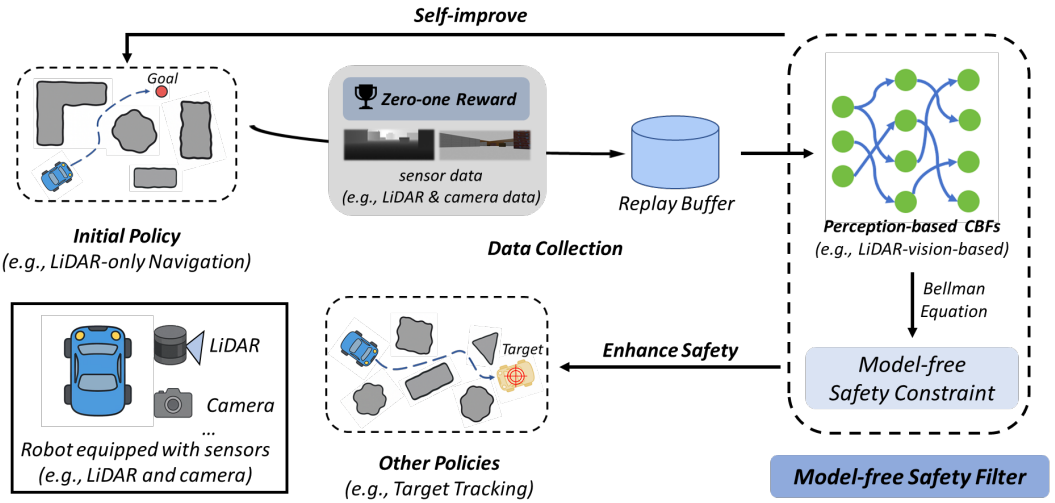


Fig. 3. The framework of our method. The perception-based control barrier function is trained by observations sampled from an initial policy based on a zero-one reward. With the model-free safety constraint, the learned CBF can be applied to enhance the safety of other policies for diverse tasks or improve the initial policy itself. The modality used for CBF (e.g., LiDAR and vision) is independent of the initial policy (e.g., LiDAR-based) and does not impose restrictions on the modality of the filtered policy.

value function of an arbitrary policy and the control barrier function under bounded errors, formalized in the following corollary:

Corollary 2. **(Estimated Value Function as Control Barrier Function)** Given the same assumption in *Corollary* 1, assume the value function $V^\pi(s)$ of policy π is estimated by $\hat{V}^\pi(s)$ with a bounded approximation error $|V^\pi(s) - \hat{V}^\pi(s)| \leq e$ for any state $s \in \mathcal{S}$, then the estimated value function $\hat{V}^\pi(s)$ can be transferred to a (conservative) control barrier function $h^\pi(s) = \hat{V}^\pi(s) - \varepsilon$ where $\varepsilon \in (e, \frac{1}{1-\gamma}]$, i.e.,

- $h^\pi(s) < 0$ if $s \in \mathcal{S}_{\text{unsafe}}$.

Proof. For $s \in \mathcal{S}_{\text{unsafe}}$,$V^\pi(s) \leq V^*(s) = 0$. Since $|V^\pi(s) - \hat{V}^\pi(s)| \leq e$, then $\hat{V}^\pi(s) \leq e$ and $h^\pi(s) = \hat{V}^\pi(s) - \varepsilon < 0$. ■

As discussed above, one of the main challenges in combining learning-based methods with CBFs is predicting future CBF values under different actions without system models. Leveraging the relationship between state and action value functions in reinforcement learning, we next propose a model-free safety filter tailored to CBF expressed by value functions.

