SAFTER: SAFETY-AWARE FILTERED TRAINING FOR EFFICIENT AND ROBUST REINFORCEMENT LEARNING WITH CLOSED-FORM BACKUP CBFS

Anonymous authors
Paper under double-blind review

ABSTRACT

Safe reinforcement learning under bounded actuation requires more than preventing an immediate constraint violation: a currently admissible action can still leave insufficient control authority for future recovery. Backup control barrier functions (bCBFs) encode this finite-horizon recoverability, but sampled backup prediction introduces many simultaneous current, future, terminal, and input-admissibility conditions. Repeatedly solving the resulting optimization inside RL training is undesirable, and an opaque or approximated projection Jacobian can also give the actor a mismatched view of how its actions are actually executed. We propose SAFTER (**S**afety-**A**ware **F**iltered **T**raining for **E**fficient and **R**obust reinforcement learning), which converts sampled backup recoverability into a differentiable analytical training map. SAFTER augments the applied command with first-order dynamics, composes current-state, backup-horizon, terminal, and continuous-model input-admissibility conditions with a hierarchical Log-Sum-Exp certificate, and reduces the resulting CBF condition to a single affine half-space with an exact minimum-intervention projection. In the sample efficiency study, SAFTER reaches 90% training success after 4.06M transitions, versus 5.85M for Vanilla SAC and 6.16–6.26M for conventional CBF-filtered SAC, while closely matching the 4.13M transition count of bCBF–LSE–QP; this indicates predictive recoverability, rather than the closed-form realization alone, is primarily associated with the sample-efficiency gain. Under Low-, Medium-, and High-Shift geometries, SAFTER achieves 98.8%, 88.6%, and 61.4% success, improving over Vanilla SAC by up to 67.8 percentage points and over CBF–CF by up to 40.2 points. With the same correction reward, replacing the exact projection Jacobian by the identity reduces High-Shift success from 61.4% to 6.2%. These results show that backup recoverability improves learning efficiency, while the analytical, differentiable realization is important for learning policies that remain effective under geometric distribution shift.

1 INTRODUCTION

Reinforcement learning (RL) can produce adaptive navigation policies from interaction and local sensing (Xu et al., 2023; 2024), but trial-and-error exploration can generate unsafe actions during both training and deployment (Brunke et al., 2022; Guerrier et al., 2024; Xu et al., 2023). Reward shaping can improve navigation behavior and generalization (Botteghi et al., 2020; Miranda et al., 2023), while reward penalties only influence expected return and do not ensure that each executed action satisfies a prescribed safety condition (Krasowski et al., 2023). Safety filters address this limitation by separating the task policy, which proposes an action, from an explicit mechanism that modifies the action when necessary before execution. Incorporation of safety filters and explicit constraints into the learning process can reduce reward engineering and improve policy learning and sample efficiency (Kim et al., 2024; Bejarano et al., 2025; Van Wijk et al., 2024).

Control barrier functions (CBFs) provide a principled realization of this architecture by enforcing forward invariance under standard regularity and feasibility conditions (Ames et al., 2019). However, geometric barriers based primarily on current obstacle clearance may not explicitly encode the finite-horizon recovery dynamics imposed by bounded actuation. Under bounded actuation, that

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

question can be too late. A robot can remain collision-free at the current instant while entering a state from which its remaining braking or steering authority is insufficient to avoid a future violation. Backup CBFs (bCBFs) address this loss-of-recoverability failure mode by evaluating a prescribed recovery maneuver over a finite horizon and requiring the predicted trajectory to remain safe and reach a terminal backup set (Gurriet et al., 2018; Chen et al., 2021).

Using backup recoverability inside RL introduces both computational and learning challenges. Sampling the horizon creates many current, future, terminal, and input-admissibility conditions; in our LiDAR setting, 120 rays and $K = 20$ positive-time samples already yield 2400 future geometric evaluations per filter call, making direct sampled bCBF enforcement a repeated multi-constraint optimization. Moreover, when the filter is embedded in actor training, replacing its local projection Jacobian by an identity or treating it as opaque gives the policy a mismatched local action model.

These observations identify two distinct goals. *Predictive recoverability* should improve what the agent experiences and learns by intervening before control authority is lost. *Analytical differentiability* should make that predictive filter practical as a policy-training map by avoiding an iterative QP solve after composition and by exposing the exact local action transformation to the actor. Existing work addresses important subsets of this problem: smooth multi-CBF composition and closed-form RL filtering (Wang et al., 2025), optimization-based smooth backup barriers (Rabiee & Hoagg, 2025), closed-form bounded-input bCBF control through nominal–backup interpolation (van Wijk et al., 2026), and differentiable task-policy learning with learned and analytical backup constructions (Yun et al., 2026).

We introduce **SAFTER**, shown in Figure 1. The task policy proposes a nominal command. A prescribed backup rollout evaluates whether recovery remains feasible. Control-input augmentation and a hierarchical Log-Sum-Exp (LSE) construction compress current-state safety, sampled backup safety, terminal recovery, and continuous-model input admissibility into one differentiable scalar certificate. Its CBF condition is one affine constraint in a surrogate input, yielding an exact Euclidean projection. During SAC training, the executed filtered action generates the transition stored in replay, while the actor differentiates through the analytical safety map. The same filter remains active at deployment.

Our contributions are threefold. **First**, we formulate recoverability-aware policy learning as a safety-filter design problem in which finite-horizon backup safety and command admissibility must be enforced repeatedly inside RL while preserving minimum intervention. We construct an augmented composite bCBF certificate that reduces these conditions to a single affine constraint and therefore an exact full-space minimum-Euclidean projection. **Second**, we embed this analytical map directly into SAC: critics learn from the actions that actually generate transitions, and the actor receives the exact local Jacobian of the recoverability-aware filter. This distinguishes SAFTER from optimization-based predictive bCBF filters and from closed-form RL filters based only on current-state CBFs. **Third**, controlled experiments disentangle the effects of predictive recoverability and analytical differentiation: SAFTER and sampled bCBF–LSE–QP achieve nearly identical transition-based convergence and both substantially outperform current-state CBF baselines, whereas the exact-Jacobian ablation shows that differentiating through SAFTER is critical for robustness under geometric distribution shift.

2 Related Work

Safety-filtered reinforcement learning. Safety layers and shielding mechanisms separate task learning from explicit safety enforcement, while constrained and safety-critic formulations provide alternative mechanisms for safe policy learning (Chow et al., 2021; Hsu et al., 2023; Yang et al., 2023). Reward shaping can bias policies away from unsafe behavior (Nilaksh et al., 2024), while explicit locomotion constraints can reduce reward engineering (Kim et al., 2024). Training-time safety filtering has also been shown to help the policy account for filter interventions and improve sample efficiency (Bejarano et al., 2025; Van Wijk et al., 2024). Reachability-based approaches can switch between task and recovery policies (He et al., 2024), while observation-conditioned reachability filtering has been developed for quadruped navigation in previously unknown environments (Lin et al., 2026). In legged navigation, SEA–Nav learns safe and agile LiDAR-based locomotion (Chen et al., 2026), and CBF–RL combines closed-form CBF filtering with barrier-inspired reward shaping so that the task policy can internalize safety behavior (Yang et al., 2025). These approaches

motivate training-time safety integration but do not directly provide sampled finite-horizon backup recoverability through the task-policy action map.

Backup CBFs and predictive safety. Backup CBFs characterize recoverability through the flow of a prescribed backup controller, beginning with active-set invariance formulations (Gurriet et al., 2018) and later extensions using time-varying backups (Singletary et al., 2022). RLBUS learns a neural backup policy to enlarge the finite-time safe backward image (Rabiee & Safari, 2025); PS2-RL couples task-policy learning with a differentiable backup-CBF projection and considers both learned and analytical backup constructions (Yun et al., 2026). Backup-CBF filtering has also been developed for bounded-input and higher-order systems (van Wijk et al., 2026; Lin & Wu, 2026). SAFTER instead combines an augmented composite backup certificate with an explicit projection in surrogate-control space and differentiates through this map during off-policy task-policy learning.

Composite and explicit safety filters. LSE composition has been used to combine multiple current-state CBF constraints and obtain a closed-form safety filter for end-to-end RL (Wang et al., 2025). CBF-based quadruped navigation has also used smooth LiDAR-derived safety functions with online QP filtering (Unlu et al., 2024). For backup barriers, smooth minimum/maximum constructions aggregate finite-horizon conditions while retaining optimization-based control synthesis (Rabiee & Hoagg, 2025). van Wijk et al. (2026) derive a closed-form bounded-input bCBF controller by interpolating between nominal and backup controls, restricting the correction to that interpolation direction. SAFTER instead constructs a single composite affine constraint in an augmented surrogate-input space and applies the exact minimum-Euclidean correction in the full surrogate control space. The differentiable mapping is then used directly in the SAC actor update.

3 Recoverability-Aware Safe RL: Problem Formulation

Safety-filtered Soft Actor-Critic RL. We consider continuous-action reinforcement learning with environment state s_t, policy observation o_t, and physical command u_t. The task policy is trained using Soft Actor-Critic (SAC) (Haarnoja et al., 2018). At policy step t, the actor proposes a nominal command $u_t^{\mathrm{nom}} \sim \pi_\phi(\cdot \mid o_t)$, which is transformed before execution by

$$u_t^{\mathrm{safe}} = \mathcal{F}_\eta(\xi_t, u_t^{\mathrm{nom}}), \tag{1}$$

where ξ_t contains the state and sensing information required by the filter and η denotes its fixed design parameters. Because the executed command generates the environment transition, it is the action stored in replay and evaluated by the SAC critics. During the actor update, SAFTER also differentiates through $\mathcal{F}_\eta$, allowing the nominal policy to adapt to the safety intervention it experiences. Full SAC objectives are given in Appendix D.

The safety filter need not model the complete RL environment. We use $x \in \mathbb{R}^n$ for the continuous state required by the safety model and $u \in \mathcal{U}$ for its physical command, with local dynamics $\dot{x} = f(x) + g(x)u$. Thus, s_t is the RL environment state, o_t is the actor observation, and x is the reduced state used to evaluate safety.

