One Filter to Deploy Them All: Robust Safety for Quadrupedal Navigation in Unknown Environments

Albert Lin[1,2], Shuang Peng[1], and Somil Bansal[2]
[1]University of Southern California [2]Stanford University
Project Website: https://sia-lab-git.github.io/One_Filter_to_Deploy_Them_All

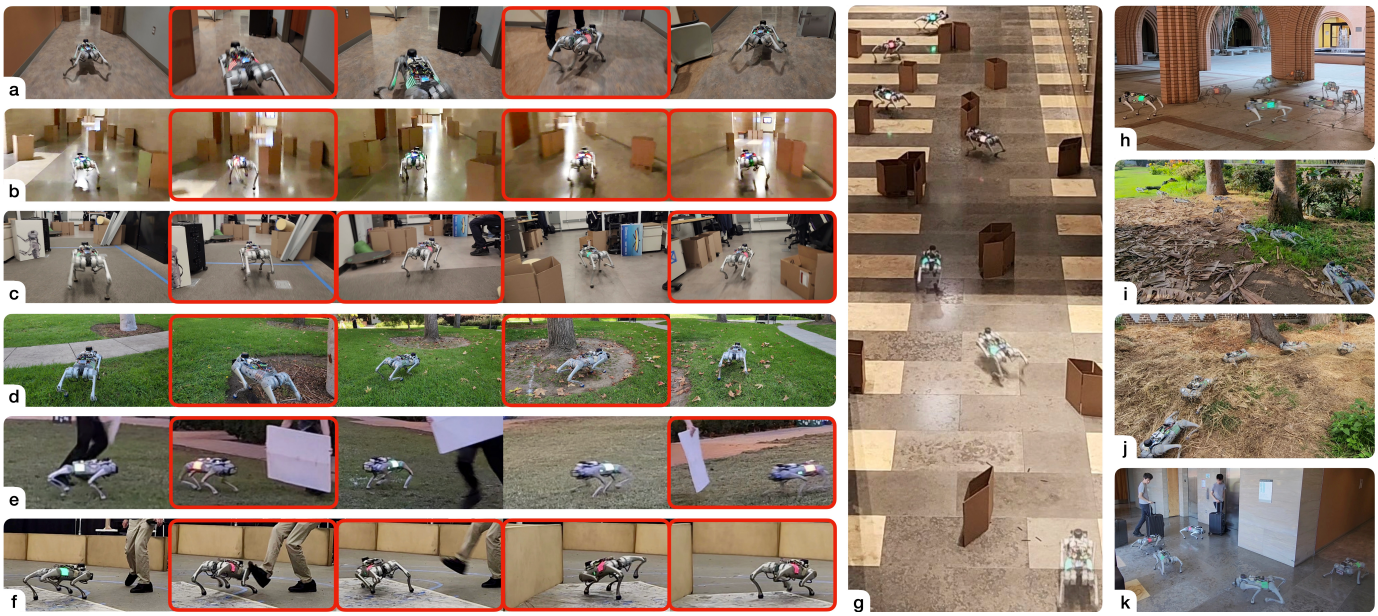

Fig. 1: Our proposed observation-conditioned reachability-based (OCR) safety-filter framework automatically safeguards different controllers in diverse settings *without a priori access to the controllers or environments*. A trained OCR value network governs the switch between nominal and filtered control using an onboard LiDAR sensor. The framework successfully safeguards a variety of high-level planners, including (a) learning-based, (c, f, k) model-based, (b, d, g, h, i, j) human teleoperated, and (e) naive planners, on top of different low-level locomotion policies, including (a, f, i, j, k) learning-based and (b, c, d, e, g, h) model-based policies. Safety is maintained despite (a, b, c) narrow corridors, (d, i, j) rough terrains, (e, k) dynamic obstacles, (f) external disturbances, and (h) collision-seeking human teleoperation.

***Abstract*—As learning-based methods for legged robots rapidly grow in popularity, it is important that we can provide safety assurances efficiently across different controllers and environments. Existing works either rely on *a priori* knowledge of the environment and safety constraints to ensure system safety or provide assurances for a specific locomotion policy. To address these limitations, we propose an observation-conditioned reachability-based (OCR) safety-filter framework. Our key idea is to use an OCR value network (OCR-VN) that predicts the optimal control-theoretic safety value function for new failure regions and dynamic uncertainty during deployment time. Specifically, the OCR-VN facilitates rapid safety adaptation through two key components: a LiDAR-based input that allows the dynamic construction of safe regions in light of new obstacles and a disturbance estimation module that accounts for dynamics uncertainty in the wild. The predicted safety value function is used to construct an adaptive safety filter that overrides the nominal quadruped controller when necessary to maintain safety. Through simulation studies and hardware experiments on a Unitree Go1 quadruped, we demonstrate that the proposed framework can automatically safeguard a wide range of hierarchical quadruped controllers, adapts to novel environments, and is robust to unmodeled dynamics *without a priori access to the controllers or environments* - hence, "One Filter to Deploy Them All". The experiment videos can be found on the project website.**

***Index Terms*—Hamilton-Jacobi reachability analysis, safety filtering, adaptive safety, robust verification, safe legged locomotion.**

I. Introduction

Legged robots hold immense potential across diverse real-world applications, such as hazardous inspections [1], [2], search and rescue missions [3], [4], entertainment [5], [6], and public safety [7]. A fundamental requirement in these scenarios is the ability to operate reliably in cluttered and *a priori unknown* environments. However, achieving this reliability poses a significant challenge, as legged locomotion controllers must balance high performance with safety (e.g., avoiding collisions) during deployment. This work focuses on designing controllers that enable safe, collision-free quadrupedal locomotion while maintaining agility in novel environments.

Existing approaches to designing (safe) controllers for legged locomotion can be broadly categorized into model-based and reinforcement learning (RL)-based methods. Model-based methods provide provable safety guarantees using

frameworks such as model predictive control (MPC), barrier functions, and reachability analysis [8]–[12]. However, model mismatches and online computational burden limit the applicability of these approaches in the wild. In contrast, RL-based controllers have demonstrated impressive agility in complex terrains and unstructured environments [13]–[19], [19]–[22] However, these controllers typically prioritize agility, treating collision avoidance as a soft constraint during training, which can result in unsafe behaviors in cluttered or unseen environments.

To address these shortcomings, recent works have explored safety critics and backup policies to safeguard RL-based controllers [23]–[30]. These approaches precompute or learn safety critics that indicate when the nominal policy is deemed unsafe. While promising, existing methods often fail to ensure safety beyond the distribution of dynamics, environments, or locomotion policies encountered during training. Additionally, learning reliable backup policies for unknown environments remains a persistent challenge.

In this work, we propose an Observation-Conditioned Reachability (OCR) safety-filter framework, designed to integrate the agility of a nominal robot policy with robust safety in cluttered and unknown environments. Our key idea is to use an OCR Value Network (OCR-VN), which predicts the optimal control-theoretic safety value function for new failure regions and dynamic uncertainties encountered during deployment. Specifically, the proposed framework achieves this adaptability through two key components: first, it leverages an observation-based exteroceptive input (a LiDAR scan in this work) to dynamically adapt the safety value function directly from raw sensory inputs, enabling robust collision avoidance in different scenarios with onboard sensing and computation. Second, it employs a disturbance estimation module to compute bounds on dynamics uncertainty (e.g., due to slippage, modeling inaccuracies, or low-level tracking errors) using recent state-action histories and adapts the robustness of the safety value function based on these bounds.

The OCR framework predicts when a nominal policy might violate safety and provides corrective control commands for the robot if necessary. A key strength of the proposed framework is its generality – it can be deployed with a wide variety of nominal legged locomotion policies without requiring any retraining or policy-specific tuning. Additionally, we propose a Hamilton-Jacobi reachability-based method to train the OCR-VN, ensuring robust and efficient safety filtering. We validate our approach through extensive simulations and real-world experiments on a Unitree Go1 quadruped, demonstrating that the OCR framework provides a reliable safety layer across multiple existing legged locomotion policies (both model-based and learning-based) and a variety of environments, without requiring prior knowledge of the specific policy or environment. In summary, our key contributions are:

- A reachability-based safety-filtering framework that ensures safety across diverse quadruped controllers and environments, without a priori access to the controller or the environment;
- An online adaptation mechanism that dynamically adapts the system safety to real-world environment variations and modeling uncertainties;
- Simulation and hardware experiments demonstrating the superior efficacy and robustness of the proposed approach in ensuring safe legged locomotion.

II. Related Works

A. Safe Legged Locomotion

1) Model-Based Safety: Traditional approaches for obstacle avoidance use collision-free motion planning techniques in the configuration space [31]–[33]. They satisfy kinematic safety constraints but do not consider the dynamics of the system, limiting motions to slow quasi-static trajectories. However, recent advances in agile locomotion and its applications have resulted in the need to consider dynamics.

Model-based approaches, such as model predictive control (MPC), use hand-designed or learned dynamics models to compute optimal maneuvers that are dynamically feasible and satisfy safety constraints [26], [34]–[39]. Despite their impressive performance, such model-based approaches are generally computationally intensive for online settings and can run into safety feasibility issues, especially in cluttered obstacle environments. Additionally, although they often perform well in settings that are captured accurately by their models, the safety guarantees are not robust to model mismatches that a robot might encounter in the wild [13], [40].

2) RL-Based Safety: Given the challenges associated with existing model-based approaches for agile locomotion, model-free RL-based approaches have emerged as popular alternatives. RL-based approaches have found remarkable success in synthesizing efficient and robust locomotion in the real world, especially as the availability of high-fidelity simulators has increased [41], [42]. They are well-suited to handle complex high-dimensional systems, multimodal feedback signals, and difficult-to-specify task objectives [43].

Previous studies have optimized locomotion policies for specific skills such as agility [44], [45], resilience [46]–[48], and difficult terrain traversal [13], [20], [21], [49]–[55]. However, these works typically focus on maximizing agility without regard to safe navigation. These methods can be combined with high-level collision-free planners; however, they suffer from the aforementioned limitations of model-based controllers and restricted mobility [29], [56]–[58].

Other works consider safe navigation during the learning process by including a large collision penalty in the reward function to incentivize collision-avoidance [14], [57], [59]–[66]. Unfortunately, there are no formal guarantees of safety, and the synthesized locomotion policies can degrade in safety when transferred to the real world due to a distribution shift away from the environments seen during training.

3) Certificate-Based Safety: In order to provide rigorous safety assurances, many works have proposed certificate-based safety methods within both model-based [26], [34], [35], [37], [39] and RL-based [23]–[25], [27]–[30], [67] frameworks, most often using control barrier functions or reachability-based value functions. These methods are typically reliant on the offline availability of a certificate function or dynamics, which limits their applicability to complex real-world systems.

Some recent works learn adaptive safety certificates and recovery policies to ensure safety at runtime [23]–[25], [27]–[29], [67]. He et al. [29] propose Agile But Safe (ABS), an approach that co-designs performance and control-theoretic safety controllers via RL and switches between them as necessary to maintain safety. Since the computed safety assurances are policy-dependent, they can be suboptimal and overly conservative, depending on the quality of the nominal policy. Moreover, the safety controller needs to be retrained as the policy is fine-tuned, which can be cumbersome and time-consuming. Additionally, since approaches like ABS lack active mechanisms to account for unmodeled dynamics during deployment (e.g., slippage), the computed safety assurances are valid only within the distribution of environments and locomotion policies seen during training. These limitations are especially relevant due to the rising popularity of diverse learning-based policies, whose safety assurances must be updated as they are fine-tuned.

Our work most closely aligns with certificate-based safety methods like ABS, but we address several existing limitations. Instead of computing the policy-conditioned safety function, we compute the optimal control-theoretic safety controller via Hamilton-Jacobi (HJ) reachability analysis. This enables us to construct a safety filter for any nominal controller without needing to recompute the safety assurances. We discard the need for high-fidelity simulators by using a reduced-order system and robustly handle the model gap as an adversarial disturbance in the dynamics. By estimating and adapting to disturbance bounds during deployment, our proposed framework is able to ensure safety more robustly across a range of settings and policies compared to previous works.

B. Reachability-Based Safety Filters

This work extends the class of Hamilton-Jacobi (HJ) reachability-based filters [68]–[70] which ensure safety at deployment by overriding a nominal controller when necessary to preserve safety. Due to the computational burden, traditional reachability-based safety filters are typically constructed offline for systems assumed to be known a priori [71]. In the case of quadrupedal navigation, a quadruped using a specific locomotion policy is often abstracted as a reduced-order system for computational tractability. The dynamics of the system depend on the locomotion policy; thus, reachability-based filters will not readily safeguard different locomotion policies. Additionally, it is challenging to adapt safety assurances to different obstacle configuration and terrain properties due to the computational burden [72], [73]. This work overcomes these limitations by distilling the safety solutions for a wide range of system settings into an Observation-Conditioned Reachability-based Value Network (OCR-VN). During deployment, the robot adapts by querying the trained OCR-VN with the current state, control, and observation history.

III. Problem Setup

See Table I for notation. In this work, we are interested in ensuring the safety of a quadruped robot in an *a priori unknown* environment $e \in \mathcal{E}$. Here, e contains all the information needed to inform the effects of the environment on dynamics, as well as failure regions. For example, e can include the terrain geometry, friction coefficients, and obstacle locations.