B. Model-free Safety Filter

In model-based methods, safety requirements are often formulated as the CBF constraint (6), which involves predicting the state at the next time instant s_{t+1} under candidate action a_t to check if the safety constraint is met. For the model-free paradigm, we propose to derive the future CBF values based on Bellman equation without predicting future states, expressed in the following **model-free safety constraint**:

$$Q^\pi(s_t, a_t) \geq \gamma \min\{\varepsilon, V^\pi(s_t)\} + 1. \tag{8}$$

Here, $V^\pi(s_{t+1})$ is estimated from $Q^\pi(s_t, a_t)$ via $V^\pi(s_{t+1}) \approx \frac{Q^\pi(s_t,a_t) - r(s_t,a_t)}{\gamma}$. Futhermore, for the special case that the system (1) is deterministic, (8) strictly guarantees safety as stated in the following theorem:

Algorithm 1 Learn Perception-based CBFs from Existing Policies

```
Initial policy π, replay buffer D ← {}, critic network Q_φ, V_Φ, discount factor γ
for each step t in training do
    Sample action a_t = π(o_t)
    Get observation o_{t+1}
    if unsafe then
        r_t ← 0
    else
        r_t ← 1
    end if
    D ← D ∪ {(o_t, a_t, r_t, o_{t+1})}
    Update critic parameters φ, Φ
end for
Choose bias ε
h ← V_Φ − ε
Output: h
```

Theorem 1. **(Safety Guarantee for Deterministic Systems)** Assume system (1) is deterministic, then $h^\pi(s_{t+1}) \geq 0$ if $h^\pi(s_t) \geq 0$ and the action a_t satisfies (8)

Proof. Due to the zero-one reward function $r(s, a)$, i.e., $r \leq 1$, we have $Q^\pi(s_t, a_t) \geq \gamma \min\{\varepsilon, V^\pi(s_t)\} + r(s_t, a_t)$ that follows from (8). From the Bellman equation, we have $Q^\pi(s_t, a_t) = r(s_t, a_t) + \gamma V^\pi(s_{t+1})$ for deterministic system, which implies that $r(s_t, a_t) + \gamma V^\pi(s_{t+1}) = Q^\pi(s_t, a_t) \geq \gamma \min\{\varepsilon, V^\pi(s_t)\} + r(s_t, a_t)$, then $V^\pi(s_{t+1}) - \min\{\varepsilon, V^\pi(s_t)\} \geq 0$. Furthermore, $\min\{\varepsilon, V^\pi(s_t)\} = \lambda\varepsilon + (1-\lambda)V^\pi(s_t)$ with $\lambda = 1$ if $\varepsilon < V^\pi(s_t)$ and 0 otherwise. As such, there always exists $\lambda \in [0, 1]$ satisfying $V^\pi(s_{t+1}) - \lambda\varepsilon - (1-\lambda)V^\pi(s_t) \geq 0$. Substituting $h^\pi(s) = V^\pi(s) - \varepsilon$

into it, we have $h^\pi(s_{t+1}) + (\lambda - 1)h^\pi(s_t) \geq 0$. It follows from (6) that $h^\pi(s_{t+1}) \geq 0$ holds. ■

Based on (8), we propose a **Model-free Safety Filter** to rectify unsafe actions within arbitrary nominal policy π_{nom} as

$$\pi_{\text{filter}}(s) = \begin{cases} \arg\min_a \|a - \pi_{\text{nom}}(s)\| & \text{if there exists } a \text{ satisfies (8)} \\ \arg\max_a Q^\pi(s,a) & \text{otherwise} \end{cases} \tag{9}$$

The formulation (9) implies that the nominal policy remains unchanged if it already satisfies the safety constraint (8); otherwise, it attempts to meet the constraint through the minimal necessary adjustments. If no action satisfies the constraint, the action that maximizes $Q^\pi(s,a)$ is selected.

C. Self-improvement over Initial Policies

The derived safety filter can improve the safety of the initial policies used for CBF formulation, as stated in the following theorem:

Theorem 2. Consider a CBF h^π learned from a given policy π for system (1), then the policy π_{filter} derived from (9) is strictly better than the initial policy π in terms of safety, if $\exists s_0, V^\pi(s_0) < \max_a Q^\pi(s_0,a)$ and $h^\pi(s_0) < 0$ hold.

Proof. Firstly, when $a_t = \pi(s_t)$, it is obvious that the value of $Q(s_t,a_t)$ will not decrease after filtering, that is $Q^\pi(s_t, \pi_{\text{filter}}(s_t)) \geq Q^\pi(s_t,\pi(s_t)) = V^\pi(s_t)$ for all s_t, which implies $V^{\pi_{\text{filter}}}(s_t) \geq V^\pi(s_t)$.