From current-state safety to backup recoverability. Let $\mathcal{C}_S = \{x : h_i(x) \geq 0,\ i = 1, \ldots, N_h\}$ denote the prescribed safe set. For a differentiable relative-degree-one barrier h, a standard minimum-intervention CBF filter solves

$$\begin{aligned} u^{\mathrm{safe}} = \ & \arg\min_{u \in \mathcal{U}} \frac{1}{2}\|u - u^{\mathrm{nom}}\|_2^2 \\ \text{s.t.} \quad & L_f h(x) + L_g h(x) u + \alpha(h(x)) \geq 0, \end{aligned} \tag{2}$$

where $L_f h(x) = \nabla h(x)^\top f(x)$ and $L_g h(x) = \nabla h(x)^\top g(x)$ are the Lie derivatives of h, and α regulates approach to the safety boundary. Under the usual regularity and feasibility assumptions, the CBF constraint renders the corresponding safe set forward invariant (Ames et al., 2019); the objective leaves the nominal command unchanged whenever it already satisfies the input and CBF constraints.

A conventional CBF constrains the instantaneous safety-margin dynamics, but with bounded actuation this need not preserve future recoverability. Let $u_{\mathrm{b}}(x)$ be a prescribed backup controller, $\varphi_{\mathrm{b}}(\tau; x)$ its flow, and $\mathcal{C}_B^x = \{x : h_{B,j}(x) \geq 0,\ j = 1, \ldots, N_B\} \subseteq \mathcal{C}_S$ a terminal set invariant under u_{b} (Gurriet et al., 2018; Chen et al., 2021). Finite-horizon recoverability requires

$$h_i(\varphi_{\mathrm{b}}(\tau; x)) \geq 0, \quad \forall \tau \in [0, T],\ \forall i, \qquad h_{B,j}(\varphi_{\mathrm{b}}(T; x)) \geq 0, \quad \forall j.$$

The backup trajectory must therefore remain safe while reaching a terminal state from which safety can subsequently be maintained. Evaluating the trajectory at $0 < \tau_1 < \cdots < \tau_K = T$ produces $N_h K$ future safety conditions in addition to current and terminal conditions. Direct enforcement therefore leads to the sampled multi-constraint bCBF–QP described in Appendix A.1: recoverability provides predictive safety information unavailable to an instantaneous CBF, but substantially enlarges the online safety problem.

Running quadruped-navigation example. In our experiments, s_t contains the full MuJoCo state, while the SAC actor observes local LiDAR, goal information, measured body velocity, and command history. The safety model uses the scan-fixed planar pose $x = [p_x, p_y, \theta]^\top$ and high-level command $u = [v_x, v_y, \omega_z]^\top$. Each valid LiDAR return defines an obstacle point and barrier $h_i(x)$. A command can satisfy all current clearance constraints yet leave too little distance to stop before collision. The prescribed backup therefore drives the commanded motion toward rest and tests whether this stopping maneuver remains collision-free over the horizon. With $N_h = 120$ rays and $K = 20$ positive-time samples, the filter evaluates 2400 future geometric barriers before terminal and input-admissibility conditions are included. Appendix B gives the full construction.

Problem 3.1 (Recoverability-aware differentiable safe RL)**.** Given π_ϕ, the safety model (f, g), safe set $\mathcal{C}_S$, bounded input set $\mathcal{U}$, backup controller u_{b}, and terminal set $\mathcal{C}_B^x$, construct $\mathcal{F}_\eta$ such that the executed command: (i) satisfies the encoded current, sampled backup, terminal, and input conditions whenever feasible; (ii) minimally modifies the nominal task command; and (iii) can be evaluated efficiently and differentiated during SAC training.

Direct sampled bCBF enforcement satisfies the predictive requirement but retains a multi-constraint optimization inside every RL interaction. Section 4 develops SAFTER to remove this bottleneck by combining bounded-command augmentation, finite-horizon backup prediction, a single composite certificate, and an analytical minimum-intervention projection whose local Jacobian can be used directly in policy optimization.

4 Recoverability-Aware Differentiable Safety Filtering

SAFTER realizes Problem 3.1 through the analytical pipeline in Figure 1. It augments the bounded physical command with first-order dynamics, evaluates finite-horizon backup recoverability in the augmented model, and combines the resulting current, future, terminal, and input-admissibility margins into one differentiable certificate. The CBF condition of this certificate is a single affine constraint in a surrogate input, yielding an exact minimum-intervention projection and an exact local Jacobian wherever the filter is differentiable.

The central technical contribution is this reduction from sampled multi-constraint backup-CBF enforcement to a single analytical projection after the constituent safety margins have been evaluated. Thus, the number of backup samples and primitive barriers no longer determines the number of constraints in the online optimization, while the same recoverability-aware action map can be differentiated through during SAC training.

Bounded-command augmentation and current safety. Section 3 uses the safety model $\dot{x} = f(x) + g(x)u$ with bounded physical command $u \in \mathcal{U}$. Following the control-dynamics construction of Rabiee & Hoagg (2024), we introduce a surrogate input $\hat{u} \in \mathbb{R}^m$ through $\dot{u} = -Au + A\hat{u}$, where $A = \mathrm{diag}(a_1, \ldots, a_m) \succ 0$. Treating the applied command as part of the state gives $z = [x^\top, u^\top]^\top$ and

$$\dot{z} = \hat{f}(z) + \hat{g}(z)\hat{u}, \qquad \hat{f}(z) = \begin{bmatrix} f(x) + g(x)u \\ -Au \end{bmatrix}, \qquad \hat{g}(z) = \begin{bmatrix} 0_{n\times m} \\ A \end{bmatrix}. \tag{3}$$

For the component-wise bounds $u_{\min} \le u \le u_{\max}$, define $I_d^+(z) := u_{\max,d} - u_d$ and $I_d^-(z) := u_d - u_{\min,d}$, $d = 1, \ldots, m$. Their nonnegativity represents the physical command bounds as state constraints. Because h_i depends on x but not directly on the command state u, $L_{\hat{g}} h_i = 0$. We therefore use the lifted current-safety margin

$$\psi_i(z) := L_f h_i(x) + L_g h_i(x) u + k_0 h_i(x), \qquad L_{\hat{g}} \psi_i(z) = L_g h_i(x) A,$$

with $k_0 > 0$. Hence, ψ_i has relative degree one with respect to $\hat{u}$ wherever $L_g h_i(x) A \neq 0$ and can be combined with the predictive and input-admissibility conditions below. Appendix A.3 gives the relative-degree calculation, while Appendix C describes the finite-rate command update and the post-projection enforcement of the physical command limits.

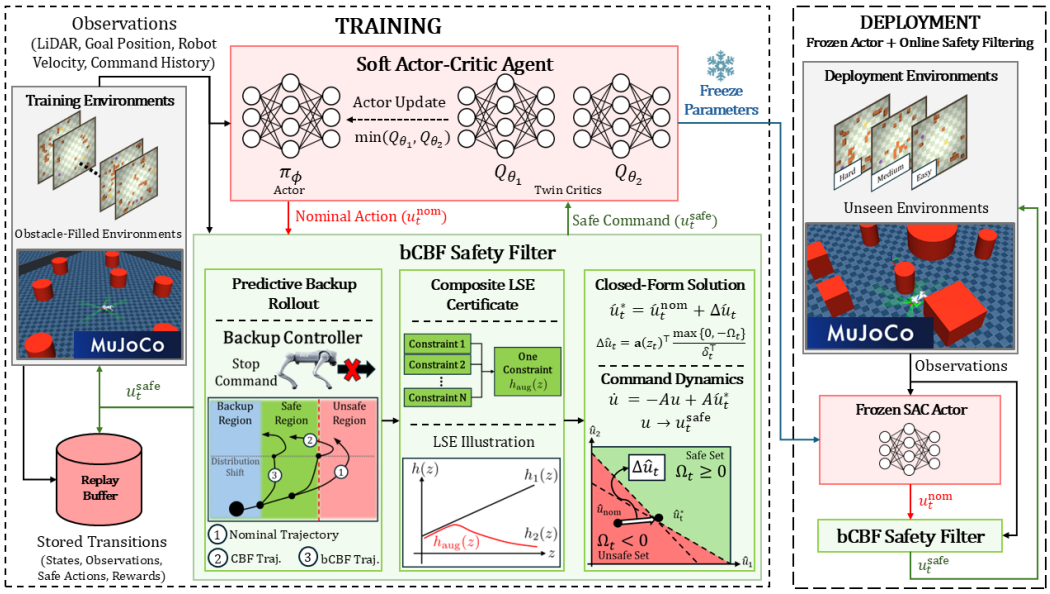


Figure 1: Overview of SAFTER. A task policy proposes a nominal command; a prescribed backup rollout evaluates finite-horizon recoverability; current, future, terminal, and input-admissibility conditions are combined into one differentiable certificate; and an analytical minimum-intervention projection determines the safety correction before execution. During training, the critic evaluates the executed action and the actor differentiates through the same safety map.

Predictive backup recoverability. Because SAFTER acts through $\hat{u}$, the backup maneuver u_{b} from Section 3 is represented in the augmented dynamics by a prescribed surrogate backup input $\hat{u}_{\text{b}}(z)$. Let $\varphi_{\text{b}}^z(\tau; z)$ denote the corresponding flow of $\dot{z} = \hat{f}(z) + \hat{g}(z)\hat{u}_{\text{b}}(z)$. With $E_x = [I_n \ 0_{n\times m}]$, the sampled future margins are $h_i^{\text{b}}(z, \tau_k) := h_i(E_x\varphi_{\text{b}}^z(\tau_k; z))$. After augmentation, terminal recoverability may also depend on the applied command state. We therefore lift the terminal set to $\mathcal{C}_B^z = \{z : h_{B,j}(z) \geq 0,\ j = 1, \ldots, N_B\}$ and define $h_{B,j}^{\text{b}}(z, T) := h_{B,j}(\varphi_{\text{b}}^z(T; z))$. The future margins test sampled safety along the recovery trajectory, while the terminal margins test whether the predicted backup state reaches $\mathcal{C}_B^z$. Their gradients require the sensitivity of the backup flow to the current augmented state; Appendix A.3.1 gives the corresponding variational dynamics.