TABLE I: Nomenclature

Symbol	Definition.
t, T	Time, time horizon.
$e \in \mathcal{E}$	Environment.
$x \in \mathcal{X}$	State.
$u \in \mathcal{U}$	Control input.
$d \in \mathcal{D}$	Disturbance input.
f	System dynamics.
π, π^{high}, π^{low}	Hierarchical, high-level, and low-level policies.
$\xi^{\pi}_{x,t}(\tau)$	State achieved at time τ by starting at initial state x at time t and applying policy π over $[t, \tau]$.
$\mathcal{F}$, l	Failure set and function.
$o \in \mathcal{O}$	Observation.
$x_r \in \mathcal{X}_r \subseteq \mathcal{X}$, f_r	Reduced-order state and dynamics.
$\omega = (v, w)$	Twist ω consisting of velocity v and yaw rate w.
$\bar{d}_{p_x,p_y}$, $\bar{d}_{p_\theta}$	Disturbance bound in p_x, p_y and in p_θ.
$\bar{d}_r = [\bar{d}_{p_x,p_y}, \bar{d}_{p_\theta}]$	Disturbance bound for reduced-order system.
V, V_ψ	Ground-truth and learned value functions.

Abbreviation	Definition.
OCR	Observation-conditioned reachability.
ABS	Agile But Safe [29].
WTW	Walk-These-Ways [21].
MPC	Model predictive control policy by Unitree [74].
PS	Predictive sampling-based planner.
NVE	Naive planner.
HMN	Human-teleoperated planner.

We model the quadruped as a nonlinear dynamical system with state $x \in \mathcal{X}$, control $u \in \mathcal{U}$, and dynamics $\dot{x} = f(x, u; e)$ governing how x evolves over time until a final time horizon T. The dynamics are also affected by the environment e, e.g., by the effects of a slippery floor. We denote the robot observations from proprioception and/or exteroception as $o^e_x = h(x; e) \in \mathcal{O}$. For exteroception, we primarily deal with LiDAR scans in this work, though other sensors can also be used. We denote the set of failure states as $\mathcal{F}^e \subseteq \mathcal{X}$ (e.g., collision states) which the robot is not allowed to enter. The failure set can be represented by the zero-sublevel set of a Lipschitz-continuous function $l^e : \mathcal{X} \to \mathbb{R}$, i.e., $x \in \mathcal{F}^e \Leftrightarrow l^e(x) \leq 0$. Note that $\mathcal{F}^e, l^e$ are also functions of e.

Let π_{nom} denote a hierarchical nominal policy for the quadruped that takes the system history and outputs the robot control. We assume that π_{nom} consists of a high-level planner, $\pi^{\text{high}}_{\text{nom}}$, that provides twist commands $\omega := (v, w)$ consisting of a forward velocity v and a yaw rate w. These twist commands are tracked by a low-level locomotion policy, $\pi^{\text{low}}_{\text{nom}}$, e.g., an RL-based policy [21], [29], [44] or an MPC-based policy [32], [74], that ultimately provides control inputs u for the robot. This architecture is popular in the legged robotics literature where $\pi^{\text{high}}_{\text{nom}}$ is typically designed for collision avoidance and navigation, and $\pi^{\text{low}}_{\text{nom}}$ could be an agile locomotion policy that can handle different terrains that the robot might encounter in the wild.

Let $\xi^{\pi}_{x,e,t}(\tau)$ denote the state achieved at time $\tau \in [t, T]$ by starting at initial state x at time t and applying the control policy π over $[t, \tau]$ in environment e. Our goal is to compute a safe policy π_{safe} that ensures that the quadruped remains outside of the failure set at all times, i.e., $\pi_{\text{safe}} : \forall \tau \in [0, T], \xi^{\pi_{\text{safe}}}_{x,e,0}(\tau) \notin \mathcal{F}^e$, while preserving the underlying performance of π_{nom} to the extent possible. The key challenge

in designing such a policy is that the robot environment (and hence the failure set and the robot dynamics) is not known beforehand, necessitating a real-time update of the safety policy with the environment. A second challenge stems from the fact that we want the safety framework to be agnostic to different nominal policies.

IV. Background

A. Hamilton-Jacobi Reachability

Our proposed framework builds upon Hamilton-Jacobi (HJ) reachability analysis, which is a popular formal verification tool for computing safety guarantees for general nonlinear dynamical systems [75], [76]. For reachability analysis, we will consider a more general form of dynamics $\dot{x} = f(x, u, d)$, where $d \in \mathcal{D}$ represents the disturbance. Later on, in our work, we will use d to model potential uncertainty in the system dynamics model. We also omit the dependence on environment for now for brevity purposes.

HJ reachability analysis is concerned with computing the system's initial-time Backward Reachable Tube, which we denote as BRT. We define BRT as the set of all initial states $x \in \mathcal{X}$ starting from which, for all control signals $\mathbf{u}(\cdot)$, there exists a disturbance signal $\mathbf{d}(\cdot)$ such that the system will inevitably enter the failure set $\mathcal{F}$ within the time horizon $[0, T]$:

$$\text{BRT} \coloneqq \{x \in \mathcal{X} : \forall \mathbf{u}(\cdot), \exists \mathbf{d}(\cdot), \exists \tau \in [0, T], \xi_{x,0}^{\mathbf{u}(\cdot),\mathbf{d}(\cdot)}(\tau) \in \mathcal{F}\}. \tag{1}$$

By ensuring that the system remains outside of BRT, we guarantee system safety for the time horizon T.

In HJ reachability, computing BRT is formulated as a robust optimal control problem. First, we implicitly represent the failure set $\mathcal{F}$ by a failure function $l(x)$ whose zero-sublevel set yields $\mathcal{F}$: $\mathcal{F} = \{x \in \mathcal{X} : l(x) \leq 0\}$. $l(x)$ is commonly the signed distance function to $\mathcal{F}$. Next, we define the cost function corresponding to a control signal $\mathbf{u}(\cdot)$ and disturbance signal $\mathbf{d}(\cdot)$ to be the minimum of $l(x)$ over the trajectory starting from state x and time t:

$$J_{\mathbf{u}(\cdot),\mathbf{d}(\cdot)}(x, t) \coloneqq \min_{\tau \in [t,T]} l(\xi_{x,t}^{\mathbf{u}(\cdot),\mathbf{d}(\cdot)}(\tau)). \tag{2}$$

Since the control aims to avoid $\mathcal{F}$ under worst-case disturbance, the value function corresponding to this robust optimal control problem is:

$$V(x, t) \coloneqq \max_{\mathbf{u}(\cdot)} \min_{\mathbf{d}(\cdot)} J_{\mathbf{u}(\cdot),\mathbf{d}(\cdot)}(x, t). \tag{3}$$

By defining our optimal control problem in this way, we can easily recover BRT using the value function. The value function being nonpositive implies that the failure function is nonpositive somewhere along the optimal trajectory, or in other words, that the system will inevitably enter $\mathcal{F}$. Conversely, the value function being positive implies that there exists a control signal that will prevent the system from entering $\mathcal{F}$ even under the worst-case disturbance signal. Thus, BRT is computed as the zero-sublevel set of the initial-time value function:

$$\text{BRT} = \{x \in \mathcal{X} : V(x, 0) \leq 0\}. \tag{4}$$

The value function in Equation (3) can be computed using dynamic programming, resulting in the following final value Hamilton-Jacobi-Isaacs Variational Inequality (HJI-VI):

$$\begin{aligned} \min\{D_t V(x,t) + H(x, t, \nabla V(x,t)), l(x) - V(x,t)\} = 0, \\ V(x, T) = l(x), \quad \forall t \in [0, T]. \end{aligned} \tag{5}$$

$D_t V(x,t)$ and $\nabla V(x,t)$ represent the temporal derivative and spatial gradient of the value function $V(x,t)$, respectively. The Hamiltonian $H(x, t, \nabla V(x,t))$ encodes how the control and disturbance interact with the system dynamics:

$$H(x, t, \nabla V(x,t)) \coloneqq \max_{u \in \mathcal{U}} \min_{d \in \mathcal{D}} \nabla V(x,t) \cdot f(x, u, d). \tag{6}$$

The value function in Equation (3) also induces the optimal safety controller:

$$\mathbf{u}^*(x, t) \coloneqq \arg\max_{u \in \mathcal{U}} \min_{d \in \mathcal{D}} \nabla V(x,t) \cdot f(x, u, d). \tag{7}$$

Intuitively, the optimal safety controller aligns the system dynamics in the direction of the value function's gradients, thus steering the system towards higher-value states, i.e., away from $\mathcal{F}$. An important result of HJ reachability theory is that safety is guaranteed despite worst-case disturbances if the system starts outside of BRT and applies the control in Equation (7) at the BRT boundary [68].

B. HJ Reachability-Based Safety Filtering

In this work, we will use HJ reachability analysis to maintain the safety of the quadruped robot during deployment via HJ reachability-based safety filtering, where the computed safety value function is used to construct a safety filter that guarantees system safety. While there are many different types of safety filters that can be constructed, we use the smooth least-restrictive safety filter, which aims to maximally preserve the underlying performance of a given nominal policy π_{nom} by intervening as seldomly and as lightly as possible [68]. When outside of the system BRT, the smooth least-restrictive filter π_{safe} outputs the nominal control. At the boundary of the BRT, π_{safe} outputs a safe control as close as possible to the nominal control by solving a quadratic program (QP).

$$\pi_{\text{safe}}(x, t) = \begin{cases} \pi_{\text{nom}}(x, t), & V(x,t) > 0 \\ \pi_{\text{QP}}(x, t), & V(x,t) = 0, \end{cases} \tag{8}$$

where $\pi_{\text{QP}}(x, t)$ is obtained by solving:

$$\begin{aligned} &\arg\min_{u \in \mathcal{U}} ||u - \pi_{\text{nom}}(x,t)||_2^2 \\ &\text{s.t. } D_t V(x,t) + \min_{d \in \mathcal{D}} \nabla V(x,t) \cdot f(x, u, d) = 0. \end{aligned} \tag{9}$$

Intuitively, the constraint in (9) enforces safe control of the system at the BRT boundary, where the system safety could be in jeopardy. It has been shown that the filter in 8 is guaranteed to maintain system safety under worst-case disturbances as long as the system starts outside of the BRT [68]. In the next section, we propose to construct an adaptive version of this safety filter by using an Observation-Conditioned Reachability-based Value Network (OCR-VN) that predicts the safety value function for new failure regions and dynamic uncertainties encountered during deployment.

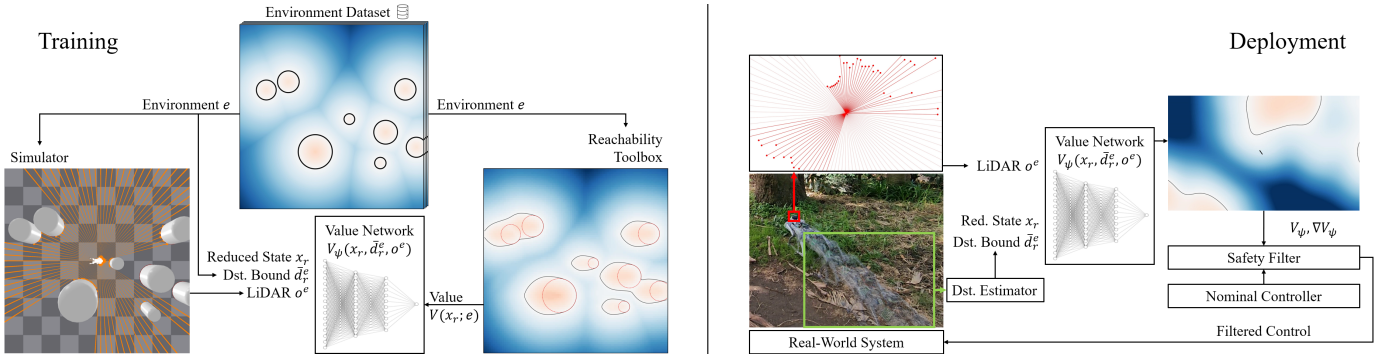

Fig. 2: The OCR framework. (Left) During training, we generate environments with random obstacles and disturbance bounds. The OCR-VN is trained to predict the value function (visualized over a grid) using the disturbance bound, the LiDAR reading, and the state. (Right) During deployment, the OCR-VN is queried with the observed LiDAR reading, the disturbance bound estimated using the most recent state and action history, and the current state estimate to construct an adaptive safety filter.

V. Approach

The main difficulty in directly applying traditional HJ reachability methods in Section IV to the problem in Section III is that the system dynamics $f(x, u; e)$ and the failure set $\mathcal{F}^e$ are functions of the unknown environment e. Thus, a safety controller must be able to adapt to both *dynamics uncertainty* and *environment uncertainty*. To overcome this issue, we propose the Observation-Conditioned Reachability (OCR) safety-filter framework.