Next, we prove that $Q^\pi(s_0,\pi_{\text{filter}}(s_0)) > V^\pi(s_0)$. If $h^\pi(s_0) < 0$, then $V^\pi(s_0) < \varepsilon \leq \frac{1}{1-\gamma}$ follows from *Corollary* 1, and we have $V^\pi(s_0) < \gamma V^\pi(s_0) + 1$, while the model-free constraint (8) now becomes $V^\pi(s_0) \geq \gamma V^\pi(s_0) + 1$, which implies the policy π necessarily violates (8) when $h^\pi < 0$. Hence, if there exists a satisfies (8), then $\pi_{\text{filter}}(s_0)$ satisfies that $Q^\pi(s_0,\pi_{\text{filter}}(s_0)) \geq \gamma V^\pi(s_0) + 1 > V^\pi(s_0)$; otherwise, as $V^\pi(s_0) < \max_a Q^\pi(s_0,a)$, $Q^\pi(s_0,\pi_{\text{filter}}(s_0)) = \max_a Q^\pi(s_0,a) > Q^\pi(s_0,\pi(s_0)) = V^\pi(s_0)$.

Finally, from policy improvement theory [30], the filtered policy π_{filter} is strictly better than π. ■

To put it simply, *Theorem* 2 shows that if there exist safer action candidates than those generated by the initial policy π at unsafe states, then the safety filter will definitely improve safety in these cases. Such a self-improvement ability enables our method to derive a better policy from a suboptimal one. Besides, *Corollary* 1 implies that how safe the initial policy is influences the conservativeness of the learned CBF. Consequently, even starting with an initial policy with limited safety guarantees, our framework enables iterative self-improvement: learning an enhanced policy with higher safety, then deriving refined CBFs, and repeating the process iteratively, which significantly improves the usability of our method.

D. Practical Improvements

1) Bounded activation function: From the analysis in subsection IV-A, the derived value functions $V^\pi(s)$ are bounded within $[0, \frac{1}{1-\gamma}]$. Accordingly, we employ a Gaussian activation function in the output layer to ensure bounded outputs while preserving smooth gradients. To further stabilize training in its early stages, we initialize the bias of the final linear layer with a negative constant and keep it fixed during optimization. This design choice enforces the critic's initial outputs to remain close to zero, providing a consistent starting point and mitigating instability in value estimation.

2) Robustness improvement: Since observation-based control barrier functions (oCBFs) [26] are more robust against environmental changes than state-based ones, we directly employ observations as inputs and proceed with the learning of oCBFs using an existing policy π under zero-one reward design, outlined as **Algorithm** 1.