For the quadruped specialization, $\hat{u}_{\text{b}} = 0$, so the first-order command dynamics drive the applied translational and yaw-rate commands toward zero. The backup test therefore asks whether sufficient obstacle clearance remains to dissipate the current commanded motion.

Composite recoverability certificate. The backup construction above yields the current margins $\psi_i(z)$, sampled future margins $h_i^{\text{b}}(z, \tau_k)$, terminal margins $h_{B,j}^{\text{b}}(z, T)$, and input-admissibility margins $I_d^{\pm}(z)$. After fixed positive rescaling, denoted by an overbar, SAFTER combines them using the LSE smooth minimum (Molnar & Ames, 2023; Rabiee & Hoagg, 2025),

$$\underset{\kappa}{\text{softmin}}(\mathcal{S}) := -\frac{1}{\kappa}\log\sum_{c\in\mathcal{S}} e^{-\kappa c}, \qquad \min\mathcal{S} - \frac{\log|\mathcal{S}|}{\kappa} \leq \underset{\kappa}{\text{softmin}}(\mathcal{S}) \leq \min\mathcal{S}, \tag{4}$$

where $\kappa > 0$ controls the approximation sharpness. Thus, $\text{softmin}_\kappa(\mathcal{S}) \geq 0$ implies nonnegativity of every constituent in $\mathcal{S}$. We aggregate recoverability and input admissibility separately,

$$h_{\text{rec}}(z) := \underset{\kappa_B}{\text{softmin}}\big(\{\bar{\psi}_i(z)\}_i \cup \{\bar{h}_i^{\text{b}}(z, \tau_k)\}_{i,k} \cup \{\bar{h}_{B,j}^{\text{b}}(z, T)\}_j\big)\,,$$
$$h_{\text{input}}(z) := \underset{\kappa_U}{\text{softmin}}\big(\{\bar{I}_d^+(z), \bar{I}_d^-(z)\}_{d=1}^m\big)\,.$$

The complete certificate is $h_{\text{aug}}(z) := \text{softmin}_{\kappa_O}(\{h_{\text{rec}}(z), h_{\text{input}}(z)\})$. Hence, $h_{\text{aug}}(z) \geq 0$ implies all encoded current, sampled backup, terminal, and input-admissibility conditions. The individual margins must still be evaluated, but their enforcement is reduced to one differentiable scalar certificate.

Analytical minimum-intervention projection. The composite certificate h_{aug} now plays the role of the barrier function in the CBF condition introduced in equation 2. Applied to the augmented dynamics equation 3, this condition becomes

$$L_{\hat{f}} h_{\mathrm{aug}}(z) + L_{\hat{g}} h_{\mathrm{aug}}(z)\hat{u} + \alpha_{\mathrm{aug}} h_{\mathrm{aug}}(z) \geq 0, \tag{5}$$

where $\alpha_{\mathrm{aug}} > 0$ is the composite CBF gain. Defining $\mathbf{a}(z) := L_{\hat{g}} h_{\mathrm{aug}}(z)$ and $b(z) := L_{\hat{f}} h_{\mathrm{aug}}(z) + \alpha_{\mathrm{aug}} h_{\mathrm{aug}}(z)$, equation 5 is simply $\mathbf{a}(z)\hat{u} + b(z) \geq 0$: a single half-space in the surrogate input. Thus, the many sampled recoverability conditions composed above no longer appear as separate constraints in the online projection. The actor, however, proposes the physical command u_t^{nom} rather than the surrogate input $\hat{u}$. To recover this nominal command when the safety constraint is inactive, we use

$$\hat{u}_t^{\mathrm{nom}} = u_t + A^{-1}\dot{u}_t^{\mathrm{nom}} + A^{-1}\Sigma\left(u_t^{\mathrm{nom}} - u_t\right), \qquad \Sigma \succ 0, \tag{6}$$

which makes the applied command u exponentially track u^{nom} under the augmented dynamics. The derivation is given in Appendix C.1. The safety filter therefore solves

$$\begin{aligned} \hat{u}_t^{\star} = \; & \arg\min_{\hat{u}\in\mathbb{R}^m} \frac{1}{2}\|\hat{u} - \hat{u}_t^{\mathrm{nom}}\|_2^2 \\ \text{s.t.} \quad & \mathbf{a}(z_t)\hat{u} + b(z_t) \geq 0. \end{aligned} \tag{7}$$

Let $\Omega_t := \mathbf{a}(z_t)\hat{u}_t^{\mathrm{nom}} + b(z_t)$ denote the composite constraint margin of the nominal surrogate, and $\delta_t := a(z_t)a(z_t)^\top = \|a(z_t)\|_2^2$ denote the squared constraint-normal magnitude.

Proposition 4.1 (Closed-form safety projection)**.** *If $\delta_t > 0$, the unique solution of equation 7 is*

$$\hat{u}_t^{\star} = \hat{u}_t^{\mathrm{nom}} + \underbrace{\mathbf{a}(z_t)^\top \frac{\max\{0, -\Omega_t\}}{\delta_t}}_{\Delta\hat{u}_t}. \tag{8}$$

If $\Omega_t \geq 0$, the nominal surrogate already satisfies the composite condition and is unchanged; otherwise, equation 8 applies the minimum-Euclidean correction along the certificate normal. The projection is exact for the resulting half-space, and approximation enters only through finite backup sampling and the LSE construction preceding it. The detailed proof of Proposition 4.1 is included in Appendix A.4.

Theorem 4.2 (Continuous-time safety and recoverability)**.** *Under Assumption A.1, suppose $z(0) \in \mathcal{C}_{\mathrm{aug}}$ and the composite CBF condition equation 5 is enforced continuously. Then $\mathcal{C}_{\mathrm{aug}} = \{z : h_{\mathrm{aug}}(z) \geq 0\}$ is forward invariant for the continuous augmented model. Consequently, the safety-model state remains in the original safe set $\mathcal{C}_S$, the applied command remains within its prescribed bounds, and the backup trajectory satisfies the encoded safety conditions at the sampled future times and reaches the terminal recovery set at the end of the horizon.*

The result applies to the continuous safety model and to backup safety at the selected horizon samples. The finite-rate implementation, inter-sample backup behavior, sensing updates, and reduced-model assumptions are discussed in Appendices C.2 and C.3.

SAFTER–SAC integration. As illustrated in Figure 1, SAFTER places the recoverability-aware safety map $\mathcal{F}_\eta$ between the SAC actor and the environment. At each step, the actor proposes u_t^{nom}, the safety map produces the executed command u_t^{safe}, and the environment transition is generated by this filtered action. Accordingly, u_t^{safe} is stored in replay and evaluated by the critics, so value learning is based on the action that actually generated the transition. The same filtering is applied to next actions when constructing SAC targets.

During the actor update, the critic evaluates $Q(o_t, \mathcal{F}_\eta(\xi_t, u_t^{\mathrm{nom}}))$, and gradients are propagated through the same finite-rate map that generates experience: the nominal-surrogate construction, analytical half-space projection, finite-rate command update, and final component-wise clipping. Away from the projection and clipping switching boundaries, this map has the local Jacobian derived in Eq. (17) and is therefore differentiable almost everywhere. This allows the policy to learn how its nominal commands are transformed by the safety intervention rather than treating the filter as a detached execution-time shield. Target filtering uses the same forward map without gradient propagation, while the SAC entropy term remains defined on the nominal policy. Full objectives and the Jacobian derivation are given in Appendix D.1; Table 4 isolates the effects of the projection Jacobian and correction reward.

Table 1: Performance on the training-map distribution with each method's runtime safety filter disabled or enabled. SR, TR, and FR denote success, timeout, and failure rates, respectively. Values are mean ± standard deviation over evaluation seeds and maps.

	Filter disabled			Filter enabled		
Method	**SR ↑**	**TR ↓**	**FR ↓**	**SR ↑**	**TR ↓**	**FR ↓**
SEA–Nav[1]	89.7±2.4	0.0±0.0	10.3±2.4	91.2±2.9	0.0±0.0	8.8±2.9
CBF–RL[2]	84.3±8.2	5.9±5.2	9.8±7.1	84.8±7.4	5.9±5.2	9.3±6.6
CBF–CF[3]	76.6±13.8	2.7±6.3	20.7±15.8	90.1±8.2	1.9±4.5	8.0±8.5
SAFTER[4]	**95.1±7.1**	0.0±0.0	**4.9±7.1**	**99.7±1.2**	0.0±0.0	**0.3±1.2**

Refs.: [1] (Chen et al., 2026); [2] (Yang et al., 2025); [3] (Rabiee & Hoagg, 2024); [4] **Ours**.

5 Experiments and Analysis

The experiments are organized to test the three mechanisms isolated in the problem formulation. **RQ1** asks whether finite-horizon backup recoverability, rather than current-state filtering alone, improves transition-based learning efficiency, and whether its analytical realization reduces safety-filter latency relative to QP-based synthesis. **RQ2** asks whether the resulting policy–filter system remains effective as obstacle geometry shifts away from the training maps. **RQ3** asks whether learning through the exact analytical projection Jacobian matters when the forward safety-filter mapping is unchanged.

The baselines and SAFTER are evaluated in MuJoCo 3.3.6 using a Unitree Go2 with a frozen 50 Hz low-level locomotion policy. The high-level policy operates at 10 Hz using local LiDAR, goal, measured-velocity, and command-history observations. SAC-based methods use 256 parallel environments, five independent training seeds, and a maximum budget of 20.48M transitions per seed. In both training and evaluation, failure denotes episode termination caused by obstacle collision, falling, or contact with the environment boundary wall. Detailed safety-filter parameters, rewards, SAC architecture and optimization settings, and map definitions are given in Appendices B.2.7, D.4, D.5, and E.1, respectively. The bCBF–LSE–QP baseline uses the same composite half-space, backup model, terminal set, command dynamics, and finite-rate realization as SAFTER. In exact arithmetic, both therefore yield the same forward Euclidean projection; the baseline obtains it through a differentiable QP solver, whereas SAFTER evaluates the analytical solution directly. This provides a controlled comparison of numerical and analytical synthesis under the same predictive recovery model.