The key idea behind our framework is to use an OCR Value Network (OCR-VN) that predicts the optimal safety value function for new dynamics $f(x, u; e)$ and failure regions $\mathcal{F}^e$ using the most recent system and observation history. The OCR-VN facilitates rapid adaptivity through two key components: an observation-based input to the network (see Section V-A3) and a disturbance estimation module (see Section V-B1). The observation-based input allows the OCR-VN to perceive new obstacles in the environment and dynamically adapt the value function to safeguard against them; whereas the disturbance estimation module allows the OCR-VN to tune the degree of uncertainty that is present within the dynamics model based on the most recent system history, and correspondingly adapt the safety value function to this dynamics uncertainty. The resultant safety value function is then used to filter the nominal policy to ensure safety while maintaining the agility of the underlying policy. Ultimately, our safety framework provides high-level twist commands, which, when combined with the low-level nominal policy, maintains the robot's safety.

The OCR framework can be divided into two distinct phases: training and deployment. These are illustrated in Figure 2. During the training phase, we collect a dataset across different environments to train the OCR-VN to predict the safety value function, directly from raw observations and uncertainty bounds. During the deployment phase, we query the OCR-VN with the onboard observation and an estimate of the dynamical uncertainty to construct an adaptive safety filter across different locomotion policies and environments. The OCR framework is described in more detail next.

A. Training Phase of the OCR Framework

1) System Dynamics Model: During the training phase, we aim to distill the ground-truth value functions for a diverse range of environments e into an OCR-VN. The first difficulty that we encounter is how to model the full quadruped dynamics $f(x, u; e)$, which is high-dimensional and complex. Our key insight is that many quadruped control schemes are hierarchically composed of a high-level navigation planner and a low-level locomotion policy. From the perspective of the high-level planner, the quadruped, along with its locomotion policy, form a system with a *reduced-order* dynamics model. Thus, we propose to use a reduced-order dynamics model $f_r(x_r, \omega)$, where $x_r \in \mathcal{X}_r \subseteq \mathcal{X}$ is the reduced state and ω is the control input of the reduced-order model. We capture any possible modeling errors of $f_r(x_r, \omega)$ as disturbances in the system. We propose to estimate bounds on these disturbances during deployment for adaptive safety guarantees.

We set $f_r(x_r, \omega)$ to the dynamics of a 3D Dubins car system with state $x_r = (p_x, p_y, p_\theta)$, where (p_x, p_y) is the quadruped's 2D location, and p_θ is the quadruped's heading. We model the error in $f_r(x_r, \omega)$ as an unknown additive disturbance d_r^e, which is a function of the underlying environment e. Thus, the reduced-order dynamics $f_r(x_r, \omega, d_r^e)$ is given as:

$$\dot{p_x} = v \cos p_\theta + d_{p_x}^e, \quad \dot{p_y} = v \sin p_\theta + d_{p_y}^e, \quad \dot{p_\theta} = w + d_{p_\theta}^e, \tag{10}$$

with the control input $\omega = (v, w)$ being a commanded twist that includes forward velocity $v \in [v_{\min}, v_{\max}]$ and yaw rate $w \in [w_{\min}, w_{\max}]$. The disturbance input $d_r^e = \left(d_{p_x}^e, d_{p_y}^e, d_{p_\theta}^e\right)$ consists of a bounded additive disturbance in position $\|[d_{p_x}^e, d_{p_y}^e]\| \leq \bar{d}_{p_x p_y}^e$ and a bounded additive disturbance in heading $|d_{p_\theta}^e| \leq \bar{d}_{p_\theta}^e$. We use $\bar{d}_r^e$ to denote the disturbance bound tuple $\bar{d}_r^e = \left(\bar{d}_{p_x, p_y}^e, \bar{d}_{p_\theta}^e\right)$, which is a function of the underlying environment e. Since the disturbances are additive in all state variables, the disturbance bounds can always be chosen large enough to contain any possible modeling errors, although at the cost of model conservatism. In order to be robust to the modeling error, we assume that the disturbances are adversarial in nature.

Due to the choice of state variables in the reduced-order model, we are restricted to safety constraints specified in terms of the quadruped's 2D location and 1D orientation. This is sufficient for our work, since we focus exclusively on safe navigation. We remark that other f_r and disturbance assumptions can be chosen depending on the desired performance, conservatism, and computational complexity.

2) Data Generation: Based on our modeling choices in Section V-A1, we let the underlying environment $e = \left(\mathcal{F}^e, \bar{d}_r^e\right)$ consist of the failure set $\mathcal{F}^e$ and the disturbance bound $\bar{d}_r^e$ described in Section V-A1. We generate each environment e by randomly spawning 2D obstacles at various locations and sampling a disturbance bound. These disturbance bounds will correspond to the model uncertainties considered during the deployment phase and will be estimated online (Section V-B1).

For each of the generated environments, we compute the initial-time ground-truth value function $V(x_r, 0; e)$ using the `hj_reachability` Python toolbox [77], which we denote as $V(x_r; e)$ for brevity. We compute the converged value function, that is, when we take the time horizon $T \to \infty$. This ground-truth value function is then used to generate data for training the OCR-VN.

Specifically, for each training environment, we generate training pairs $\left(\left(x_r, \bar{d}_r^e, o^e\right), V\left(x_r; e\right)\right)$ by setting the system origin to different states in the environment and rendering the corresponding observation o^e at the origin. For each system origin and its corresponding observation o^e, we sample the value function $V\left(x_r; e\right)$ and its spatial gradients $\nabla V\left(x_r; e\right)$ at uniformly random states x_r (specified in the frame of the new system origin) that are not occluded by obstacles. The collected data is then used to train the OCR-VN via supervised learning as described in the next section. Details on the specific parameters of our data generation are provided next.

To generate an environment, we spawn n circular 2D obstacles, where n is uniformly sampled from $\{1, ..., 10\}$. The radius r of each obstacle is uniformly sampled from $[0.1, 1]$ m. The location (p_x, p_y) of each obstacle is uniformly sampled from $[-5, 5]$ m $\times$ $[-5, 5]$ m. Disturbance bounds $\bar{d}_{p_x,p_y}^e$ and $\bar{d}_{p_\theta}^e$ are uniformly sampled from $[0, 1]$ m/s and $[0, 2]$ rad/s, respectively. In this work, we use LiDAR observations which consists of 100 evenly spaced angles from $[-\pi, \pi]$ rad and clipped to within $[0.2, 10]$ m, determined by hardware limits.

For obtaining the ground-truth value function, we use a grid of shape $(100, 100, 60)$ spanning $[-5, 5]$ m $\times$ $[-5, 5]$ m $\times$ $[-\pi, \pi]$ rad and a time horizon of 2 s, when the value function approximately converges. The control bounds for the system are $v \in [0, 2]$ m/s and $|w| \leq 2$ rad/s. We generate a total of $1,000$ training environments and 100 validation environments. Generating the datasets takes roughly 35 minutes on an NVIDIA 3090Ti GPU. The ground-truth value function $V(x_r; e)$ and the corresponding LiDAR observation o^e for a validation environment are shown in Figure 3.

We generate a training batch by first sampling 10 training environments. For each environment, we set the system origin to 10 different states sampled outside of the obstacle set and capture the corresponding LiDAR observation, resulting in 10 different egocentric observations. This is done to increase the diversity of LiDAR observations seen during training without substantially increasing the computational effort, which is important because the observations are a high-dimensional network input. For each sampled system origin, we query the ground-truth value function and its spatial gradients at 500 random states that are not occluded by obstacles. This ultimately results in $N = 50,000$ samples per training batch.

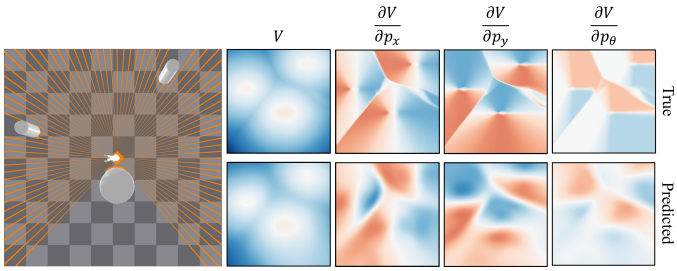

Fig. 3: (Left) A LiDAR observation o^e in a validation environment where $\bar{d}_{p_x,p_y}^e = 0.82$ m/s, $\bar{d}_{p_\theta}^e = 0.56$ rad/s. (Right top-row) The ground-truth value function and its spatial gradients. (Right bottom-row) OCR-VN predictions using o^e and $\bar{d}_r^e$. As shown above, the OCR-VN predictions for the value function and its spatial gradients are highly accurate.

3) OCR-VN Architecture and Training: We train an OCR-VN to predict $V(x_r; e)$ from the reduced system state x_r, disturbance bound $\bar{d}_r^e$, and the LiDAR observation o^e captured at the system origin. Let us denote the value predicted by the OCR-VN as $V_\psi(x_r, \bar{d}_r^e, o^e)$, where ψ denotes the learnable weights of the neural network. We choose V_ψ to be a multilayer perceptron with 4 hidden layers of 512 neurons each. We desire V_ψ to accurately model the spatial gradients of V as well, since they will be used in the safety filter construction. Thus, we use sinusoidal activations, which have been shown to lead to more effectively gradient modeling than ReLU activations [78].

For each training batch as described in Section V-A2, we minimize the MSE training loss:

$$\frac{1}{N}\sum_{i=1}^{N}(||V_\psi^i(x_r, \bar{d}_r^e, o^e) - V^i(x_r; e)||_2^2 + ||\nabla_x V_\psi^i(x_r, \bar{d}_r^e, o^e) - \nabla_x V^i(x_r; e)||_2^2)$$

which includes a loss term for both the value and its spatial gradients. We use the Adam optimizer with a learning rate of 10^{-5}. Training converges in roughly 8 hours on an NVIDIA 3090Ti GPU. As shown in Figure 3, the OCR-VN can accurately model the value function and its spatial gradients for a validation environment using a single LiDAR observation o^e.

4) OCR-VN Calibration: The OCR-VN will inevitably contain learning errors that can be critical for safety. To safeguard against these errors, we calibrate the OCR-VN by shifting its output by a probabilistic error bound δ computed on the validation dataset. We will refer to δ as the "calibration level" and the shifted network output as the "calibrated output". We employ conformal prediction, a popular uncertainty quantification tool in the machine learning literature, to compute δ up to a desired confidence β and violation rate ϵ [79], [80].

Our procedure is as follows. Select a desired confidence $\beta \in (0, 1)$ and violation rate $\epsilon \in (0, 1)$. Sample N calibration

TABLE II: Calibration Levels for the OCR-VN with $N = 10^7, \beta = 10^{-12}$

Violation Rate ϵ	10^{-1}	10^{-2}	10^{-3}
Calibration Level δ (m)	0.22	0.49	0.99

points according to a distribution $\mathbb{P}$ over the validation dataset Δ. Compute the conformal scores $\{s_i\}_{i=1}^N$ as the prediction errors $s_i = V_\psi^i\left(x_r, \bar{d}_r^e, o^e\right) - V^i\left(x_r; e\right)$. The following theorem provides a probabilistic bound on the OCR-VN error:

Theorem 1 (Conformal OCR-VN Calibration)**.** *Compute the number of "outliers" k as:*

$$\arg\max_k k : \sum_{i=0}^{k} \binom{N}{i} \epsilon^i (1-\epsilon)^{N-i} \leq \beta, \tag{11}$$

where β, ϵ, and N are as defined above. Compute the calibration level δ as the $\frac{N-k}{N}$ quantile of the conformal scores $\{s_i\}_{i=1}^N$. Then, with probability at least $1-\beta$ over the draws of the calibration samples, the following holds:

$$\mathop{\mathbb{P}}_{(x_r, e, \bar{d}_r^e, o^e) \in \Delta} \left(V_\psi(x_r, \bar{d}_r^e, o^e) - V(x_r; e) > \delta\right) \leq \epsilon \tag{12}$$

The proof of Theorem 1 is presented in Appendix A. Ignoring the confidence parameter β for a moment, Inequality (12) tells us that the volume of the validation dataset for which the OCR-VN overestimates the safety value by more than δ is bounded by the violation parameter ϵ. This enables us to be sure, up to a desired ϵ, that the true safety value is at least as large as that predicted by the OCR-VN, once adjusted by δ.

To interpret the confidence parameter β, note that the calibration level δ is a random variable that depends on the randomly sampled calibration set. It might be the case that we happen to draw an unrepresentative calibration set, in which case the ϵ bound does not hold. β controls the probability of this adverse event, which regards the correctness of the probabilistic guarantee given by Inequality (12). Fortunately, β goes to 0 exponentially with N, so β can be chosen sufficiently small, such as 10^{-12}, when we sample large N. $1-\beta$ will then be so close to 1 that it does not have any practical importance.

We set $N = 10^7, \beta = 10^{-12}$ and compute δ corresponding to various ϵ in Table II. We select $\delta = 0.49$ m associated with $\epsilon = 10^{-2}$ as a comfortable trade-off between conservatism and performance for the tasks in this work.