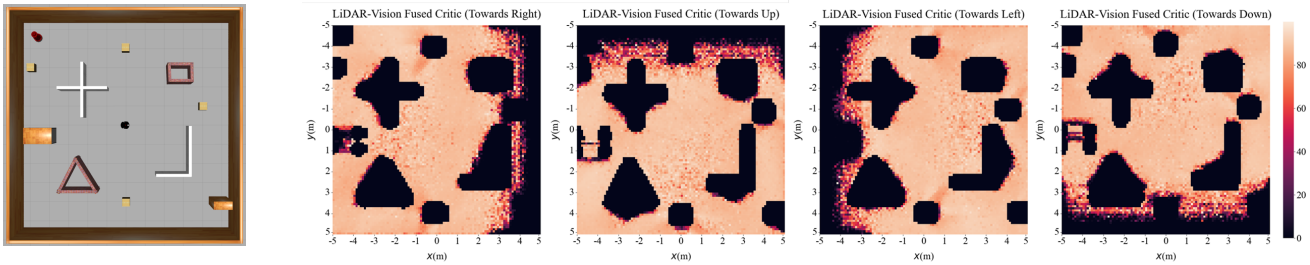
V. Experiments

In this section, we first apply our method to learn multi-modal CBFs and construct the safety filter based on LiDAR and RGB-camera data. Then we incorporate the safety filter into the nominal policies of three safety-critical tasks to increase their safety. The LiDAR-Vision data are collected during the execution of a LiDAR-only DRL-based navigation policy π_{nav} [36] in a $10\text{m} \times 10\text{m}$ Gazebo simulation environment with multiple irregular obstacles as shown in Fig. 4(a). The robot is equipped with a 180° FOV LiDAR and a 110° FOV RGB camera, and the action $a = [v, \omega]$ represents linear and angular velocity respectively where $v \in [0,1]$ m/s and $\omega \in [-1,1]$ rad/s. The discount factor $\gamma = 0.99$, implying that the value of critics $V^{\pi_{\text{nav}}} \in [0,100]$. The bias ε is chosen as 50. In each training episode with a maximum of 500 steps, the starting point of the robot and the positions of some obstacles in the environment are randomly reset to increase the data diversity.

A. Learn LiDAR-Vision Fused Control Barrier Functions

1) Single-frame Multimodal Control Barrier Functions: We first learn CBFs based on LiDAR-vision fusion with single frames. To derive the value function of π_{nav}, TD3 algorithm [37] is employed for training, but without updating the parameters of the policy network during the process. The structure of the critic network is designed as depicted in Fig. 5.

Following 600 training episodes, we systematically evaluate the trained CBF by sampling robot poses across obstacle-free regions, with the resulting heatmaps visualized in Fig. 4(b). Regions containing obstacles are assigned zero values in the heatmap since they are inaccessible to the robot. The results show that positions closer to obstacles exhibit lower values, indicating that collisions may remain unavoidable even at certain distances from obstacles, which is an inherent consequence of accounting for input constraints. Moreover, the results demonstrate the significant influence of robot orientation on regional safety assessment. For instance, in the heatmap where the robot is facing to the right, the left-hand regions of all obstacles consistently display lower values, suggesting that these areas are more prone to collisions.



(a) Simulation environment (b) LiDAR-Vision fused critics

Fig. 4. Simulation environment and single-frame multimodal critics based on LiDAR and vision.

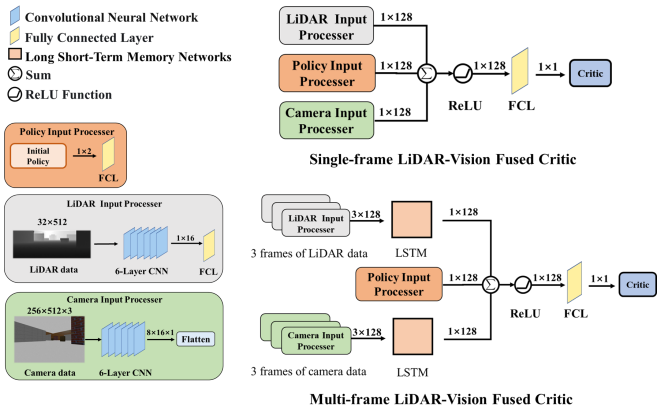
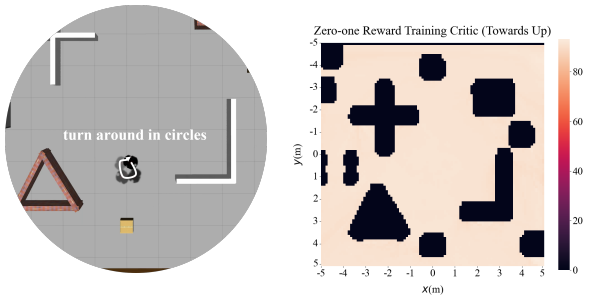


Fig. 5. Network structures of LiDAR-Vision fused critics based on single or multiple frames.



(a) (b)

Fig. 6. Reward hacking of training both policy and critic using zero-one reward directly. The policy easily converges to an unexpected and conservative policy, which always turns around in circles and results in an ineffective CBF due to inadequate sampling.

We further compare CBFs derived from existing safe policies against those obtained through reinforcement learning with direct zero-one rewards and the associated value functions [29]. After the same episodes, the latter policy degenerates into behaviors such as spinning in place as Fig. 6(a), and the resulting heatmap Fig. 6(b) fails to encode obstacle-related information, with nearly all regions remaining uniformly white. This overly conservative behavior hampers adequate sampling, thereby preventing the acquisition of meaningful obstacle awareness.