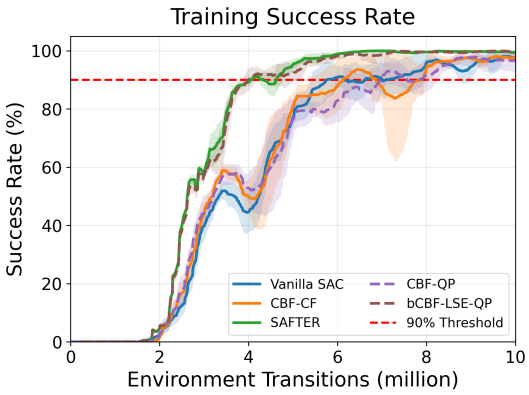


Figure 2: Training success rates of Vanilla SAC, CBF–QP/CBF–CF, bCBF–LSE–QP, and SAFTER under the same training environments.

RQ1: Learning Efficiency and Policy Internalization. For the controlled convergence study, sample efficiency is measured by the first attainment of 90% training success for each seed, using a 20-sample trailing rolling mean; the threshold is counted at its first crossing and is not required to persist. As shown in Figure 2, the two backup-recoverability methods converge earliest: SAFTER reaches the threshold after 4.06±0.08M transitions and bCBF–LSE–QP after 4.13±0.43M. SAFTER therefore requires 30.6% fewer transitions than Vanilla SAC and 34.1%–35.1% fewer than the current-state CBF baselines. The similar convergence of SAFTER and bCBF–LSE–QP indicates that the sample-efficiency gain is primarily associated with predictive backup recoverability rather than the analytical realization alone.

Table 2: Safety-filter runtime comparison. Total latency includes certificate construction and synthesis, and speedup is measured relative to the direct bCBF–QP. Values are mean ± standard deviation over 300 timed repetitions.

Method	Batch = 1			Batch = 100		
	Total (ms)	Synthesis (ms)	Speedup (×)	Total (ms)	Synthesis (ms)	Speedup (×)
bCBF–QP	30.546±0.324	28.369±0.194	1.00	2504.789±22.463	2487.383±16.949	1.00
bCBF–LSE–QP	5.732±0.456	3.342±0.405	5.33	7.084±0.569	4.588±0.481	353.58
SAFTER	**2.495±0.312**	**0.102±0.011**	**12.24**	**2.529±0.283**	**0.106±0.014**	**990.43**

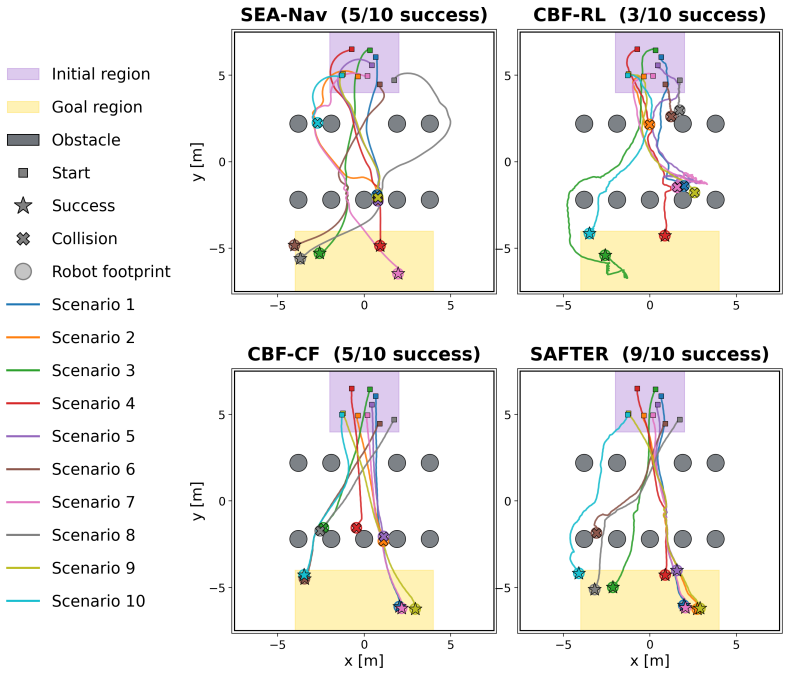


Figure 3: Representative trajectories in the Medium-Shift environment. Purple and yellow regions denote the initial and goal regions, squares mark initial positions, stars successful terminations, crosses collision terminations, and faint circles indicate the robot footprint.

Recoverability-aware training also appears to shape the learned nominal policy rather than relying solely on runtime filtering. To examine this effect, we evaluate the trained policies on the training-map distribution with their runtime safety filters enabled and disabled. Among the methods reported in Table 1, SAFTER achieves 95.1% success without the runtime filter and 99.7% with it. The strong filter-disabled performance suggests that recoverability-aware training promotes nominal behavior compatible with the prescribed backup strategy.

To evaluate computational efficiency, Table 2 compares SAFTER with the matched bCBF–LSE–QP and the direct unaggregated bCBF–QP. All methods are evaluated on identical fixed filter inputs using 30 warm-up calls and 300 timed repetitions on an NVIDIA RTX 4090 GPU with CUDA synchronization around each timed region. The matched bCBF–LSE–QP uses `qpth`, while the direct unaggregated bCBF–QP uses OSQP. Because the direct bCBF–QP and matched bCBF–LSE–QP use different QP solvers, comparisons against the direct formulation reflect end-to-end implementation latency rather than a solver-isolated speedup. SAFTER achieves the lowest total latency at both batch sizes, providing 12.24× and 990.43× lower latency than the direct bCBF–QP at batch sizes 1 and 100, respectively. Relative to the matched bCBF–LSE–QP, SAFTER is 2.30× and 2.80× faster.

RQ2: Robustness under Geometric Distribution Shift. Predictive backup recoverability improves robustness as obstacle geometry departs from the training distribution. For distribution-shift evalu-

Table 3: Navigation performance across unseen static environments. SR, TR, and FR denote success, timeout, and failure rates, respectively. Values are mean ± standard deviation across 25 evaluation-seed rates.

	Low-Shift			Medium-Shift			High-Shift		
Method	**SR ↑**	**TR ↓**	**FR ↓**	**SR ↑**	**TR ↓**	**FR ↓**	**SR ↑**	**TR ↓**	**FR ↓**
SEA–Nav[1]	93.8±6.7	0.0±0.0	6.2±6.7	50.6±10.9	0.0±0.0	49.4±10.9	0.0±0.0	0.0±0.0	100.0±0.0
CBF–RL[2]	61.2±10.7	13.6±6.7	25.2±9.8	46.2±11.6	17.2±7.6	36.6±9.2	31.0±10.0	15.4±8.9	53.6±12.2
Vanilla SAC[3]	34.2±10.1	0.0±0.0	65.8±10.1	20.8±6.9	0.0±0.0	79.2±6.9	17.4±8.3	0.0±0.0	82.6±8.3
CBF–QP[4]	58.8±9.4	0.2±1.0	41.0±9.2	43.2±11.0	0.0±0.0	56.8±11.0	18.0±6.9	0.0±0.0	82.0±6.9
CBF–CF[4]	69.0±11.5	0.0±0.0	31.0±11.5	48.4±8.4	0.0±0.0	51.6±8.4	33.8±12.2	0.0±0.0	66.2±12.2
bCBF–LSE–QP[5]	**99.4±1.7**	0.0±0.0	**0.6±1.7**	75.0±9.7	0.0±0.0	25.0±9.7	**67.4±11.7**	0.0±0.0	**32.6±11.7**
SAFTER[5]	98.8±2.2	0.0±0.0	1.2±2.2	**88.6±7.3**	0.0±0.0	**11.4±7.3**	61.4±10.2	0.0±0.0	38.6±10.2

Refs.: [1] (Chen et al., 2026); [2] (Yang et al., 2025); [3] (Haarnoja et al., 2018); [4] (Rabiee & Hoagg, 2024); [5] **Ours**.

Table 4: Ablation of the projection Jacobian and correction reward on unseen environments. Exact and Identity use the same forward safety-filter map and differ only in the backward differentiation through the analytical projection. Values are mean ± standard deviation across 25 evaluation-seed rates.

Ablation Setting		Low-Shift			Medium-Shift			High-Shift		
Projection Jacobian	**Correction Reward**	**SR↑**	**TR↓**	**FR↓**	**SR↑**	**TR↓**	**FR↓**	**SR↑**	**TR↓**	**FR↓**
Exact	On	**98.8±2.2**	0.0±0.0	1.2±2.2	88.6±7.3	0.0±0.0	11.4±7.3	61.4±10.2	0.0±0.0	38.6±10.2
Identity	On	89.8±6.0	0.0±0.0	10.2±6.0	41.6±9.3	0.0±0.0	58.4±9.3	6.2±4.4	0.0±0.0	93.8±4.4
Exact	Off	95.6±4.9	4.2±4.9	**0.2±1.0**	**94.4±4.4**	3.8±3.9	**1.8±3.5**	**76.2±9.2**	3.2±3.8	**20.6±8.3**
Identity	Off	76.8±7.1	0.0±0.0	23.2±7.1	9.0±6.1	0.0±0.0	91.0±6.1	2.0±3.5	0.0±0.0	98.0±3.5

ation, each method uses a fixed checkpoint selected independently of unseen-map performance and is evaluated over 25 seeds with 20 episodes each (500 episodes per environment), with statistics computed across evaluation-seed rates. As shown in Table 3, SAFTER achieves 98.8%, 88.6%, and 61.4% success on Low-, Medium-, and High-Shift, respectively, exceeding Vanilla SAC and both current-state CBF baselines across all three environments. The matched bCBF–LSE–QP achieves similarly strong performance at 99.4%, 75.0%, and 67.4%, indicating that the robustness gain is primarily associated with predictive backup recoverability rather than the analytical realization alone. Figure 3 provides qualitative support on Medium-Shift, while path-length, navigation-time, and guarantee-scope results are reported in Appendices E.2 and C.3.

RQ3: Why Does SAFTER Learn Differently? Accurate differentiation through the analytical projection has a substantially larger and more consistent effect on robustness than discouraging filter intervention. As shown in Table 4, replacing the Exact projection Jacobian with the Identity approximation reduces success by up to 85.4 percentage points across the evaluated distribution shifts. In contrast, removing the correction reward has a mixed effect: the Exact-Jacobian policy improves from 88.6% to 94.4% on Medium-Shift and from 61.4% to 76.2% on High-Shift, while decreasing from 98.8% to 95.6% on Low-Shift. Because the forward safety-filter map is unchanged within each Exact/Identity pair, these results provide evidence that propagating the analytical projection Jacobian materially improves robustness, whereas minimizing filter intervention is not uniformly beneficial under larger geometric shifts.