B. Deployment Phase of the OCR Framework

In this section, we describe the steps involved in deploying the OCR framework onto a quadruped robot. During deployment, the quadruped periodically receives a LiDAR observation o^e from an onboard sensor as a function of the environment e. Using a LiDAR-based state localization algorithm, the quadruped maintains an estimate of its reduced state x_r in the relative frame defined by o^e. The quadruped then estimates the disturbance bound $\bar{d}_r^e$ as a function of the most recent state and action history $\left(x_r^{i-k:i}, \omega^{i-k:i-1}\right)$ using an online disturbance bound estimation scheme. It queries the OCR-VN with the input $\left(x_r, \bar{d}_r^e, o^e\right)$ to construct an adaptive safety filter that minimally overrides the nominal controller when necessary to maintain safety. We describe the details of the online disturbance bound estimation scheme and the adaptive safety filter next.

1) Online Disturbance Bound Estimation: We propose to estimate the disturbance bound $\bar{d}_r^e$ using the most recent state and action history, to enable rapid adaptation to dynamical uncertainty. Let $x_r^{i-k:i}$ and $\omega^{i-k:i-1}$ denote a discrete history of estimated states and twist commands ending at time τ. To efficiently roll out the state trajectory in discrete time for online computation, we use Euler's method. Since disturbance is additive in $f_r(x_r, \omega, d_r^e)$ in Equation (10), we have:

$$\begin{aligned} x_r^i &\approx x_r^{i-k} + \sum_{j=i-k}^{i-1} \eta \cdot \left(f_r(x_r^j, \omega^j) + (d_r^e)^j\right) \\ &\approx x_r^{i-k} + \sum_{j=i-k}^{i-1} \eta \cdot f_r(x_r^j, \omega^j) + \eta \sum_{j=i-k}^{i-1} (d_r^e)^j, \end{aligned}$$

where η is the discrete time step. To ensure that we capture systemic disturbances in the dynamics and ignore transient noise in the state estimation, we assume that the disturbance input takes the form of a low-frequency signal and remains a constant d_r^τ throughout the short history:

$$x_r^i \approx x_r^{i-k} + \sum_{j=i-k}^{i-1} \eta \cdot f_r(x_r^j, \omega^j) + \eta \cdot k \cdot d_r^\tau.$$

Thus, we estimate the disturbance d_r^τ as:

$$d_r^\tau \approx \frac{x_r^i - \hat{x}_r^i}{\eta \cdot k}, \tag{13}$$

where $\hat{x}_r^i \coloneqq x_r^{i-k} + \sum_{j=i-k}^{i-1} \eta \cdot f_r(x_r^j, \omega^j)$ is the state predicted by disturbance-free dynamics. Intuitively, d_r^τ is the error in the state prediction normalized by the prediction time horizon $\eta \cdot k$. We set $\eta \cdot k = 2$ s in practice.

By computing d_r^τ over the most recent φ discrete time steps ending at the current time t, i.e., for $\tau = t - j \cdot \eta, \forall j \in \{0, 1, ..., \varphi - 1\}$, we construct a sliding window of estimated disturbances $d_r^{1:\varphi}$. To ensure good coverage of the estimated disturbances, we compute the disturbance bound $\bar{d}_r^e$ as:

$$\bar{d}_r^e = |\mu(\left(d_r^{1:\varphi}\right)_c)| \pm b \cdot \sigma(\left(d_r^{1:\varphi}\right)_c), \tag{14}$$

where the coverage parameter $c \in [0, 1]$ determines the middle fraction of the sorted window of disturbances $d_r^{1:\varphi}$ to be considered. This makes sure that the safety value function does not become overly conservative due to outlier disturbances. $\mu(\cdot)$ is the sample mean, and $\sigma(\cdot)$ is the sample standard deviation. $b \geq 0$ is the spread of disturbances, in standard deviation units, that we compute the disturbance bound for, modulating the degree of robustness of the disturbance bound. A larger choice of b results in a more robust disturbance bound. We set $c = 0.8$, $b = 2$, and $\eta \cdot \varphi = 2$ s in practice. We compute two separate disturbance bounds: $\bar{d}_{p_x, p_y}^e$ bounding the uncertainty in the position dynamics and $\bar{d}_{p_\theta}^e$ bounding the uncertainty in the yaw dynamics. For computing a bound on the norm of d_{p_x, p_y}^e, we use Equation (14) on a window of disturbance norms instead of raw values.

2) Adaptive Safety Filter: We use the OCR-VN to construct a smooth least-restrictive safety filter (see Section IV-B) safeguarding a potentially unsafe nominal controller that is given to us. Specifically, we compute a filtered high-level policy $\pi_{\text{safe}}^{\text{high}}$ which, when combined with the underlying low-level policy $\pi_{\text{nom}}^{\text{low}}$, maintains the overall system safety.

We compute the hierarchical filtered policy $\pi_{\text{safe}}(x, \bar{d}_r^e, o^e)$ as follows:

$$\pi_{\text{safe}}(x, \bar{d}_r^e, o^e) = \begin{cases} \pi_{\text{nom}}(x), & V_\psi(x_r, \bar{d}_r^e, o^e) > \delta \\ \pi_{\text{nom}}^{\text{low}} \circ \pi_{\text{QP}}^{\text{high}}(x_r, \bar{d}_r^e, o^e), & V_\psi(x_r, \bar{d}_r^e, o^e) \leq \delta, \end{cases} \tag{15}$$

where $\pi_{\text{QP}}^{\text{high}}(x_r, \bar{d}_r^e, o^e)$ is obtained by solving:

$$\begin{aligned} &\min_{\omega \in \Omega, s \geq 0} \left(||\omega - \pi_{\text{nom}}^{\text{high}}(x_r)||_2^2 + \lambda s^2 \right) \\ &\text{s.t.} \min_{d_r \in \mathcal{D}_r^e} \nabla V_\psi(x_r, \bar{d}_r^e, o^e) \cdot f_r(x_r, \omega, d_r) \geq -s \end{aligned} \tag{16}$$

where the slack variable s ensures that there always exists a feasible solution, and λ (set to 10^3) is the relative weight of s in the objective. $\mathcal{D}_r^e$ is the set of disturbances respecting the disturbance bound $\bar{d}_r^e$. Intuitively, (16) minimally adjusts the nominal policy so that the resultant twist commands ensure system safety in the current environment. Furthermore, the introduction of the slack variable ensures that the QP problem in (16) is always feasible, especially when the disturbance bounds are overly conservative. Note that the disturbance optimization in the constraint in (16) is independent of the optimization of ω, since the disturbance enters additively into f_r in Equation (10). Since f_r is affine in control and disturbance, (16) is a quadratic program (QP) that we can solve efficiently online, making the proposed framework amenable for real-world robotic systems.

We remark that the design of the filter in (15) is grounded in reachability theory. If V_ψ perfectly models the underlying value function and the system starts outside of the BRT, then (15) is guaranteed to maintain safety [68].

Remark. *When the quadruped is controlled by an end-to-end nominal controller, we can use the OCR-VN framework to maintain safety with a minor adjustment. While $V_\psi > 0$, we permit the nominal controller to execute unhindered. Otherwise, we must provide our own high-level planner $\pi_{nom}^{high}(x)$ for solving* (16) *and a low-level policy π_{nom}^{low} for execution.*

VI. Simulation Experiments

We test the proposed framework on a Unitree Go1 quadruped in Isaac Sim (this section) and on a hardware testbed (Section VII) in a variety of unseen obstacle settings and terrains, as well as on different nominal controllers. The goal of our simulation studies and experiments is to answer the following questions about the proposed framework:

1) Can the safety filter dynamically adapt to unknown obstacles to maintain safety?
2) Can the safety filter adapt to system and environment uncertainty (e.g., uneven terrains, slippery surfaces, etc.)?
3) Can the safety filter maintain safety under different nominal policies?

A. Nominal Controllers

We test our framework on several different hierarchical nominal policies. A nominal policy is generated by selecting one of the three high-level planners and one of the three low-level locomotion policies described below. These policies are selected to contain both RL-based policies as well as MPC-based policies. We test all different combinations of high-level and low-level policies, but present the results for only a few of them here for brevity purposes.

Additionally, we include an end-to-end nominal policy from ABS [29] (called ABS-Agile here on), as an illustration of how our method can generalize to end-to-end nominal policies. To implement our method on top of the end-to-end ABS-Agile policy, we compute the safety twist commands using the proposed filter framework in Equation (15) with the Predictive Sampling-based (PS) high-level planner (see VI-A1a) and use the ABS-Recovery policy (see VI-A2c) to track the filtered twist commands. Whenever the system safety is not at risk, we use the nominal ABS-Agile policy.

1) High-Level Planners:

a) Predictive Sampling-Based Planner (PS): PS is a sampling-based high-level MPC planner that optimizes the robot trajectory to reach the goal while avoiding obstacles. The cost function for a discrete control sequence $\boldsymbol{\omega} = \{\omega^j\}_{j=1}^{\lfloor T/\eta \rfloor}$ evaluated at the reduced state x_r is given by:

$$J_{\boldsymbol{\omega}}(x_r) = \sum_{j=1}^{\lfloor T/\eta \rfloor} \left(||\hat{x}_r^j - g|| + c \cdot \mathbb{1}\{\hat{x}_r^j \in \hat{\mathcal{L}}^e\} \right)$$

where T is the prediction horizon and η is the discrete time step. $\hat{x}_r^j$ is the future state at time $j \cdot \eta$ predicted by the reduced-order dynamics f_r in Equation (10) after applying $\boldsymbol{\omega}$ to the system. g is the goal state, c is a collision penalty, and $\hat{\mathcal{L}}^e$ is the estimated obstacle map computed online using obtained LiDAR scans via `tinySLAM` [81]. Intuitively, the cost function is minimized by control sequences that bring the system to the goal fastest without collisions. To compute the optimal high-level commands, PS samples N control sequences $\boldsymbol{\omega}_i, \forall i = 1, ..., N$ from a Gaussian centered at $\boldsymbol{\omega}_{\text{seed}}$ with standard deviation σ. The cost $J_{\boldsymbol{\omega}_i}(x_r)$ for each sequence is computed and finally the best control sequence $\boldsymbol{\omega}_{\text{best}}$ is selected which minimizes the cost. PS executes the first twist command in $\boldsymbol{\omega}_{\text{best}}$ and sets $\boldsymbol{\omega}_{\text{seed}} = \boldsymbol{\omega}_{\text{best}}$ for the next control iteration. We set $N = 1,000$, $\sigma = 0.5$, $T = 4$ s, $\eta = 0.2$ s, $c = 10^9$, and the initial $\boldsymbol{\omega}_{\text{seed}}$ to the control range center.

b) Naive Planner (NVE): NVE performs basic goal-seeking without obstacle avoidance. Let $p_{\theta_{\text{goal}}}$ be the angle to the goal relative to the robot's heading. NVE computes:

$$v = \begin{cases} v_{\max}, & |p_{\theta_{\text{goal}}}| \leq \frac{\pi}{2} \\ v_{\min}, & |p_{\theta_{\text{goal}}}| > \frac{\pi}{2}, \end{cases}$$

$$w = \begin{cases} w_{\min} \cdot \min\{|\frac{p_{\theta_{\text{goal}}}}{p_{\theta_{\max}}}|, 1\}, & p_{\theta_{\text{goal}}} \leq 0 \\ w_{\max} \cdot \min\{|\frac{p_{\theta_{\text{goal}}}}{p_{\theta_{\max}}}|, 1\}, & p_{\theta_{\text{goal}}} > 0, \end{cases}$$

where $p_{\theta_{\max}}$ is the heading difference beyond which yaw rate is maximized. We set $p_{\theta_{\max}} = \frac{\pi}{4}$ rad. Intuitively, NVE turns greedily towards the goal and commands maximum/minimum forward velocity when the goal is ahead/behind.

c) Human Teleoperation (HMN): a human teleoperator provides the high-level twist commands.

2) Low-Level Policies:

a) Walk-These-Ways (WTW): an RL-based policy that encodes a structured family of locomotion strategies to enable diverse task generalization such as crouching, hopping, running, and others [21]. During training, the policy is rewarded for accurate twist tracking and gait following. During deployment, different locomotion strategies can be used by changing gait parameters. We use the gait parameters corresponding to a 3 Hz trot for its stability and agility.

b) Legged Model Predictive Control (MPC) by Unitree: a model-based policy provided by Unitree Robotics as a ROS package for the Go1 quadruped robot [74]. The policy tracks twist commands using an internal MPC-based algorithm that computes motor torque controls. It is widely used by consumers of the Go1 quadruped hardware.

c) ABS Recovery Policy: an RL-based policy used in the ABS framework to track twist commands as fast as possible in order to serve as a backup shielding policy [29]. During training, the policy is rewarded for accurate twist tracking and maintaining a stable posture. Similar to its role in the ABS framework, we will use the ABS-Recovery policy as the low-level backup shielding policy for the ABS-Agile policy.

B. Baseline and Ablations

We evaluate our framework against a range of nominal controllers and the end-to-end safety framework, ABS, proposed in [29]. ABS integrates an RL-based agile policy with a safety-oriented recovery policy. A value function-based predictor determines when to switch to the recovery policy, which then generates safe twist commands to be executed by the recovery policy. We refer the interested readers to [29] for more details.

We also evaluate the OCR framework without the disturbance estimation module (OCR $\setminus$ DE) and calibration step (OCR $\setminus$ C), as well as if the safety filter is not used at all (No Filter) to understand the importance of each of these modules.