TABLE I

STEPS COMPARISON BASED ON DIFFERENT SAFETY FILTERS

Safety filters	Inputs	Min.	Max.	Average
W/o safety filters	-	35	202	87
Proposed in [29]	Single-frame	403	924	539
	Multi-frame	509	1865	987
Ours	Single-frame	462	1007	648
	Multi-frame	**524**	**2089**	**1116**

2) Multi-frame Multimodal Control Barrier Functions: Since single-frame data cannot capture the velocity information, which is critical for safety, we train a multi-frame CBF (here we use three frames: current and the previous two frames) and design the corresponding network architecture, as illustrated in Fig. 5. As the outputs of multi-frame CBF no longer have a one-to-one correspondence with robot poses, making direct visualization via heatmaps infeasible, we instead evaluate its effectiveness through downstream tasks discussed in detail in the next subsection.

B. Cross-task Safety Enhancement and Self-improvement

To further validate the correctness of our method, we deploy the proposed safety filter with multimodal CBFs across diverse tasks, including cruising, target tracking, and navigation, to enhance the safety performance of existing policies.

1) Safe Cruising: The first task is cruising, where the preset policy is a constant velocity $v_{\text{nom}} = 0.3$ m/s without considering obstacle avoidance, and the safety of the policy is evaluated by the number of steps taken before a collision. A total of 20 tests are conducted with random initial positions and obstacles for each test, and a sample trajectory is depicted in Fig. 7(a). We also test the preset policy without safety filters and another safety constraint proposed in [29] as the formulation $Q(s, a) \geq \varepsilon$ for comparison with statistical results presented in Table I. The results show that our safety filter provides stronger safety guarantees, and multi-frame CBFs that implicitly capture velocity information yield better performance.

2) Safe Target Tracking: We utilize two robots to validate performance in target tracking tasks. The leader robot follows a predefined trajectory and transmits coordinate information to the follower robot. The follower implements a prede-

(a) Cruising (b) Target tracking (c) Navigation

Fig. 7. Trajectories results of three safety-assured tasks using our multi-frame LiDAR-Vision fused safety filter.

fined target-tracking policy without considering collision avoidance and is enhanced via the proposed safety filter based on the multi-frame CBF. As shown in Fig. 7(b), the follower robot, without global map information, successfully avoids collisions by correcting unsafe actions on the fly. In particular, it also maintains collision-free with the leader robot, a case that was never encountered during training, which validates the reliability and generalizability to new and unseen scenarios.

3) Self-improved Safe Navigation: To verify the self-improvement ability of our method, we integrate the CBF-based safety filter into the LiDAR-based policy π_{nav} used to learn the CBF, obtaining the new navigation policy $\pi_{\text{nav}}^{\text{filter}}$. We compare the performance of π_{nav} and $\pi_{\text{nav}}^{\text{filter}}$ in navigation tasks, and a representative experimental result is depicted in Fig. 7(c). In 20 test runs with randomized start, goal positions, and obstacle configurations, π_{nav} succeeds 13 times (65% success rate), while $\pi_{\text{nav}}^{\text{filter}}$ achieves 17 successful runs (85% success rate), which confirms that the learned CBF filter can improve the initial policy itself in safety.

VI. Conclusion

In this paper, we propose a method to learn robust perception-based CBFs from existing policies and derive a flexible model-free safety filter, eliminating the need to predict the next observation via explicit system models. The learned CBFs and safety filter are applied to multiple tasks to enhance safety, including cruising, target tracking, and navigation, demonstrating the robustness, flexibility, and self-improvement ability of our method. The bias currently used to convert value functions into CBF thresholds is selected heuristically, and it is empirically shown that the value of the bias affects the conservatism. A proper selection strategy on the bias deserves further investigation in the future.

References

[1] M. Wei, L. Zheng, Y. Wu, H. Liu, and H. Cheng, "Safe learning-based control for multiple uavs under uncertain disturbances," *IEEE Transactions on Automation Science and Engineering*, vol. 21, no. 4, pp. 7349–7362, 2024.