6 Conclusion

We introduced SAFTER, a recoverability-aware safe-RL framework that combines command-state augmentation, hierarchical LSE composition, and an exact minimum-Euclidean projection into a differentiable training map. Experiments show improved transition-based sample efficiency and strong performance under geometric distribution shift, while the Jacobian ablation highlights the importance of differentiating through the analytical safety-filter map. Future work will consider uneven terrain and dynamic obstacles.

AI USE STATEMENT

In this work, we used generative AI tools to aid and polish manuscript writing and to assist with research execution through debugging and modifying portions of the research code. We have not used generative AI tools to generate experimental data or make final scientific conclusions, and other disclosure categories not relevant to these uses are not applicable to this work. We reviewed all AI-assisted work. AI-assisted code was manually inspected and tested for correctness, while AI-assisted text was reviewed and revised by the authors for technical accuracy and consistency with the implemented methods and reported results. We take responsibility for the final content of this work, including text, claims, code, and artifacts produced with the aid of generative AI.

REFERENCES

Aaron D Ames, Samuel Coogan, Magnus Egerstedt, Gennaro Notomista, Koushil Sreenath, and Paulo Tabuada. Control barrier functions: Theory and applications. In *Proc. 18th European Control Conf.*, pp. 3420–3431, 2019.

Federico Pizarro Bejarano, Lukas Brunke, and Angela P. Schoellig. Safety filtering while training: Improving the performance and sample efficiency of reinforcement learning agents. *IEEE Robotics and Automation Letters*, 10(1):788–795, 2025. doi: 10.1109/LRA.2024.3512374.

Nicolò Botteghi, Beril Sirmacek, Khaled AA Mustafa, Mannes Poel, and Stefano Stramigioli. On reward shaping for mobile robot navigation: A reinforcement learning and slam based approach. *arXiv preprint arXiv:2002.04109*, 2020.

Lukas Brunke, Melissa Greeff, Adam W Hall, Zhaocong Yuan, Siqi Zhou, Jacopo Panerati, and Angela P Schoellig. Safe learning in robotics: From learning-based control to safe reinforcement learning. *Annual Review of Control, Robotics, and Autonomous Systems*, 5(1):411–444, 2022.

Shiyi Chen, Mingye Yang, Haiyan Mao, Jiaqi Zhang, Haiyi Liu, Shuheng He, Debing Zhang, Zihao Qiu, and Chun Zhang. Sea-nav: Efficient policy learning for safe and agile quadruped navigation in cluttered environments. *arXiv preprint arXiv:2603.09460*, 2026.

Yuxiao Chen, Mrdjan Jankovic, Mario Santillo, and Aaron D. Ames. Backup control barrier functions: Formulation and comparative study. In *2021 60th IEEE Conference on Decision and Control (CDC)*, pp. 6835–6841. IEEE, 2021. doi: 10.1109/CDC45484.2021.9683111.

Yinlam Chow, Ofir Nachum, Aleksandra Faust, Edgar Dueñez-Guzman, and Mohammad Ghavamzadeh. Safe policy learning for continuous control. In *Proceedings of the 2020 Conference on Robot Learning*, volume 155 of *Proceedings of Machine Learning Research*, pp. 801–821. PMLR, 2021.

Maeva Guerrier, Hassan Fouad, and Giovanni Beltrame. Learning control barrier functions and their application in reinforcement learning: A survey. *arXiv preprint arXiv:2404.16879*, 2024.

Thomas Gurriet, Mark Mote, Aaron D Ames, and Eric Feron. An online approach to active set invariance. In *2018 IEEE Conference on Decision and Control (CDC)*, pp. 3592–3599. IEEE, 2018.

Tuomas Haarnoja, Aurick Zhou, Kristian Hartikainen, George Tucker, Sehoon Ha, Jie Tan, Vikash Kumar, Henry Zhu, Abhishek Gupta, Pieter Abbeel, et al. Soft actor-critic algorithms and applications. *arXiv preprint arXiv:1812.05905*, 2018.

Tairan He, Chong Zhang, Wenli Xiao, Guanqi He, Changliu Liu, and Guanya Shi. Agile but safe: Learning collision-free high-speed legged locomotion. In *Proceedings of Robotics: Science and Systems*, 2024. doi: 10.15607/RSS.2024.XX.059.

Kai-Chieh Hsu, Duy Phuong Nguyen, and Jaime Fernàndez Fisac. ISAACS: Iterative soft adversarial actor-critic for safety. In *Proceedings of the 5th Annual Learning for Dynamics and Control Conference*, volume 211 of *Proceedings of Machine Learning Research*, pp. 90–103. PMLR, 2023.

Yunho Kim, Hyunsik Oh, Jeonghyun Lee, Jinhyeok Choi, Gwanghyeon Ji, Moonkyu Jung, Donghoon Youm, and Jemin Hwangbo. Not only rewards but also constraints: Applications on legged robot locomotion. *IEEE Transactions on Robotics*, 40:2984–3003, 2024. doi: 10.1109/TRO.2024.3400935.

Hanna Krasowski, Jakob Thumm, Marlon Müller, Lukas Schäfer, Xiao Wang, and Matthias Althoff. Provably safe reinforcement learning: Conceptual analysis, survey, and benchmarking. *Transactions on Machine Learning Research*, 2023.

Albert Lin, Shuang Peng, and Somil Bansal. One filter to deploy them all: Robust safety for quadrupedal navigation in unknown environments. *IEEE Transactions on Robotics*, 42:545–560, 2026. doi: 10.1109/TRO.2025.3644957.

Zijie Lin and Baolin Wu. Safety-critical control for multiple spacecraft close-proximity inspection. *Aerospace Science and Technology*, 168:110938, 2026.

Victor RF Miranda, Armando A Neto, Gustavo M Freitas, and Leonardo A Mozelli. Generalization in deep reinforcement learning for robotic navigation by reward shaping. *IEEE Transactions on Industrial Electronics*, 71(6):6013–6020, 2023.

Tamas G Molnar and Aaron D Ames. Composing control barrier functions for complex safety specifications. *IEEE Control Systems Letters*, 7:3615–3620, 2023.

Nilaksh, Abhishek Ranjan, Shreenabh Agrawal, Aayush Jain, Pushpak Jagtap, and Shishir Kolathaya. Barrier functions inspired reward shaping for reinforcement learning. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 10807–10813, 2024. doi: 10.1109/ICRA57147.2024.10610391.

Pedram Rabiee and Jesse B Hoagg. A closed-form control for safety under input constraints using a composition of control barrier functions. *arXiv preprint arXiv:2406.16874*, 2024.

Pedram Rabiee and Jesse B Hoagg. Soft-minimum and soft-maximum barrier functions for safety with actuation constraints. *Automatica*, 171:111921, 2025.

Pedram Rabiee and Amirsaeid Safari. Safe exploration in reinforcement learning: Training backup control barrier functions with zero training-time safety violations. In *Proceedings of the 7th Annual Learning for Dynamics & Control Conference*, volume 283 of *Proceedings of Machine Learning Research*, pp. 1326–1337, 2025.

Andrew Singletary, Aiden Swann, Ivan Dario Jimenez Rodriguez, and Aaron D Ames. Safe drone flight with time-varying backup controllers. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4577–4584. IEEE, 2022.

Halil Utku Unlu, Vinicius Mariano Gonçalves, Dimitris Chaikalis, Anthony Tzes, and Farshad Khorrami. A control barrier function-based motion planning scheme for a quadruped robot. In *2024 IEEE international Conference on Robotics and Automation (ICRA)*, pp. 12172–12178. IEEE, 2024.

David Van Wijk, Kyle Dunlap, Manoranjan Majji, and Kerianne Hobbs. Safe spacecraft inspection via deep reinforcement learning and discrete control barrier functions. *Journal of Aerospace Information Systems*, 21(12):996–1013, 2024.

David EJ van Wijk, Ersin Daş, Tamas G Molnar, Aaron D Ames, and Joel W Burdick. Safety-critical control with bounded inputs: A closed-form solution for backup control barrier functions. In *2026 American Control Conference (ACC)*, pp. 1215–1222. IEEE, 2026.

Chenggang Wang, Xinyi Wang, Yutong Dong, Lei Song, and Xinping Guan. Multi-constraint safe reinforcement learning via closed-form solution for log-sum-exp approximation of control barrier functions. In *Proceedings of the 7th Annual Learning for Dynamics & Control Conference*, volume 283 of *Proceedings of Machine Learning Research*, pp. 698–710, 2025.

Letian Xu, Hao Liu, Haopeng Zhao, Tianyao Zheng, Tongzhou Jiang, and Lipeng Liu. Autonomous navigation of unmanned vehicle through deep reinforcement learning. In *Proceedings of the 5th International Conference on Artificial Intelligence and Computer Engineering*, pp. 480–484, 2024.

Zifan Xu, Bo Liu, Xuesu Xiao, Anirudh Nair, and Peter Stone. Benchmarking reinforcement learning techniques for autonomous navigation. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 9224–9230, 2023. doi: 10.1109/ICRA48891.2023.10160583.

Lizhi Yang, Blake Werner, Massimiliano de Sa, and Aaron D Ames. Cbf-rl: Safety filtering reinforcement learning in training with control barrier functions. *arXiv preprint arXiv:2510.14959*, 2025.

Qisong Yang, Thiago D. Simão, Simon H. Tindemans, and Matthijs T. J. Spaan. Safety-constrained reinforcement learning with a distributional safety critic. *Machine Learning*, 112:859–887, 2023. doi: 10.1007/s10994-022-06187-8.

Kai S Yun, Zeyang Li, and Navid Azizan. Provably safe, yet scalable reinforcement learning. *arXiv preprint arXiv:2606.14536*, 2026.