C. Metrics

Each experiment trial ends in either a success, a collision, or a timeout after 60 s. Across successful runs, we report the average velocity $\bar{v}$ of the quadruped along the trajectory, the average rate of safety filter activation $\bar{r}$, and the average minimum distance to the obstacle set $\bar{q}$.

D. Simulation Setups

We test our framework as well as baselines on randomly generated environments. Each environment contains 4 circular obstacles, where each obstacle is spawned at some location drawn uniformly randomly from $[-2, 2]$ m $\times$ $[-2, 2]$ m. Each obstacle has a radius drawn uniformly randomly from $[0.1, 1]$ m. Each environment also draws an additional payload uniformly randomly from $[-1, -0.5] \cup [0.5, 1]$ kg and a ground friction coefficient $[0.5, 0.75] \cup [1.25, 1.5]$ for dynamical variation. We remark that these payloads and frictions are contained within the range used to train the nominal policies and baselines so that the comparison is fair, but they represent the outer limits to effectively test robustness. The quadruped must navigate from $(-5, 0)$ m to $(5, 0)$ m without colliding (see Figure 4).

Additionally, we present hand-designed environments that stress-test and highlight the abilities of our framework.

E. Simulation Results

Recall that the goal of our simulation studies and experiments is to answer the following questions: can the safety filter adapt to unknown obstacles in the environment, uncertainty in the system and environment, and different nominal policies?

1) Safeguarding Different Nominal Controllers: We first evaluate the efficacy of the OCR framework and its ablations for safeguarding various nominal controllers in different obstacle settings. Numerical results are listed in Table III. The OCR framework achieves high success rates, low collision rates, and comfortable distances to the obstacle set across all nominal controllers, highlighting its ability to automatically safeguard different nominal controllers in different environments without *a priori* knowledge. Its success rates are consistently high regardless of whether the underlying nominal control has built-in obstacle avoidance (ABS-Agile and PS + WTW) or not (NVE + WTW). Figure 4 illustrates the robot trajectories under the OCR framework and the framework's ability to ensure safety when the nominal controller would cause collision.

The results in Table III also clearly show the importance of the disturbance estimation module and the calibration step for enabling the OCR framework to be robust to modeling and learning errors, respectively. Interestingly, even though the ABS framework is particularly designed to safeguard the ABS-Agile policy, our method also outperforms the ABS framework on ABS-Agile. A key reason for this is the robustness of the proposed framework not only to environment uncertainty (i.e.,

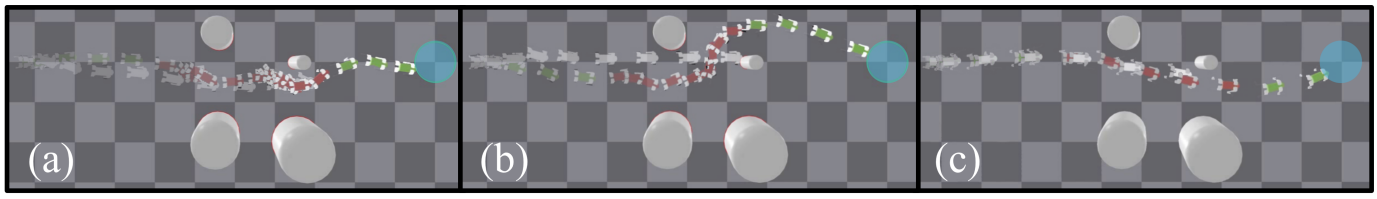

Fig. 4: The OCR framework (green/red : nominal/filtered) safeguards different nominal controllers navigating to a goal (cyan) in an environment with a payload of -0.6 kg and a friction of 0.7. By themselves (white), the (a) PS + WTW, (b) NVE + WTW, and (c) ABS-Agile controllers fail to maintain safety due to dynamical uncertainty caused by low payload and friction.

TABLE III: Simulation Results for OCR, Ablations, and ABS across Nominal Controllers (100 Trials)

Controller	Filter	Success Rate ↑	Collision Rate ↓	Timeout Rate ↓	$\bar{v}$ (m/s) ↑	$\bar{r}$ ↓	$\bar{q}$ (m) ↑
ABS-Agile	No Filter	0.75	0.25	0.00	**2.10**	**0.00**	0.41
	ABS	0.80	0.20	0.00	2.03	0.03	0.41
	OCR \ DE	0.40	0.35	0.25	0.98	0.62	0.41
	OCR \ C	0.81	0.19	0.00	1.70	0.28	0.41
	OCR (ours)	**0.91**	**0.09**	0.00	1.22	0.59	**0.58**
PS + WTW	No Filter	0.72	0.28	0.00	**1.36**	**0.00**	0.43
	OCR \ DE	0.78	0.22	0.00	0.84	0.61	0.42
	OCR \ C	0.90	0.09	0.01	1.21	0.18	0.44
	OCR (ours)	**1.00**	**0.00**	0.00	0.97	0.42	**0.66**
NVE + WTW	No Filter	0.20	0.80	0.00	**1.70**	**0.00**	0.31
	OCR \ DE	0.21	0.79	0.00	0.93	0.61	0.37
	OCR \ C	0.45	0.55	0.00	1.32	0.28	0.39
	OCR (ours)	**0.91**	**0.08**	0.01	1.04	0.47	**0.62**

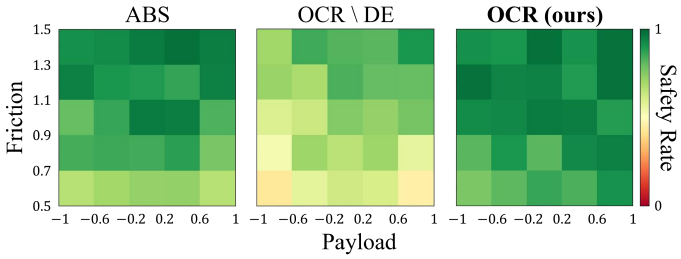

Fig. 5: (In color) Safety rates across settings (1,000 trials).

unknown obstacles) but also to dynamics uncertainty, as we discuss next.

2) Robustness to Dynamical Uncertainty: Results in Table III demonstrate the ability of the OCR framework to handle variation in the system dynamics. Recall that each environment has its own payload and friction parameters, which significantly influence the dynamics of the system. Additionally, different low-level policies have different abilities to track velocity and yaw rate commands. Despite these challenges, the OCR framework consistently achieves a high success rate without a priori access to the underlying nominal policy or simulation environments.

To study robustness more thoroughly, we plot safety rates across environment settings for different frameworks using the ABS-Agile policy as the nominal policy in Figure 5. The OCR framework achieves high safety rates across all settings, surpassing the ABS baseline particularly in low-friction environments, but only when using disturbance estimation. This leads us to conclude that disturbance estimation plays a key role in the robustness of the proposed framework. For all numerical results, see Appendix B.

Figure 6 (a) visualizes the behavior of the OCR and ABS frameworks in an environment with high perturbations in dynamics (a payload of -0.9 kg and a friction of 0.5). The OCR framework estimates large disturbances online and accordingly intervenes to steer the system away from collision. The adaptation facilitated by the disturbance estimation module is visualized in Figure 6 (b) in a hand-designed environment with a sudden friction change in the blue region. Correspondingly, the proposed framework automatically estimates larger disturbance bounds in this region of low friction (the purple point), which subsequently, leads to a more aggressive safety filter to ensure system safety. The ABS framework, in contrast, intervenes later and allows the quadruped to approach dangerously close to the wall obstacle, eventually leading to collision.

3) Further Comparisons with ABS: We now further compare the proposed OCR framework with ABS, as it leverages a similar safety filtering framework. One of the key advantages of the OCR framework is that it grants the user the freedom

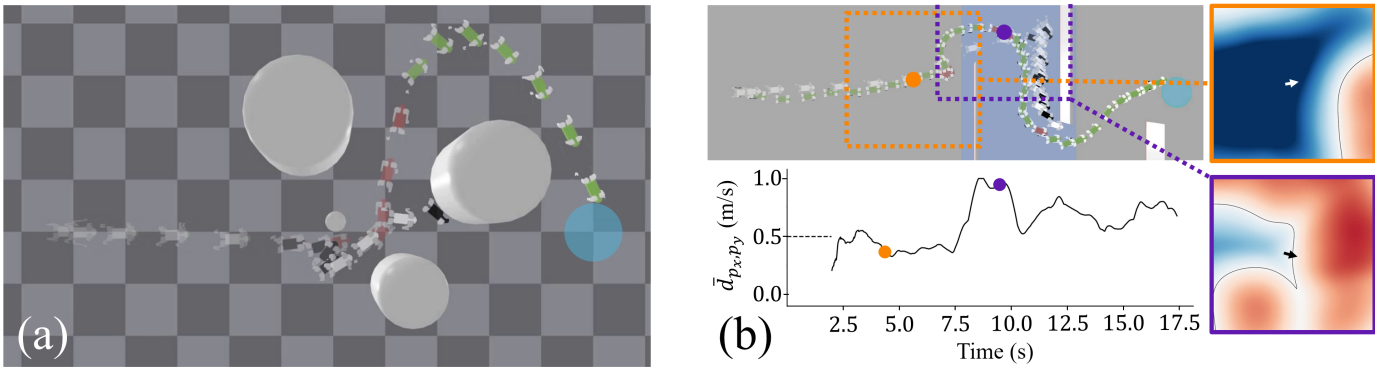

Fig. 6: OCR (green/red : nominal/filtered) and ABS (white/black : nominal/filtered) frameworks in (a) a validation environment with a payload of -0.9 kg and a friction of 0.5 and (b) a hand-designed obstacle configuration with a region of low friction (blue). In (b), we plot the evolution of the estimated disturbance bound in position $\bar{d}^e_{p_x,p_y}$, as well as the OCR-VN predictions at two different states (orange, purple).

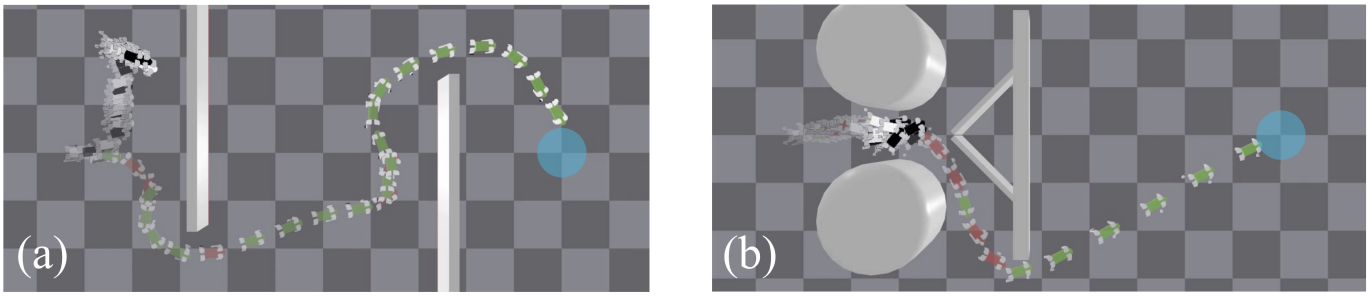

Fig. 7: OCR (green/red : nominal/filtered) and ABS (white/black : nominal/filtered) frameworks in environments with hand-designed obstacle configurations. In (b), the OCR framework is used without calibration due to the highly cluttered setting.

to choose the most appropriate nominal controller for the task at hand without needing to recompute safety assurances, whereas the ABS framework is specific to a particular nominal policy. We demonstrate this utility in Figure 7 (a), where the robot must navigate two sharp turns. The OCR framework successfully completes the task using the PS + WTW nominal controller. On the other, the ABS baseline gets stuck, because the learning-based ABS-Agile policy is not suited for the required navigation style.

The policy-independence of the OCR framework is a consequence of the *optimality* of the underlying value function synthesized during the training phase described in Section V-A. We demonstrate the advantages of using the optimal value function in Figure 7 (b) in a hand-designed "dead-end" environment. Since the setting is highly cluttered, we use OCR \ C to reduce conservatism. We remark that this remains a fair evaluation with respect to the ABS baseline because OCR \ C retains a comparable safety rate to ABS in Table III.

Using a control search guided by a policy-conditioned value function, the ABS baseline suffers from unnecessary interventions and suboptimality related to the quality of the ABS-Agile policy. On the other hand, OCR framework reasons about the optimal safety behavior (regardless of the nominal policy), reducing unnecessary interventions and permitting task progress.

VII. Hardware Experiments

We deploy the OCR framework with nominal controllers from Section VI-A on a Unitree Go1 quadruped robot equipped with an onboard Slamtec RPLiDAR A2 sensor. The quadruped maintains an estimate of its state in the global frame via `tinySLAM` [81], a LiDAR-based simultaneous localization and mapping (SLAM) algorithm, which is subsequently used for disturbance bound estimation.

A. Hardware Setups

We first quantitatively and qualitatively compare the proposed framework and all baselines in two different experiment settings. In the first setting, the robot needs to navigate through an obstacle maze consisting of walls and circular obstacles. The second setting is the same as the first, except that a rectangular region directly preceding the first obstacle has a very low friction due to an oil-soaked tarp, to test the dynamic safety adaptivity of different methods. The setups are shown in Figure 8. In our experiments, we also include results on a few additional nominal policies to stress-test our system.