[2] Z. Zhou, O. S. Oguz, M. Leibold, and M. Buss, "A general framework to increase safety of learning algorithms for dynamical systems based on region of attraction estimation," *IEEE Transactions on Robotics*, vol. 36, no. 5, pp. 1472–1490, 2020.

[3] D. Zhu, C. Zhu, Z. Zhang, S. Xin, and Y. Liu, "Learning safe locomotion for quadrupedal robots by derived-action optimization," in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2024, pp. 6870–6876.

[4] M. Tong, C. Dawson, and C. Fan, "Enforcing safety for vision-based controllers via control barrier functions and neural radiance fields," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 10 511–10 517.

[5] N. C. Janwani, E. Daş, T. Touma, S. X. Wei, T. G. Molnar, and J. W. Burdick, "A learning-based framework for safe human-robot collaboration with multiple backup control barrier functions," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 2024, pp. 11 676–11 682.

[6] J. Kim, J. Lee, and A. D. Ames, "Safety-critical coordination of legged robots via layered controllers and forward reachable set based control barrier functions," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 2024, pp. 3478–3484.

[7] M. Harms, M. Kulkarni, N. Khedekar, M. Jacquet, and K. Alexis, "Neural control barrier functions for safe navigation," in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2024, pp. 10 415–10 422.

[8] C. Vlachos, P. Rousseas, C. P. Bechlioulis, and K. J. Kyriakopoulos, "Reinforcement learning-based optimal multiple waypoint navigation," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 1537–1543.

[9] H. Lin, W. Ding, Z. Liu, Y. Niu, J. Zhu, Y. Niu, and D. Zhao, "Safety-aware causal representation for trustworthy offline reinforcement learning in autonomous driving," *IEEE Robotics and Automation Letters*, vol. 9, no. 5, pp. 4639–4646, 2024.

[10] Y. Kim, H. Oh, J. Lee, J. Choi, G. Ji, M. Jung, D. Youm, and J. Hwangbo, "Not only rewards but also constraints: Applications on legged robot locomotion," *IEEE Transactions on Robotics*, vol. 40, pp. 2984–3003, 2024.

[11] H. Cao, H. Xiong, W. Zeng, H. Jiang, Z. Cai, L. Hu, L. Zhang, and W. Lu, "Safe reinforcement learning-based motion planning for functional mobile robots suffering uncontrollable mobile robots," *IEEE Transactions on Intelligent Transportation Systems*, vol. 25, no. 5, pp. 4346–4363, 2024.

[12] B. Thananjeyan, A. Balakrishna, S. Nair, M. Luo, K. Srinivasan, M. Hwang, J. E. Gonzalez, J. Ibarz, C. Finn, and K. Goldberg, "Recovery rl: Safe reinforcement learning with learned recovery zones," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 4915–4922, 2021.

[13] H. Oh and T. Matsubara, "Leveraging demonstrator-perceived precision for safe interactive imitation learning of clearance-limited tasks," *IEEE Robotics and Automation Letters*, vol. 9, no. 4, pp. 3387–3394, 2024.

[14] N. Sojib and M. Begum, "Self supervised detection of incorrect human demonstrations: A path toward safe imitation learning by robots in

the wild," in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2024, pp. 2862–2869.

[15] S. Mamedov, R. Reiter, S. M. B. Azad, R. Viljoen, J. Boedecker, M. Diehl, and J. Swevers, "Safe imitation learning of nonlinear model predictive control for flexible robots," in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2024, pp. 3613–3619.

[16] X. Kong, Y. Xia, Z. Sun, D.-H. Zhai, Y. Deng, and S. Zhang, "Differential high order control barrier function-based safe reinforcement learning," *IEEE Robotics and Automation Letters*, vol. 10, no. 7, pp. 7524–7531, 2025.

[17] Nilaksh, A. Ranjan, S. Agrawal, A. Jain, P. Jagtap, and S. Kolathaya, "Barrier functions inspired reward shaping for reinforcement learning," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 2024, pp. 10 807–10 813.