A Theoretical Foundations and Proofs

A.1 Direct Sampled bCBF Projection

The bCBF–QP runtime reference is defined in the same augmented surrogate-control coordinates as SAFTER. In particular, it uses the augmented state $z = [x^\top, u^\top]^\top$, surrogate input $\hat{u}$, prescribed surrogate backup $\hat{u}_{\mathrm{b}}$, backup flow φ^z_{b}, lifted current-time margins ψ_i, positive-time backup margins $h^{\mathrm{b}}_i(z, \tau_k)$, terminal margins $h^{\mathrm{b}}_{B,j}(z, T)$, and physical input-admissibility margins $I^{\pm}_d(z)$ introduced in Section 4. Instead of composing these quantities with the hierarchical LSE certificate, the direct baseline enforces their CBF conditions separately:

$$
\begin{aligned}
\hat{u}_t^{\text{bCBF-QP}} = \arg\min_{\hat{u}\in\mathbb{R}^m} \ & \frac{1}{2}\|\hat{u} - \hat{u}_t^{\mathrm{nom}}\|_2^2 \\
\text{s.t.} \quad & L_{\hat{f}}\psi_i(z_t) + L_{\hat{g}}\psi_i(z_t)\hat{u} + \alpha_0(\psi_i(z_t)) \geq 0, && \forall i, \\
& L_{\hat{f}}h^{\mathrm{b}}_i(z_t, \tau_k) + L_{\hat{g}}h^{\mathrm{b}}_i(z_t, \tau_k)\hat{u} + \alpha_{\mathrm{b}}(h^{\mathrm{b}}_i(z_t, \tau_k)) \geq 0, && \forall i, k, \\
& L_{\hat{f}}h^{\mathrm{b}}_{B,j}(z_t, T) + L_{\hat{g}}h^{\mathrm{b}}_{B,j}(z_t, T)\hat{u} + \alpha_B(h^{\mathrm{b}}_{B,j}(z_t, T)) \geq 0, && \forall j, \\
& L_{\hat{f}}I^+_d(z_t) + L_{\hat{g}}I^+_d(z_t)\hat{u} + \alpha_I\big(I^+_d(z_t)\big) \geq 0, \quad d = 1, \ldots, m, \\
& L_{\hat{f}}I^-_d(z_t) + L_{\hat{g}}I^-_d(z_t)\hat{u} + \alpha_I\big(I^-_d(z_t)\big) \geq 0, \quad d = 1, \ldots, m.
\end{aligned}
$$

Here, α_0, α_{b}, α_B, and α_I denote the class-$\mathcal{K}$ functions associated with the respective CBF conditions.

The Lie derivatives of the future and terminal terms include the sensitivity of φ^z_{b} to its initial augmented state, as derived in Appendix A.3.1. At finite policy rate, the experimental bCBF–QP baseline uses the same command-state dynamics, command limits, backup horizon, and terminal set as SAFTER. Its synthesis difference is that the sampled safety conditions remain separate QP constraints, whereas SAFTER first composes them into a single affine condition. Consequently, bCBF–QP and SAFTER use the same command-state augmentation, backup maneuver, horizon samples, terminal set, and physical command limits. Their control-synthesis difference is that bCBF–QP retains one affine constraint per constituent, whereas SAFTER first composes the constituents and projects onto a single composite half-space. Sampling the backup horizon does not, by itself, certify safety between adjacent τ_k. The finite-rate and inter-sample limitations are discussed in Appendix C.3.

A.2 LSE Properties

For the set-based smooth minimum in equation 4, enumerate $\mathcal{S} = \{c_\ell\}_{\ell=1}^M$. Its gradient is

$$
\nabla \operatorname*{softmin}_{\kappa}(\mathcal{S}) = \sum_{\ell=1}^{M} w_\ell \nabla c_\ell, \qquad w_\ell = \frac{e^{-\kappa c_\ell}}{\sum_{r=1}^{M} e^{-\kappa c_r}}, \tag{9}
$$

so $w_\ell > 0$ and $\sum_\ell w_\ell = 1$. Hence every constituent contributes to the gradient of the single composite certificate.

Proof of the LSE bound in equation 4. Let $c_{\min} = \min_\ell c_\ell$. Since

$$
e^{-\kappa c_{\min}} \leq \sum_{\ell=1}^{M} e^{-\kappa c_\ell} \leq M e^{-\kappa c_{\min}},
$$

taking $-\kappa^{-1}\log(\cdot)$ gives the stated bound. Differentiating the same expression gives equation 9. □

A.3 Relative Degree and Backup-Flow Sensitivity

Because h_i depends on x but not on the command state u, $L_{\hat{g}}h_i = 0$ under equation 3. For

$$
\psi_i(z) = L_f h_i(x) + L_g h_i(x) u + k_0 h_i(x),
$$

we have

$$\frac{\partial \psi_i}{\partial u} = L_g h_i(x), \qquad L_{\hat{g}} \psi_i = L_g h_i(x) A.$$

Thus ψ_i has relative degree one with respect to $\hat{u}$ wherever $L_g h_i(x) A \neq 0$. Moreover, $\psi_i = \dot{h}_i + k_0 h_i \geq 0$ together with $h_i(x(0)) \geq 0$ implies $h_i(x(t)) \geq 0$ for the continuous augmented model.

A.3.1 Backup-flow sensitivity

Let

$$\hat{f}_{\mathrm{b}}(z) := \hat{f}(z) + \hat{g}(z)\hat{u}_{\mathrm{b}}(z), \qquad z_{\mathrm{b}}(\tau) = \varphi^z_{\mathrm{b}}(\tau; z_0).$$

The sensitivity

$$J_{\mathrm{b}}(\tau, z_0) := \frac{\partial \varphi^z_{\mathrm{b}}(\tau; z_0)}{\partial z_0}$$

satisfies

$$\dot{J}_{\mathrm{b}} = \frac{\partial \hat{f}_{\mathrm{b}}}{\partial z}(z_{\mathrm{b}}(\tau)) J_{\mathrm{b}}, \qquad J_{\mathrm{b}}(0, z_0) = I_{n+m}.$$

For $E_x = [I_n \; 0_{n\times m}]$,

$$\nabla_{z_0} h^{\mathrm{b}}_i(z_0, \tau) = J_{\mathrm{b}}(\tau, z_0)^\top E_x^\top \nabla_x h_i(E_x z_{\mathrm{b}}(\tau)).$$

The terminal gradients follow by the same chain rule. These sensitivities are what allow the future and terminal margins to contribute to $L_{\hat{g}} h_{\mathrm{aug}}$.

A.4 Closed-Form Projection: KKT Derivation

Consider equation 7 and define only for this derivation

$$\Omega_t := \mathbf{a}(z_t)\hat{u}^{\mathrm{nom}}_t + b(z_t).$$

With multiplier $\mu \geq 0$, the Lagrangian is

$$\mathcal{L}(\hat{u}, \mu) = \frac{1}{2}\|\hat{u} - \hat{u}^{\mathrm{nom}}_t\|_2^2 - \mu\big(\mathbf{a}(z_t)\hat{u} + b(z_t)\big).$$

Stationarity gives $\hat{u} = \hat{u}^{\mathrm{nom}}_t + \mu \mathbf{a}(z_t)^\top$. If $\|\mathbf{a}(z_t)\|_2 > 0$, complementary slackness yields

$$\mu^\star = \frac{\max\{0, -\Omega_t\}}{\|\mathbf{a}(z_t)\|_2^2},$$

which recovers equation 8. If $\mathbf{a}(z_t) = 0$, the surrogate input has no local authority over the certificate: $b(z_t) \geq 0$ makes every $\hat{u}$ feasible, whereas $b(z_t) < 0$ is locally infeasible.

A.5 Continuous Safety Assumptions and Proof

Assumption A.1 (Continuous composite-CBF conditions)**.** On the domain where Theorem 4.2 is invoked: (i) the augmented dynamics, backup-flow terms, terminal functions, and h_{aug} are continuously differentiable; (ii) the augmented model represents the continuous high-level safety model; (iii) the terminal backup set $\mathcal{C}^z_B$ is forward invariant under the prescribed surrogate backup $\hat{u}_{\mathrm{b}}$ and satisfies $E_x(\mathcal{C}^z_B) \subseteq \mathcal{C}_S$; (iv) $h_i(x(0)) \geq 0$ for every primitive barrier and $h_{\mathrm{aug}}(z(0)) \geq 0$; and (v) the composite affine CBF condition is feasible whenever it is enforced.

Proof of Theorem 4.2. The projected surrogate satisfies equation 5, which is the standard CBF condition for h_{aug} under the augmented dynamics. Continuous enforcement therefore makes $\mathcal{C}_{\mathrm{aug}} = \{z : h_{\mathrm{aug}}(z) \geq 0\}$ forward invariant.

Because the LSE smooth minimum under-approximates the ordinary minimum, $h_{\mathrm{aug}} \geq 0$ implies nonnegativity of every normalized constituent, and each fixed positive normalization preserves sign. In particular, $\psi_i = \dot{h}_i + k_0 h_i \geq 0$ and $h_i(x(0)) \geq 0$ imply

$$h_i(x(t)) \geq h_i(x(0)) e^{-k_0 t} \geq 0,$$

so the reduced safety-model state remains in $\mathcal{C}_S$. The positive-time backup margins are nonnegative at every selected τ_k, and the terminal margins place the predicted backup state in $\mathcal{C}_B^z$ at T. By Assumption A.1(iii), continuing the prescribed backup from that terminal state remains in $\mathcal{C}_B^z$ and therefore in the original safe set after projection through E_x. Finally, the input-admissibility constituents remain nonnegative, so the continuous command state remains within its prescribed componentwise bounds. These are the safety and sampled-horizon recoverability consequences stated in the theorem. Inter-sample and finite-rate limitations are separated explicitly in Appendix C.3. □

B Quadruped Navigation Specialization

B.1 Policy Observation, Action, and Safety-Filter Inputs

The high-level actor observation is

$$o_t = \begin{bmatrix} (\bar{p}_{G,t}^B)^\top & \nu_t^\top & (u_{t-1}^{\mathrm{safe}})^\top & \bar{\rho}_t^\top & (u_{t-1}^{\mathrm{nom}})^\top & \iota_{t-1} \end{bmatrix}^\top \in \mathbb{R}^{N_h+12}.$$

Here, $p_{G,t}^B \in \mathbb{R}^2$ is the body-frame goal displacement and $\bar{p}_{G,t}^B = p_{G,t}^B / d_{\mathrm{norm}}$ is its normalized form with $d_{\mathrm{norm}} = 7.5\,\mathrm{m}$ in the implementation. The vector $\nu_t = [v_{x,t}^{\mathrm{meas}}, v_{y,t}^{\mathrm{meas}}, \omega_{z,t}^{\mathrm{meas}}]^\top \in \mathbb{R}^3$ contains measured body-frame translational and yaw velocities for the actor observation; it is not the augmented control-input state used by the safety-filter model. The quantities $u_{t-1}^{\mathrm{safe}}, u_{t-1}^{\mathrm{nom}} \in \mathbb{R}^3$ are the previous executed and nominal physical control inputs, respectively, and $\bar{\rho}_t \in [0,1]^{N_h}$ is the LiDAR scan normalized by the configured maximum range. The flag $\iota_{t-1} \in \{0,1\}$ indicates whether the previous nominal control input is valid for the finite-difference derivative in Appendix C.1. For the 360° configuration, $N_h = 120$ and therefore $o_t \in \mathbb{R}^{132}$. The high-level actor does not observe the full quadruped proprioceptive state because full-body stabilization and low-level tracking are handled by the frozen locomotion policy.