Additionally, we qualitatively test our framework in a diverse set of real-world experiments, including cluttered environments, rough terrains, external disturbances, and adversarial human teleoperation.

B. Hardware Results

The hardware results listed in Table IV reaffirm the findings in simulation. Namely, the OCR framework is effective at safeguarding a diverse set of controllers. The framework also displays a significant robustness to changes in the system dynamics, as evidenced by the high success rates even in the slippery condition, where the ABS baseline fails. Figure 8 illustrates the difference between the OCR and ABS frameworks in the slippery condition. Whereas the ABS baseline intervenes too late to maintain safety, the OCR framework slows down the robot and turns in time to avoid collision.

We further demonstrate the capabilities of the framework in a variety of real-world scenarios. Videos of these demonstrations can be found on the project website listed above Figure 1. Many of these scenarios are illustrated in Figure 1.

1) Safeguarding Different Nominal Controllers: Figure 1 illustrates the ability of the OCR framework to automatically safeguard a variety of high-level planners, including (a) learning-based, (c, f, k) model-based, (b, d, g, h, i, j) human teleoperated, and (e) blind planners, on top of different low-level locomotion policies, including (a, f, i, j, k) learning-based and (b, c, d, e, g, h) model-based policies.

2) External Disturbances: Subplot (f) in Figure 1 demonstrates the framework's ability to preserve safety even under forceful disturbances. After receiving a kick while on a slippery floor, the framework guides the robot to turn sharply from the obstacle and move away to maintain safety.

3) Adversarial Human Teleoperation: In subplot (h) of Figure 1, an adversarial human teleoperator attempts to collide the robot into a pillar multiple times. The filter activates only when necessary and causes the robot to veer to avoid collision while respecting the commanded input as much as possible.

4) Dynamic Obstacles: The framework's ability to react to dynamic obstacles in real time is illustrated in subplots (e, k) in Figure 1. The robot slows down from a velocity of roughly

TABLE IV: Hardware Results for OCR and ABS across Nominal Controllers (10 Trials)

Controller	Filter	Normal Condition # Successes ↑	Normal Condition # Collisions ↓	Normal Condition # Timeouts ↓	Slippery Condition # Successes ↑	Slippery Condition # Collisions ↓	Slippery Condition # Timeouts ↓
ABS-Agile	No Filter	9	1	0	1	9	0
	ABS	9	1	0	2	8	0
	OCR (ours)	**10**	**0**	0	**8**	**1**	1
PS + WTW	No Filter	5	5	0	1	9	0
	OCR (ours)	**9**	**1**	0	**8**	**2**	0
NVE + WTW	No Filter	0	10	0	0	10	0
	OCR (ours)	**9**	**0**	1	**8**	**2**	0
PS + MPC	No Filter	5	5	0	0	10	0
	OCR (ours)	**10**	**0**	0	**9**	**1**	0
NVE + MPC	No Filter	0	10	0	0	10	0
	OCR (ours)	**9**	**0**	1	**7**	**3**	0

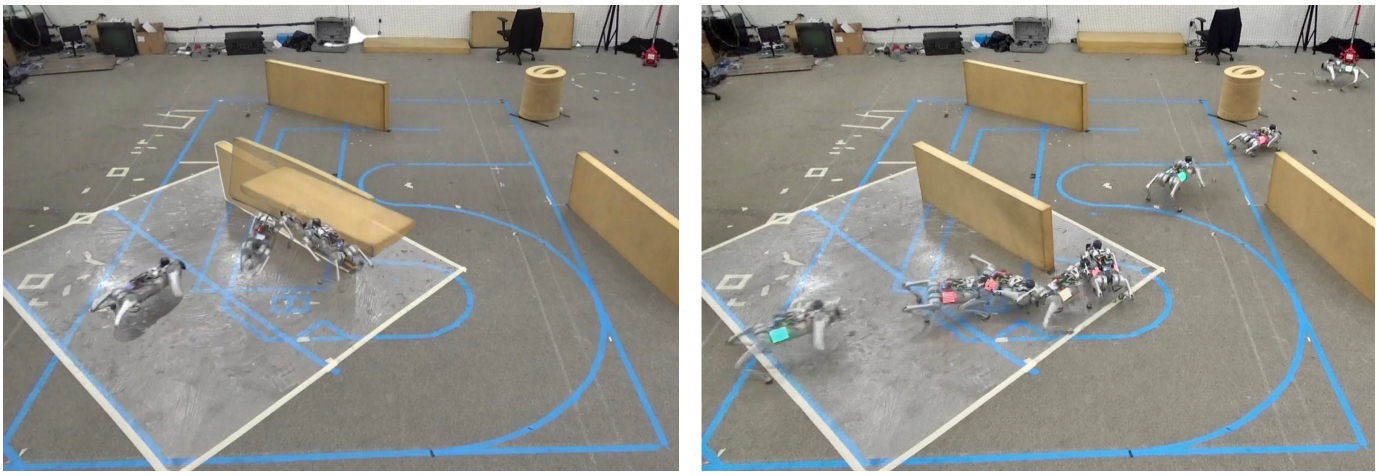

Fig. 8: Hardware experiments with a slippery region outlined in white. (Left) The ABS baseline collides due to drifting caused by the slippery floor. (Right) The OCR framework (green/red : nominal/filtered) stops and turns in time to prevent a collision, demonstrating superior robustness to changes in the system dynamics.

2 m/s to provide enough turn radius to avoid the box placed suddenly by a human in front of the robot in subplot (e).

5) Cluttered Indoor Environments: The robot navigates narrow corridors in subplots (a, b, c) in Figure 1. Due to the conservatism of the OCR framework, a few adjustments are needed for good performance in highly cluttered settings. First, we discard LiDAR readings outside of a front-facing cone spanning $\pi/2$ rad. We believe this is necessary because of a shift in distribution of LiDAR readings from what is seen during training, which has sparse obstacles and no walls. Second, we use the uncalibrated output of the OCR-VN. We theorize that the conservatism of the OCR framework is an inherent result of using a worst-case analysis for disturbance, which makes the safety problem especially difficult in crowded settings. We defer addressing these limitations to future work.

6) Rough Terrain: The robot avoids collisions on rough outdoor terrains in subplots (d, i, j) of Figure 1, further highlighting the ability of the proposed framework to adapt to dynamics uncertainties.

VIII. LIMITATIONS

Although our proposed framework achieves high safety rates in both simulation and hardware results in Tables III and IV, the framework still exhibits a nonzero collision rate. We attribute this to several failure modes of the framework that we identify and discuss next.

First, during deployment, the system may experience disturbances that exceed the estimated disturbance bounds which the framework is designed to safeguard against. Furthermore, there is an unavoidable delay before environment changes will reflect in the estimated disturbance bounds. During this delay period, the system can encounter failure before the framework has a chance to adapt.

Second, the OCR-VN can contain learning errors that are critical to safety. While the calibration scheme presented in Section V-A4 can help us better understand the degree of these errors, we cannot directly extrapolate the theoretical guarantees from a validation dataset to the real world, due to potential distribution shifts. It would be interesting to explore online calibration methods to overcome this challenge.

In addition to these failure modes which affect safety, there are several limitations of the framework that affect its performance. For some low-level locomotion policies and environment settings, the error in the reduced-order dynamics model can be very large, leading to high dynamical uncertainty. Large estimated disturbances can also appear as an artifact of latency

and state estimation issues with real-world hardware. The resulting disturbance bounds can produce overly conservative BRTs which severely inhibit progress by the system. In highly cluttered environments, this can manifest as stalling behavior.

The conservatism of the framework is also partly a consequence of our choice to model the dynamics uncertainty as *adversarial* in nature. Future works can explore approaches to overcome these issues by using proactive methods of modeling and estimating dynamics and disturbances, for example by using observations of the environment or learning controller-specific dynamics to anticipate future system behavior and reduce uncertainty, as well as considering different characterizations of disturbances.

IX. Conclusion

We propose the OCR framework, which uses an adaptive safety filter to robustly ensure the safety of a quadruped robot running an *a priori unknown* locomotion policy in *a priori unknown* environments using only LiDAR observations. The offline training of the OCR-VN is done *without a priori access to the controllers or a simulator* - hence, "One Filter to Deploy Them All". In simulation and hardware experiments on a Unitree Go1 quadruped, we demonstrate the superior efficacy of the proposed approach to ensure, in zero-shot fashion, the safety of numerous controllers across diverse environment configurations and perturbed dynamics. Videos demonstrating the efficacy of the proposed approach across various settings can be found on the project website listed above Figure 1.

Acknowledgments

This work is supported in part by a NASA Space Technology Graduate Research Opportunity, the NSF CAREER Program under award 2240163, and the DARPA ANSR program.

References

[1] S. Halder, K. Afsari, E. Chiou, R. Patrick, and K. A. Hamed, "Construction inspection and monitoring with quadruped robots in future human-robot teaming: A preliminary study," *Journal of Building Engineering*, vol. 65, p. 105814, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2352710222018204

[2] C. Gehring, P. Fankhauser, L. Isler, R. Diethelm, S. Bachmann, M. Potz, L. Gerstenberg, and M. Hutter, "Anymal in the field: Solving industrial inspection of an offshore hvdc platform with a quadrupedal robot," in *Field and Service Robotics*, G. Ishigami and K. Yoshida, Eds. Singapore: Springer Singapore, 2021, pp. 247–260.

[3] N. Li, J. Cao, and Y. Huang, "Fabrication and testing of the rescue quadruped robot for post-disaster search and rescue operations," in *2023 IEEE 3rd International Conference on Electronic Technology, Communication and Information (ICETCI)*. IEEE, 2023, pp. 723–729.

[4] C. Cruz Ulloa, J. del Cerro, and A. Barrientos, "Mixed-reality for quadruped-robotic guidance in sar tasks," *Journal of Computational Design and Engineering*, vol. 10, no. 4, pp. 1479–1489, 2023.

[5] F. Gao, C. Lei, X. Long, J. Wang, and P. Song, "Design and development of an intelligent pet-type quadruped robot," in *2021 IEEE 4th International Conference on Multimedia Information Processing and Retrieval (MIPR)*. IEEE, 2021, pp. 366–371.

[6] H. Lee, "A human-robot interaction entertainment pet robot," *Journal of the Korean Institute of Intelligent Systems*, vol. 24, no. 2, pp. 179–185, 2014.

[7] Z. Chen, T. Fan, X. Zhao, J. Liang, C. Shen, H. Chen, D. Manocha, J. Pan, and W. Zhang, "Autonomous social distancing in urban environments using a quadruped robot," *IEEE Access*, vol. 9, pp. 8392–8403, 2021.

[8] A. D. Ames, P. Tabuada, A. Jones, W.-L. Ma, M. Rungger, B. Schürmann, S. Kolathaya, and J. W. Grizzle, "First steps toward formal controller synthesis for bipedal robots with experimental implementation," *Nonlinear Analysis: Hybrid Systems*, vol. 25, pp. 155–173, 2017. [Online]. Available: http://ames.caltech.edu/ames2017first.pdf

[9] R. Grandia, A. J. Taylor, A. D. Ames, and M. Hutter, "Multi-layered safety for legged robots via control barrier functions and model predictive control," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 8352–8358.

[10] H. U. Unlu, V. M. Gonçalves, D. Chaikalis, A. Tzes, and F. Khorrami, "A control barrier function-based motion planning scheme for a quadruped robot," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 2024, pp. 12 172–12 178.

[11] M. Tayal and S. Kolathaya, "Safe legged locomotion using collision cone control barrier functions (c3bfs)," *arXiv preprint arXiv:2309.01898*, 2023.

[12] J. Kim, J. Lee, and A. D. Ames, "Safety-critical coordination for cooperative legged locomotion via control barrier functions," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2023, pp. 2368–2375.

[13] J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter, "Learning quadrupedal locomotion over challenging terrain," *Science Robotics*, vol. 5, no. 47, p. eabc5986, 2020. [Online]. Available: https://www.science.org/doi/abs/10.1126/scirobotics.abc5986

[14] D. Hoeller, N. Rudin, D. Sako, and M. Hutter, "Anymal parkour: Learning agile navigation for quadrupedal robots," *Science Robotics*, vol. 9, no. 88, p. eadi7566, 2024. [Online]. Available: https://www.science.org/doi/abs/10.1126/scirobotics.adi7566

[15] G. Bellegarda and A. Ijspeert, "Visual cpg-rl: Learning central pattern generators for visually-guided quadruped navigation," *arXiv preprint arXiv:2212.14400*, 2022.

[16] J. Tan, T. Zhang, E. Coumans, A. Iscen, Y. Bai, D. Hafner, S. Bohez, and V. Vanhoucke, "Sim-to-real: Learning agile locomotion for quadruped robots," *arXiv preprint arXiv:1804.10332*, 2018.

[17] Y.-S. Luo, J. H. Soeseno, T. P.-C. Chen, and W.-C. Chen, "Carl: controllable agent with reinforcement learning for quadruped locomotion," *ACM Trans. Graph.*, vol. 39, no. 4, Aug. 2020. [Online]. Available: https://doi.org/10.1145/3386569.3392433

[18] H. Shi, Q. Zhu, L. Han, W. Chi, T. Li, and M. Q.-H. Meng, "Terrain-aware quadrupedal locomotion via reinforcement learning," *arXiv preprint arXiv:2310.04675*, 2023.