[18] Z. Marvi and B. Kiumarsi, "Safe reinforcement learning: A control barrier function optimization approach," *International Journal of Robust and Nonlinear Control*, vol. 31, no. 6, pp. 1923–1940, 2021.

[19] W. Xiao, T.-H. Wang, R. Hasani, M. Chahine, A. Amini, X. Li, and D. Rus, "Barriernet: Differentiable control barrier functions for learning of safe robot control," *IEEE Transactions on Robotics*, vol. 39, no. 3, pp. 2289–2307, 2023.

[20] W. Xiao, T.-H. Wang, C. Gan, and D. Rus, "Abnet: Adaptive explicit-barrier net for safe and scalable robot learning," in *Forty-second International Conference on Machine Learning*.

[21] C. Zhang, L. Dai, H. Zhang, and Z. Wang, "Control barrier function-guided deep reinforcement learning for decision-making of autonomous vehicle at on-ramp merging," *IEEE Transactions on Intelligent Transportation Systems*, vol. 26, no. 6, pp. 8919–8932, 2025.

[22] B. Dai, P. Krishnamurthy, and F. Khorrami, "Learning a better control barrier function," in *2022 IEEE 61st Conference on Decision and Control (CDC)*, 2022, pp. 945–950.

[23] A. Robey, H. Hu, L. Lindemann, H. Zhang, D. V. Dimarogonas, S. Tu, and N. Matni, "Learning control barrier functions from expert demonstrations," in *2020 59th IEEE Conference on Decision and Control (CDC)*. Ieee, 2020, pp. 3717–3724.

[24] H. Yu, S. Farrell, R. Yoshimitsu, Z. Qin, H. I. Christensen, and S. Gao, "Estimating control barriers from offline data," *arXiv preprint arXiv:2503.10641*, 2025.

[25] A. E. Chriat and C. Sun, "On the optimality, stability, and feasibility of control barrier functions: An adaptive learning-based approach," *IEEE Robotics and Automation Letters*, vol. 8, no. 11, pp. 7865–7872, 2023.

[26] C. Dawson, B. Lowenkamp, D. Goff, and C. Fan, "Learning safe, generalizable perception-based hybrid control with certificates," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 1904–1911, 2022.

[27] O. So, Z. Serlin, M. Mann, J. Gonzales, K. Rutledge, N. Roy, and C. 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)*, 2024, pp. 11 532–11 539.

[28] P.-F. Massiani, S. Heim, F. Solowjow, and S. Trimpe, "Safe value functions," *IEEE Transactions on Automatic Control*, vol. 68, no. 5, pp. 2743–2757, 2023.

[29] D. C. Tan, R. McCarthy, F. Acero, A. M. Delfaki, Z. Li, and D. Kanoulas, "Safe value functions: Learned critics as hard safety constraints," in *2024 IEEE 20th International Conference on Automation Science and Engineering (CASE)*, 2024, pp. 2441–2448.

[30] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[31] J. Zeng, B. Zhang, and K. Sreenath, "Safety-critical model predictive control with discrete-time control barrier function," in *2021 American Control Conference (ACC)*, 2021, pp. 3882–3889.

[32] A. D. Ames, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs with application to adaptive cruise control," in *53rd IEEE Conference on Decision and Control*, 2014, pp. 6271–6278.

[33] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs for safety critical systems," *IEEE Transactions on Automatic Control*, vol. 62, no. 8, pp. 3861–3876, 2017.

[34] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, "Control barrier functions: Theory and applications," in *2019 18th European Control Conference (ECC)*, 2019, pp. 3420–3431.

[35] D. C. Tan, F. Acero, R. McCarthy, D. Kanoulas, and Z. Li, "Value functions are control barrier functions: Verification of safe policies using control theory," *arXiv e-prints*, pp. arXiv–2306, 2023.

[36] R. Cimurs, I. H. Suh, and J. H. Lee, "Goal-driven autonomous exploration through deep reinforcement learning," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 730–737, 2022.

[37] S. Fujimoto, H. Hoof, and D. Meger, "Addressing function approximation error in actor-critic methods," in *International conference on machine learning*. PMLR, 2018, pp. 1587–1596.