The normalized actor output $\tilde{u}_t^{\mathrm{nom}} \in [-1,1]^3$ is mapped to the physical nominal command according to

$$u_t^{\mathrm{nom}} = c_u + S_u \tilde{u}_t^{\mathrm{nom}}, \qquad c_u = \frac{u_{\max} + u_{\min}}{2}, \qquad S_u = \mathrm{diag}\left(\frac{u_{\max} - u_{\min}}{2}\right).$$

For the experiments, we used $u_{\min} = [-1,-1,-1]^\top$ and $u_{\max} = [2,1,1]^\top$. At policy step t, define $u_t^- := u_{t-1}^{\mathrm{safe}}$ as the applied command state entering the first-order augmentation. Thus the augmented filter state uses $z_t = [x_t^\top, (u_t^-)^\top]^\top$, while the measured velocity ν_t is used only by the actor observation. The policy observation o_t and safety-filter input ξ_t therefore contain related but nonidentical information. In particular, the filter does not use the goal displacement or measured body velocity when evaluating the analytical safety projection; it uses the current LiDAR scan and the command history required by the local augmented model.

B.2 LiDAR Safety Construction

B.2.1 Scan-Fixed Local Geometry

At each high-level safety-filter update, the generic SAFTER plant state x is specialized to the reduced planar navigation state

$$x = \begin{bmatrix} p_x & p_y & \theta \end{bmatrix}^\top,$$

where $p = [p_x, p_y]^\top$ is planar position and θ is yaw. The applied high-level command is $u = [v_x, v_y, \omega_z]^\top$, giving

$$\dot{p} = R(\theta) \begin{bmatrix} v_x \\ v_y \end{bmatrix}, \qquad \dot{\theta} = \omega_z, \qquad \dot{u} = -Au + A\hat{u},$$

with

$$R(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}, \qquad A = \mathrm{diag}(a_x, a_y, a_\omega) \succ 0.$$

Thus the generic augmented state $z = [x^\top, u^\top]^\top$ becomes $z = [p_x, p_y, \theta, v_x, v_y, \omega_z]^\top$ for the navigation certificate.

A temporary scan-fixed frame is initialized at every filter update with $p(0) = 0$ and $\theta(0) = 0$. Each valid LiDAR return is converted into an obstacle-boundary point

$$q_i = p_L^B + \rho_i \begin{bmatrix} \cos\beta_i \\ \sin\beta_i \end{bmatrix},$$

where ρ_i and β_i are the measured range and ray angle, and p_L^B is the LiDAR offset from the robot-body origin. The points q_i are held fixed during one safety-filter computation. As the robot evolves relative to this fixed scan, q_i remains fixed while its body-frame coordinate is the derived quantity

$$\zeta_i(x) = R(\theta)^\top (q_i - p). \tag{10}$$

B.2.2 Elliptical Safety Barrier

The robot footprint and safety clearance are conservatively represented by a body-aligned ellipse with longitudinal and lateral semi-axes r_x and r_y. The semi-axes account for the physical robot dimensions, LiDAR offset, and configured safety clearance; for the verified Go2 configuration, $r_x = 0.58945$ m and $r_y = 0.39445$ m. Define

$$\Pi = \mathrm{diag}(r_x^{-2}, r_y^{-2}).$$

For obstacle point $\zeta_i = [\zeta_{i,x}, \zeta_{i,y}]^\top$ in the instantaneous body frame, the geometric barrier is

$$h_i(p, \theta) = \zeta_i^\top \Pi \zeta_i - 1 = \frac{\zeta_{i,x}^2}{r_x^2} + \frac{\zeta_{i,y}^2}{r_y^2} - 1.$$

Thus, $h_i \geq 0$ requires the observed obstacle point to lie outside the inflated body-aligned ellipse. The elliptical footprint permits yaw to influence the barrier whenever $r_x \neq r_y$. Let

$$J = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}.$$

Differentiating equation 10 gives $\dot{\zeta}_i = -[v_x, v_y]^\top - \omega_z J \zeta_i$ and hence

$$\dot{h}_i = -2\zeta_i^\top \Pi \begin{bmatrix} v_x \\ v_y \end{bmatrix} - 2\omega_z \zeta_i^\top \Pi J \zeta_i.$$

For the directly applied high-level control input $u = [v_x, v_y, \omega_z]^\top$, the corresponding CBF row is

$$L_g h_i = \begin{bmatrix} -2r_x^{-2}\zeta_{i,x} & -2r_y^{-2}\zeta_{i,y} & -2(r_y^{-2} - r_x^{-2})\zeta_{i,x}\zeta_{i,y} \end{bmatrix}.$$

The yaw coefficient is generally nonzero for an anisotropic ellipse and vanishes in the circular special case $r_x = r_y$.

B.2.3 Current-Time HOCBF Condition

Under the augmented control-input dynamics, the primitive barrier h_i has no first-order dependence on the surrogate control input $\hat{u}$. The current-time lifted constituent is therefore

$$\psi_i(z) = \dot{h}_i(z) + k_0 h_i(z), \qquad k_0 > 0.$$

At the beginning of each scan-fixed computation, $p = 0$ and $\theta = 0$, so $\zeta_i = q_i$ in the temporary local frame. To avoid overloading the plant input matrix $g(x)$ or the KKT multiplier μ, define the two point-gradient components

$$\Gamma_{i,x}^{\psi} := -2r_x^{-2}v_x - 2\omega_z(r_y^{-2} - r_x^{-2})\zeta_{i,y} + 2k_0 r_x^{-2}\zeta_{i,x},$$
$$\Gamma_{i,y}^{\psi} := -2r_y^{-2}v_y - 2\omega_z(r_y^{-2} - r_x^{-2})\zeta_{i,x} + 2k_0 r_y^{-2}\zeta_{i,y}.$$

The complete gradient with respect to $z = [p_x, p_y, \theta, v_x, v_y, \omega_z]^\top$ is

$$\nabla_z \psi_i = \begin{bmatrix} -\Gamma_{i,x}^{\psi} \\ -\Gamma_{i,y}^{\psi} \\ \Gamma_{i,x}^{\psi}\zeta_{i,y} - \Gamma_{i,y}^{\psi}\zeta_{i,x} \\ -2r_x^{-2}\zeta_{i,x} \\ -2r_y^{-2}\zeta_{i,y} \\ -2(r_y^{-2} - r_x^{-2})\zeta_{i,x}\zeta_{i,y} \end{bmatrix}.$$

Its last three entries give $\partial\psi_i/\partial u = L_g h_i$, so multiplication by the control-state input block A recovers $L_{\hat{g}}\psi_i = L_g h_i A$ as stated in Section 4. The first three entries account for the dependence of the scan-fixed point coordinate on robot translation and yaw. Together, these expressions provide the code-aligned current-time gradient used in the hierarchical certificate.

B.2.4 Backup Rollout and Positive-Time Barrier

The quadruped specialization uses the zero surrogate backup $\hat{u}_{\mathrm{b}} = 0$. Along this backup flow, the applied control-input state decays as

$$u(\tau) = e^{-A\tau} u_0.$$

The corresponding yaw trajectory and planar position are

$$\theta_{\mathrm{b}}(\tau) = \theta_0 + \omega_{z,0}\frac{1 - e^{-a_\omega \tau}}{a_\omega},$$

$$p_{\mathrm{b}}(\tau) = p_0 + \int_0^\tau R(\theta_{\mathrm{b}}(\sigma)) \begin{bmatrix} v_{x,0} e^{-a_x \sigma} \\ v_{y,0} e^{-a_y \sigma} \end{bmatrix} d\sigma.$$

These equations define the exact continuous augmented backup flow $z_{\mathrm{b}}(\tau) = \varphi^z_{\mathrm{b}}(\tau; z_0)$ used in the theoretical construction. The implementation evaluates the planar integral with fixed eight-point Gauss–Legendre quadrature and differentiates that fixed quadrature map analytically. Thus the implemented rollout is a smooth numerical approximation of the exact flow; it does not use sequential Euler integration or finite-difference flow sensitivities. All requested horizon states and their implementation-level sensitivities are therefore evaluated as smooth functions of the same initial augmented state.

The implementation used a total of 21 rollout nodes, including the current-time node at $\tau_0 = 0$, along the trajectory generated by the backup controller. Because the raw zero-time geometric barrier is replaced by the lifted constituent ψ_i, the LSE receives $K = 20$ positive-time samples,

$$\tau_k = \frac{kT}{20} = 0.2k\,\mathrm{s}, \qquad k = 1, \ldots, 20, \qquad T = 4.0\,\mathrm{s}.$$

For each positive-time sample, the backup safety value is

$$h_i^{\mathrm{b}}(z_0, \tau_k) = h_i(p_{\mathrm{b}}(\tau_k), \theta_{\mathrm{b}}(\tau_k))\,.$$

Let

$$J_{\mathrm{b}}(\tau, z_0) = \frac{\partial \varphi^z_{\mathrm{b}}(\tau; z_0)}{\partial z_0}$$

denote the augmented backup-flow sensitivity. At a future state, the primitive barrier gradient is

$$\nabla_z h_i = \begin{bmatrix} -\left(R(\theta)(2\Pi\zeta_i)\right)_1 \\ -\left(R(\theta)(2\Pi\zeta_i)\right)_2 \\ (2\Pi\zeta_i)_x \zeta_{i,y} - (2\Pi\zeta_i)_y \zeta_{i,x} \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

The flow-composed gradient with respect to the current augmented state is

$$\nabla_{z_0} h_i^{\mathrm{b}}(z_0, \tau_k) = J_{\mathrm{b}}(\tau_k, z_0)^\top \nabla_z h_i(z_{\mathrm{b}}(\tau_k))\,.$$

This chain rule is the quantity used when the positive-time barriers contribute to the LSE gradient and its Lie derivatives.