[19] L. Schneider, J. Frey, T. Miki, and M. Hutter, "Learning risk-aware quadrupedal locomotion using distributional reinforcement learning," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 2024, pp. 11 451–11 458.

[20] A. Kumar, Z. Fu, D. Pathak, and J. Malik, "Rma: Rapid motor adaptation for legged robots," 2021.

[21] G. B. Margolis and P. Agrawal, "Walk these ways: Tuning robot control for generalization with multiplicity of behavior," *Conference on Robot Learning*, 2022.

[22] S. Gangapurwala, M. Geisert, R. Orsolino, M. Fallon, and I. Havoutis, "Real-time trajectory adaptation for quadrupedal locomotion using deep reinforcement learning," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 5973–5979.

[23] W. Xiao, T. He, J. Dolan, and G. Shi, "Safe deep policy adaptation," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 2024, pp. 17 286–17 292.

[24] R. Cheng, G. Orosz, R. M. Murray, and J. W. Burdick, "End-to-end safe reinforcement learning through barrier functions for safety-critical continuous control tasks," in *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence and Thirty-First Innovative Applications of Artificial Intelligence Conference and Ninth AAAI Symposium on Educational Advances in Artificial Intelligence*, ser. AAAI'19/IAAI'19/EAAI'19. AAAI Press, 2019. [Online]. Available: https://doi.org/10.1609/aaai.v33i01.33013387

[25] W. Zhao, T. He, and C. Liu, "Model-free safe control for zero-violation reinforcement learning," in *Proceedings of the 5th Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, A. Faust, D. Hsu, and G. Neumann, Eds., vol. 164. PMLR, 08–11 Nov 2022, pp. 784–793. [Online]. Available: https://proceedings.mlr.press/v164/zhao22a.html

[26] Y. K. Nakka, A. Liu, G. Shi, A. Anandkumar, Y. Yue, and S.-J. Chung, "Chance-constrained trajectory optimization for safe exploration and learning of nonlinear systems," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 389–396, 2021.

[27] 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.

[28] K.-C. Hsu, A. Z. Ren, D. P. Nguyen, A. Majumdar, and J. F. Fisac, "Sim-to-lab-to-real: Safe reinforcement learning with shielding and generalization guarantees," *Artificial Intelligence*, vol. 314, p. 103811, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0004370222001515

[29] T. He, C. Zhang, W. Xiao, G. He, C. Liu, and G. Shi, "Agile but safe: Learning collision-free high-speed legged locomotion," in *Robotics: Science and Systems (RSS)*, 2024.

[30] H. Bharadhwaj, A. Kumar, N. Rhinehart, S. Levine, F. Shkurti, and A. Garg, "Conservative safety critics for exploration," in *International Conference on Learning Representations*, 2021. [Online]. Available: https://openreview.net/forum?id=iaO86DUuKi

[31] D. Kim, D. Carballo, J. Di Carlo, B. Katz, G. Bledt, B. Lim, and S. Kim, "Vision aided dynamic exploration of unstructured terrain with a small-scale quadruped robot," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 2464–2470.

[32] T. Dudzik, M. Chignoli, G. Bledt, B. Lim, A. Miller, D. Kim, and S. Kim, "Robust autonomous navigation of a small-scale quadruped robot in real-world environments," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 3664–3671.

[33] R. Buchanan, L. Wellhausen, M. Bjelonic, T. Bandyopadhyay, N. Kottege, and M. Hutter, "Perceptive whole-body planning for multilegged robots in confined spaces," *Journal of Field Robotics*, vol. 38, no. 1, pp. 68–84, 2021. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.21974

[34] M. Gaertner, M. Bjelonic, F. Farshidian, and M. Hutter, "Collision-free mpc for legged robots in static and dynamic scenes," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 8266–8272.

[35] J.-R. Chiu, J.-P. Sleiman, M. Mittal, F. Farshidian, and M. Hutter, "A collision-free mpc for whole-body dynamic locomotion and manipulation," in *2022 International Conference on Robotics and Automation (ICRA)*, 2022, pp. 4686–4693.

[36] M. Mattamala, N. Chebrolu, and M. Fallon, "An efficient locally reactive controller for safe navigation in visual teach and repeat missions," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 2353–2360, 2022.

[37] Q. Liao, Z. Li, A. Thirugnanam, J. Zeng, and K. Sreenath, "Walking in narrow spaces: Safety-critical locomotion control for quadrupedal robots with duality-based optimization," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2023, pp. 2723–2730.

[38] L. Wellhausen and M. Hutter, "Artplanner: Robust legged robot navigation in the field," *Field Robotics*, vol. 3, pp. 413–434, 2023.

[39] S. Teng, Y. Gong, J. W. Grizzle, and M. Ghaffari, "Toward safety-aware informative motion planning for legged robots," 2021. [Online]. Available: https://arxiv.org/abs/2103.14252

[40] F. Jenelten, J. Hwangbo, F. Tresoldi, C. D. Bellicoso, and M. Hutter, "Dynamic locomotion on slippery ground," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 4170–4176, 2019.

[41] V. Makoviychuk, L. Wawrzyniak, Y. Guo, M. Lu, K. Storey, M. Macklin, D. Hoeller, N. Rudin, A. Allshire, A. Handa, and G. State, "Isaac gym: High performance gpu-based physics simulation for robot learning," 2021.

[42] N. Rudin, D. Hoeller, P. Reist, and M. Hutter, "Learning to walk in minutes using massively parallel deep reinforcement learning," in *5th Annual Conference on Robot Learning*, 2021. [Online]. Available: https://openreview.net/forum?id=wK2fDDJ5VcF

[43] J. Kober, J. Bagnell, and J. Peters, "Reinforcement learning in robotics: A survey," *The International Journal of Robotics Research*, vol. 32, pp. 1238–1274, 09 2013.

[44] G. B. Margolis, G. Yang, K. Paigwar, T. Chen, and P. Agrawal, "Rapid locomotion via reinforcement learning," *The International Journal of Robotics Research*, vol. 43, no. 4, pp. 572–587, 2024. [Online]. Available: https://doi.org/10.1177/02783649231224053

[45] J. Hwangbo, J. Lee, A. Dosovitskiy, D. Bellicoso, V. Tsounis, V. Koltun, and M. Hutter, "Learning agile and dynamic motor skills for legged robots," *Science Robotics*, vol. 4, no. 26, p. eaau5872, 2019. [Online]. Available: https://www.science.org/doi/abs/10.1126/scirobotics.aau5872

[46] S. Gangapurwala, M. Geisert, R. Orsolino, M. Fallon, and I. Havoutis, "Rloc: Terrain-aware legged locomotion using reinforcement learning and optimal control," *IEEE Transactions on Robotics*, vol. 38, no. 5, pp. 2908–2927, 2022.

[47] C. Yang, K. Yuan, Q. Zhu, W. Yu, and Z. Li, "Multi-expert learning of adaptive legged locomotion," *Science Robotics*, vol. 5, no. 49, p. eabb2174, 2020. [Online]. Available: https://www.science.org/doi/abs/10.1126/scirobotics.abb2174

[48] T. Miki, J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter, "Learning robust perceptive locomotion for quadrupedal robots in the wild," *Science Robotics*, vol. 7, no. 62, p. eabk2822, 2022. [Online]. Available: https://www.science.org/doi/abs/10.1126/scirobotics.abk2822

[49] Z. Zhuang, Z. Fu, J. Wang, C. Atkeson, S. Schwertfeger, C. Finn, and H. Zhao, "Robot parkour learning," in *Conference on Robot Learning (CoRL)*, 2023.

[50] C. Zhang, N. Rudin, D. Hoeller, and M. Hutter, "Learning agile locomotion on risky terrains," *arXiv preprint arXiv:2311.10484*, 2023.

[51] F. Jenelten, J. He, F. Farshidian, and M. Hutter, "Dtc: Deep tracking control," *Science Robotics*, vol. 9, no. 86, p. eadh5401, 2024. [Online]. Available: https://www.science.org/doi/abs/10.1126/scirobotics.adh5401

[52] X. Cheng, K. Shi, A. Agarwal, and D. Pathak, "Extreme parkour with legged robots," *arXiv preprint arXiv:2309.14341*, 2023.

[53] T. Miki, J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter, "Learning robust perceptive locomotion for quadrupedal robots in the wild," *Science Robotics*, vol. 7, no. 62, Jan. 2022. [Online]. Available: http://dx.doi.org/10.1126/scirobotics.abk2822

[54] Y. Wang, Z. Jiang, and J. Chen, "Learning robust, agile, natural legged locomotion skills in the wild," in *RoboLetics: Workshop on Robot Learning in Athletics @CoRL 2023*, 2023. [Online]. Available: https://openreview.net/forum?id=b5hiuuX1sm

[55] A. Agarwal, A. Kumar, J. Malik, and D. Pathak, "Legged locomotion in challenging terrains using egocentric vision," in *Conference on robot learning*. PMLR, 2023, pp. 403–415.

[56] L. Wellhausen and M. Hutter, "Artplanner: Robust legged robot navigation in the field," *Field Robotics*, vol. 3, no. 1, p. 413–434, Jan. 2023. [Online]. Available: http://dx.doi.org/10.55417/fr.2023013

[57] C. Zhang, J. Jin, J. Frey, N. Rudin, M. Mattamala, C. Cadena, and M. Hutter, "Resilient legged local navigation: Learning to traverse with compromised perception end-to-end," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 2024, pp. 34–41.

[58] H. Liu and Q. Yuan, "Safe and robust motion planning for autonomous navigation of quadruped robots in cluttered environments," *IEEE Access*, vol. 12, pp. 69 728–69 737, 2024.

[59] D. Hoeller, L. Wellhausen, F. Farshidian, and M. Hutter, "Learning a state representation and navigation in cluttered and dynamic environments," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 5081–5088, 2021.

[60] R. Yang, M. Zhang, N. Hansen, H. Xu, and X. Wang, "Learning vision-guided quadrupedal locomotion end-to-end with cross-modal transformers," in *International Conference on Learning Representations*, 2022. [Online]. Available: https://openreview.net/forum?id=nhnJ3oo6AB

[61] N. Rudin, D. Hoeller, M. Bjelonic, and M. Hutter, "Advanced skills by learning locomotion and local navigation end-to-end," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022, pp. 2497–2503.

[62] C. Zhang, N. Rudin, D. Hoeller, and M. Hutter, "Learning agile locomotion on risky terrains," 2024. [Online]. Available: https://arxiv.org/abs/2311.10484

[63] M. Seo, R. Gupta, Y. Zhu, A. Skoutnev, L. Sentis, and Y. Zhu, "Learning to walk by steering: Perceptive quadrupedal locomotion in dynamic environments," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 5099–5105.

[64] S. Kareer, N. Yokoyama, D. Batra, S. Ha, and J. Truong, "Vinl: Visual navigation and locomotion over obstacles," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 2018–2024.

[65] J. Truong, M. Rudolph, N. H. Yokoyama, S. Chernova, D. Batra, and A. Rai, "Rethinking sim2real: Lower fidelity simulation leads to higher sim2real transfer in navigation," in *6th Annual Conference on Robot Learning*, 2022. [Online]. Available: https://openreview.net/forum?id=BxHcg_Zlpxj

[66] N. Yokoyama, A. Clegg, J. Truong, E. Undersander, T.-Y. Yang, S. Arnaud, S. Ha, D. Batra, and A. Rai, "Asc: Adaptive skill coordination for robotic mobile manipulation," *IEEE Robotics and Automation Letters*, vol. 9, no. 1, pp. 779–786, 2024.

[67] T.-Y. Yang, T. Zhang, L. Luu, S. Ha, J. Tan, and W. Yu, "Safe reinforcement learning for legged locomotion," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022, pp. 2454–2461.

[68] J. Borquez, K. Chakraborty, H. Wang, and S. Bansal, "On safety and liveness filtering using hamilton–jacobi reachability analysis," *IEEE Transactions on Robotics*, vol. 40, pp. 4235–4251, 2024.

[69] K. P. Wabersich, A. J. Taylor, J. J. Choi, K. Sreenath, C. J. Tomlin, A. D. Ames, and M. N. Zeilinger, "Data-driven safety filters: Hamilton-jacobi reachability, control barrier functions, and predictive methods for uncertain systems," *IEEE Control Systems Magazine*, vol. 43, no. 5, pp. 137–177, 2023.

[70] K.-C. Hsu, H. Hu, and J. F. Fisac, "The safety filter: A unified view of safety-critical control in autonomous systems," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 7, 2023.

[71] S. Bansal, M. Chen, S. Herbert, and C. J. Tomlin, "Hamilton-Jacobi Reachability: A brief overview and recent advances," in *IEEE Conference on Decision and Control (CDC)*, 2017.

[72] S. Bansal, V. Tolani, S. Gupta, J. Malik, and C. Tomlin, "Combining optimal control and learning for visual navigation in novel environments," in *Conference on Robot Learning (CoRL)*, 2019.

[73] J. Borquez, K. Nakamura, and S. Bansal, "Parameter-conditioned reachable sets for updating safety assurances online," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 10 553–10 559.

[74] Unitree Robotics. unitree_ros. [Online]. Available: https://github.com/unitreerobotics/unitree_ros

[75] J. Lygeros, "On reachability and minimum cost optimal control," *Automatica*, vol. 40, no. 6, pp. 917–927, 2004.

[76] I. Mitchell, A. Bayen, and C. J. Tomlin, "A time-dependent Hamilton-Jacobi formulation of reachable sets for continuous dynamic games," *IEEE Transactions on Automatic Control (TAC)*, vol. 50, no. 7, pp. 947–957, 2005.

[77] Edward Schmerling. hj_reachability: Hamilton-jacobi reachability analysis in jax. [Online]. Available: https://github.com/StanfordASL/hj_reachability

[78] S. Bansal and C. J. Tomlin, "DeepReach: A deep learning approach to high-dimensional reachability," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2021.

[79] A. N. Angelopoulos and S. Bates, "Conformal prediction: A gentle introduction," *Foundations and Trends® in Machine Learning*, vol. 16, no. 4, pp. 494–591, 2023. [Online]. Available: http://dx.doi.org/10.1561/2200000101

[80] A. Lin and S. Bansal, "Verification of neural reachable tubes via scenario optimization and conformal prediction," in *Proceedings of the 6th Annual Learning for Dynamics and Control Conference*, ser. Proceedings of Machine Learning Research, A. Abate, M. Cannon, K. Margellos, and A. Papachristodoulou, Eds., vol. 242. PMLR, 15–17 Jul 2024, pp. 719–731. [Online]. Available: https://proceedings.mlr.press/v242/lin24a.html

[81] B. Steux and O. E. Hamzaoui, "tinyslam: A slam algorithm in less than 200 lines c-language program," in *2010 11th International Conference on Control Automation Robotics and Vision*, 2010, pp. 1975–1979.

[82] "*NIST Digital Library of Mathematical Functions*," https://dlmf.nist.gov/, Release 1.1.11 of 2023-09-15, 2023, f. W. J. Olver, A. B. Olde Daalhuis, D. W. Lozier, B. I. Schneider, R. F. Boisvert, C. W. Clark, B. R. Miller, B. V. Saunders, H. S. Cohl, and M. A. McClain, eds. [Online]. Available: https://dlmf.nist.gov/

APPENDIX A
PROOF OF THEOREM 1

Proof. Theorem 1 is a straightforward application of the split conformal prediction method detailed in [79], where we set their "input x" as our network input $(x_r, \bar{d}_r^e, o^e)$, their "output y" as our true value $V(x_r; e)$, their "score function $s(x, y)$" as our prediction error $V_\psi(x_r, \bar{d}_r^e, o^e) - V(x_r; e)$, their "calibration size n" as our calibration size N, their "error rate α" in terms of our number of outliers k as $\frac{k+1}{N+1}$, and their "quantile $\hat{q}$" as our calibration level δ. Indeed, recall that we compute δ as the $\frac{N-k}{N} = \frac{N-(\alpha(N+1)-1)}{N} = \frac{(N+1)(1-\alpha)}{N}$ quantile of the calibration scores $\{s_i\}_{i=1}^N$, which is precisely how $\hat{q}$ is computed in [79]. Section 3.2 in [79] yields:

$$\mathop{\mathbb{P}}_{(x_r, \bar{d}_r^e, o^e)} \left(V_\psi(x_r, \bar{d}_r^e, o^e) - V(x_r; e) \leq \delta \right) \sim \mathcal{B}(N-k, k+1), \tag{17}$$

where $\mathcal{B}$ is the Beta distribution. Line (17) tells us that the probability that the OCR-VN overestimates the true value by at most δ follows a Beta distribution that depends on the number of calibration points N and outliers k used to determine δ. We are interested in lower bounding this probability by $1-\epsilon$ with a confidence of at least $1-\beta$. Fortunately, since the relationship on Line (17) is known to us, we can compute the k needed to satisfy a desired ϵ and β. The cumulative distribution function of the Beta distribution on Line (17) evaluated at x is given in terms of k by the incomplete beta function ratio $I_x(N-k, k+1) = \sum_{j=N-k}^{N} \binom{N}{j} x^j (1-x)^{N-j}$ from DLMF, (8.17.5) [82]. Changing the index $i = N - j$ yields $I_x(N-k, k+1) = \sum_{i=0}^{k} \binom{N}{N-i} x^{N-i} (1-x)^{N-(N-i)} = \sum_{i=0}^{k} \binom{N}{i} x^{N-i} (1-x)^i$. Thus, Line (17) is equivalent to the claim that for any violation parameter $\epsilon \in (0, 1)$ and confidence parameter $\beta \in (0, 1)$, $\mathop{\mathbb{P}}_{(x_r, \bar{d}_r^e, o^e)} \left(V_\psi(x_r, \bar{d}_r^e, o^e) - V(x_r; e) \leq \delta \right) \geq 1 - \epsilon$ holds with probability at least $1 - \beta$ over the draws of the samples as long as $\beta \geq I_{1-\epsilon}(N-k, k+1) = \sum_{i=0}^{k} \binom{N}{i} \epsilon^i (1-\epsilon)^{N-i}$. This is the same requirement on k as presented in Theorem 1. □

APPENDIX B
ALL EXPERIMENT RESULTS

Simulation experiments are conducted in easy, medium, and hard conditions to determine the effect of increasing dynamical variation on framework efficacy. The experiment difficulty determines the range of possible payload and friction parameters, as listed in Table V. As the difficulty increases, the parameter ranges stray further from normal conditions. We report results for the hard condition in Section VI in the main text. All simulation results can be found in Table VI.

All hardware results can be found in Table VII.

TABLE V: Simulation Environment Parameters

Payload Range (kg)			**Friction Range**		
Easy	Medium	Hard	Easy	Medium	Hard
$[0, 0]$	$[-1, 1]$	$[-1, -0.5] \cup [0.5, 1]$	$[1, 1]$	$[0.5, 1.5]$	$[0.5, 0.75] \cup [1.25, 1.5]$

TABLE VI: All Simulation Results (100 Trials)

Difficulty	**Controller**	**Filter**	**Success Rate ↑**	**Collision Rate ↓**	**Timeout Rate ↓**	$\bar{\mathbf{v}}$ **(m/s) ↑**	$\bar{\mathbf{r}}$ **↓**	$\bar{\mathbf{q}}$ **(m) ↑**
Easy	ABS-Agile	No Filter	0.89	0.11	0.00	**2.12**	**0.00**	0.44
		ABS Framework	0.88	0.12	0.00	2.08	0.02	0.44
		OCR \ DE	0.51	0.21	0.28	0.86	0.68	0.44
		OCR \ C	0.88	0.12	0.00	1.77	0.26	0.46
		OCR (ours)	**0.95**	**0.05**	0.00	1.24	0.60	**0.56**
	PS + WTW	No Filter	0.80	0.20	0.00	**1.46**	**0.00**	0.44
		OCR \ DE	0.85	0.15	0.00	0.84	0.62	0.43
		OCR \ C	0.91	0.09	0.00	1.31	0.14	0.42
		OCR (ours)	**1.00**	**0.00**	0.00	0.99	0.46	**0.62**
	NVE + WTW	No Filter	0.25	0.75	0.00	**1.82**	**0.00**	0.36
		OCR \ DE	0.22	0.78	0.00	1.23	0.50	0.40
		OCR \ C	0.46	0.54	0.00	1.53	0.22	0.37
		OCR (ours)	**0.98**	**0.02**	0.00	1.06	0.48	**0.58**
Medium	ABS-Agile	No Filter	0.83	0.17	0.00	**2.12**	**0.00**	0.43
		ABS Framework	0.87	0.13	0.00	2.05	0.02	0.43
		OCR \ DE	0.53	0.27	0.20	1.02	0.61	0.45
		OCR \ C	0.84	0.16	0.00	1.74	0.27	0.44
		OCR (ours)	**0.93**	**0.07**	0.00	1.27	0.57	**0.55**
	PS + WTW	No Filter	0.81	0.19	0.00	**1.42**	**0.00**	0.46
		OCR \ DE	0.81	0.19	0.00	0.86	0.61	0.44
		OCR \ C	0.83	0.17	0.00	1.33	0.12	0.46
		OCR (ours)	**0.99**	**0.01**	0.00	1.00	0.42	**0.65**
	NVE + WTW	No Filter	0.25	0.75	0.00	**1.79**	**0.00**	0.39
		OCR \ DE	0.22	0.78	0.00	1.11	0.56	0.40
		OCR \ C	0.42	0.58	0.00	1.47	0.22	0.39
		OCR (ours)	**0.97**	**0.03**	0.00	1.05	0.50	**0.57**
Hard	ABS-Agile	No Filter	0.75	0.25	0.00	**2.10**	**0.00**	0.41
		ABS Framework	0.80	0.20	0.00	2.03	0.03	0.41
		OCR \ DE	0.40	0.35	0.25	0.98	0.62	0.41
		OCR \ C	0.81	0.19	0.00	1.70	0.28	0.41
		OCR (ours)	**0.91**	**0.09**	0.00	1.22	0.59	**0.58**
	PS + WTW	No Filter	0.72	0.28	0.00	**1.36**	**0.00**	0.43
		OCR \ DE	0.78	0.22	0.00	0.84	0.61	0.42
		OCR \ C	0.90	0.09	0.01	1.21	0.18	0.44
		OCR (ours)	**1.00**	**0.00**	0.00	0.97	0.42	**0.66**
	NVE + WTW	No Filter	0.20	0.80	0.00	**1.70**	**0.00**	0.31
		OCR \ DE	0.21	0.79	0.00	0.93	0.61	0.37
		OCR \ C	0.45	0.55	0.00	1.32	0.28	0.39
		OCR (ours)	**0.91**	**0.08**	0.01	1.04	0.47	**0.62**

TABLE VII: All Hardware Results (10 Trials)

Condition	Controller	Filter	# Successes ↑	# Collisions ↓	# Timeouts ↓	$\bar{v}$ (m/s) ↑	$\bar{r}$ ↓	$\bar{q}$ (m) ↑
Normal	ABS-Agile	No Filter	9	1	0	**1.67**	**0.00**	0.29
		ABS	9	1	0	1.59	0.02	0.29
		OCR (ours)	**10**	**0**	0	0.94	0.64	**0.43**
	PS + WTW	No Filter	5	5	0	**1.53**	**0.00**	0.32
		OCR (ours)	**9**	**1**	0	0.80	0.58	**0.61**
	NVE + WTW	No Filter	0	10	0	N/A	N/A	N/A
		OCR (ours)	**9**	**0**	1	**0.77**	**0.61**	**0.55**
	PS + MPC	No Filter	5	5	0	**1.55**	**0.00**	0.33
		OCR (ours)	**10**	**0**	0	0.81	0.54	**0.58**
	NVE + MPC	No Filter	0	10	0	N/A	N/A	N/A
		OCR (ours)	**9**	**0**	1	**0.93**	**0.56**	**0.54**
Slippery	ABS-Agile	No Filter	1	9	0	1.38	**0.00**	0.33
		ABS	2	8	0	**1.49**	0.03	0.22
		OCR (ours)	**8**	**1**	1	0.90	0.60	**0.38**
	PS + WTW	No Filter	1	9	0	**1.04**	**0.00**	**0.52**
		OCR (ours)	**8**	**2**	0	0.74	0.65	0.45
	NVE + WTW	No Filter	0	10	0	N/A	N/A	N/A
		OCR (ours)	**8**	**2**	0	**0.61**	**0.68**	**0.39**
	PS + MPC	No Filter	0	10	0	N/A	N/A	N/A
		OCR (ours)	**9**	**1**	0	**0.80**	**0.54**	**0.56**
	NVE + MPC	No Filter	0	10	0	N/A	N/A	N/A
		OCR (ours)	**7**	**3**	0	**0.82**	**0.57**	**0.32**

APPENDIX C
BIOGRAPHY SECTION

Albert Lin Ph.D. student in Aeronautics and Astronautics at Stanford University, where he is advised by Prof. Somil Bansal and is a member of the Safe and Intelligent Autonomy Lab. He received his B.S.E. in Computer Science from Princeton University in 2023. His research interests are at the intersection of machine learning and cognitive science, especially as it relates to robot control and safety.

Shuang Peng Ph.D. student in Electrical and Computer Engineering at the University of Southern California, where he is a member of the Safe and Intelligent Autonomy Lab. He received his Bachelor's degree in Robotics from Southeast University, China. He is interested in exploring control and safety assurances for legged robots.

Somil Bansal Assistant professor in the Aeronautics and Astronautics department at Stanford University, where he leads the Safe and Intelligent Autonomy Lab. Previously, he was an assistant professor in the Electrical and Computer Engineering department at the University of Southern California. He received a Ph.D. in Electrical Engineering and Computer Sciences from the University of California, Berkeley in 2020. Before that, he obtained a B.Tech. in Electrical Engineering from the Indian Institute of Technology, Kanpur, and an M.S. in Electrical Engineering and Computer Sciences from UC Berkeley in 2012 and 2014, respectively. His research focuses on understanding how machine learning methods can be combined with classical, model-based planning and control methods to enable intelligent and safe decision-making.