B.2.5 Forward-Invariant Elliptical Terminal Backup Set

The terminal construction uses a fixed enlarged ellipse together with a separate residual-motion condition, rather than embedding a smoothed stopping distance directly into the ellipse semi-axes. Define

$$R_x = r_x + d_{\mathrm{b}}, \qquad R_y = r_y + d_{\mathrm{b}}, \qquad S_B = \mathrm{diag}(R_x, R_y),$$

where $d_{\mathrm{b}} > 0$ is the additional terminal margin. For terminal obstacle coordinate $\zeta_i = [\zeta_{i,x}, \zeta_{i,y}]^\top$, define

$$E_i = \zeta_i^\top S_B^{-2} \zeta_i = \frac{\zeta_{i,x}^2}{R_x^2} + \frac{\zeta_{i,y}^2}{R_y^2}, \qquad \ell_i = \frac{1}{2} \log E_i.$$

Thus, $\ell_i \geq 0$ is equivalent to $E_i \geq 1$ and selects the exterior of the fixed terminal ellipse. The logarithmic coordinate is dimensionless and is well defined on the evaluation domain $E_i > 0$.

To quantify residual motion under the zero-surrogate backup, define

$$c_\theta = \frac{1}{2} \left| \frac{R_x}{R_y} - \frac{R_y}{R_x} \right|, \qquad V_{\mathrm{b}} = \frac{v_x^2}{R_x^2} + \frac{v_y^2}{R_y^2} + c_\theta^2 \omega_z^2,$$

and let $\lambda_A := \min(a_x, a_y, a_\omega)$ denote the lowest value of the diagonal matrix A. The quantity V_{b} has units of inverse time squared and measures normalized residual translation together with a bound on the yaw contribution induced by ellipse anisotropy. Under $\hat{u}_{\mathrm{b}} = 0$, the components satisfy $\dot{v}_x = -a_x v_x$, $\dot{v}_y = -a_y v_y$, and $\dot{\omega}_z = -a_\omega \omega_z$. Consequently,

$$\dot{V}_{\mathrm{b}} \leq -2\lambda_A V_{\mathrm{b}}. \tag{11}$$

For each valid LiDAR point, we define two terminal backup-set constituents: $h_{B,i}^{\mathrm{geom}}$ for geometric clearance and $h_{B,i}^{\mathrm{rec}}$ for recoverability under the residual commanded motion.

$$h_{B,i}^{\mathrm{geom}} := \ell_i, \qquad h_{B,i}^{\mathrm{rec}} := \ell_i^2 - \frac{2V_{\mathrm{b}}}{\lambda_A^2}.$$

The per-ray set is $\mathcal{C}_{B,i}^z := \{z : h_{B,i}^{\mathrm{geom}}(z) \geq 0,\ h_{B,i}^{\mathrm{rec}}(z) \geq 0\}$ and the complete augmented terminal set is $\mathcal{C}_B^z := \bigcap_i \mathcal{C}_{B,i}^z$. The generic terminal index j in Section 4 therefore ranges over these two terminal constituents for all LiDAR rays. Neither constituent is claimed to be forward invariant independently; the invariant object is their intersection. Equivalently, inside the geometric branch the recoverability condition requires $\ell_i \geq \sqrt{2V_{\mathrm{b}}}/\lambda_A$. At zero residual motion, the set reduces to the exterior of the fixed enlarged terminal ellipse.

Proposition B.1 (Forward invariance of the terminal backup set)**.** *For the fixed-scan local model in Appendix B.2.1, $\mathcal{C}_B^z$ is forward invariant under the continuous zero-surrogate backup controller $\hat{u}_{\mathrm{b}} = 0$. Moreover, $E_x(\mathcal{C}_B^z) \subseteq \mathcal{C}_S$ because $R_x > r_x$ and $R_y > r_y$.*

Proof. Let

$$w_i = S_B^{-1} \zeta_i, \qquad \bar{v}_B = S_B^{-1} \begin{bmatrix} v_x \\ v_y \end{bmatrix}, \qquad M_B = S_B^{-1} J S_B.$$

Because $E_i = \|w_i\|_2^2$, we have $\ell_i = \log \|w_i\|_2$. Using $\dot{\zeta}_i = -[v_x, v_y]^\top - \omega_z J \zeta_i$ gives $\dot{w}_i = -\bar{v}_B - \omega_z M_B w_i$. The symmetric part of M_B has spectral norm c_θ, so $|w_i^\top M_B w_i| \leq c_\theta \|w_i\|_2^2$. Therefore, on the geometric branch $\ell_i \geq 0$, where $\|w_i\|_2 \geq 1$,

$$\dot{\ell}_i = -\frac{w_i^\top \bar{v}_B}{\|w_i\|_2^2} - \omega_z \frac{w_i^\top M_B w_i}{\|w_i\|_2^2} \tag{12}$$

$$\geq -\|\bar{v}_B\|_2 - c_\theta |\omega_z| \tag{13}$$

$$\geq -\sqrt{2V_{\mathrm{b}}}. \tag{14}$$

Consider a boundary point with $h_{B,i}^{\mathrm{rec}} = 0$ and $h_{B,i}^{\mathrm{geom}} \geq 0$. Then $\ell_i = \sqrt{2V_{\mathrm{b}}}/\lambda_A$, and equation 11– equation 14 give

$$\begin{aligned} \dot{h}_{B,i}^{\mathrm{rec}} &= 2\ell_i \dot{\ell}_i - \frac{2}{\lambda_A^2} \dot{V}_{\mathrm{b}} \\ &\geq -2\ell_i \sqrt{2V_{\mathrm{b}}} + \frac{4V_{\mathrm{b}}}{\lambda_A} = 0. \end{aligned}$$

Hence, the backup vector field does not point outward through the recoverability boundary. If instead $h_{B,i}^{\mathrm{geom}} = \ell_i = 0$ while $h_{B,i}^{\mathrm{rec}} \geq 0$, the recoverability condition implies $V_{\mathrm{b}} = 0$ and therefore $v_x = v_y = \omega_z = 0$. Consequently $\dot{\ell}_i = 0$, so the vector field is tangent to the geometric boundary. The

Nagumo boundary condition holds on every active boundary of $\mathcal{C}^z_{B,i}$, and the intersection over all valid fixed-scan LiDAR points is forward invariant under the same backup vector field. Finally, the terminal ellipse strictly contains the ordinary safety ellipse because $R_x = r_x + d_\mathrm{b} > r_x$ and $R_y = r_y + d_\mathrm{b} > r_y$; being outside the terminal ellipse therefore implies being outside the ordinary ellipse, yielding $E_x(\mathcal{C}^z_B) \subseteq \mathcal{C}_S$. □

For the experimental geometry, $r_x = 0.58945\,\mathrm{m}$, $r_y = 0.39445\,\mathrm{m}$, and $d_\mathrm{b} = 0.60\,\mathrm{m}$. Therefore $R_x = 1.18945\,\mathrm{m}$, $R_y = 0.99445\,\mathrm{m}$, and $c_\theta = 0.180015$ to the reported precision. With $A = \mathrm{diag}(8, 8, 8)\,\mathrm{s}^{-1}$, the residual-motion decay rate is $\lambda_A = 8\,\mathrm{s}^{-1}$. These values match the implementation summary in Appendix B.2.7. Proposition B.1 verifies the terminal-set condition in Assumption A.1(iii) for this specialization.

For implementation, define $\Pi_B := S_B^{-2}$, so $E_i = \zeta_i^\top \Pi_B \zeta_i$. The terminal-state gradient of the logarithmic clearance can then be written without introducing an additional helper symbol as

$$\nabla_z \ell_i = \begin{bmatrix} -\left(R(\theta)\dfrac{\Pi_B \zeta_i}{E_i}\right)_1 \\ -\left(R(\theta)\dfrac{\Pi_B \zeta_i}{E_i}\right)_2 \\ \left(\dfrac{\Pi_B \zeta_i}{E_i}\right)_x \zeta_{i,y} - \left(\dfrac{\Pi_B \zeta_i}{E_i}\right)_y \zeta_{i,x} \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

The residual-motion gradient is

$$\nabla_z V_\mathrm{b} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 2v_x/R_x^2 \\ 2v_y/R_y^2 \\ 2c_\theta^2 \omega_z \end{bmatrix}.$$

Hence $\nabla_z h^{\mathrm{geom}}_{B,i} = \nabla_z \ell_i$ and

$$\nabla_z h^{\mathrm{rec}}_{B,i} = 2\ell_i \nabla_z \ell_i - \frac{2}{\lambda_A^2} \nabla_z V_\mathrm{b}.$$

The gradients with respect to the current augmented state are obtained through the same terminal backup-flow sensitivity used for the rollout constituents,

$$\nabla_{z_0} h^{(\cdot)}_{B,i}(z_0, T) = J_\mathrm{b}(T, z_0)^\top \nabla_z h^{(\cdot)}_{B,i}(z_\mathrm{b}(T)).$$

This preserves one common sensitivity pipeline for both terminal constituents and introduces no additional finite-difference approximation.

B.2.6 Implementation-Specific LSE Scaling

The implementation applies only fixed, code-defined scaling before hierarchical LSE aggregation. The current-time HOCBF constituent has units of inverse time and is normalized as

$$\bar{\psi}_i = \frac{\psi_i}{s_\psi}, \qquad s_\psi = 1.0\,\mathrm{s}^{-1}.$$

The positive-time geometric barriers $h^\mathrm{b}_i(z, \tau_k)$ are dimensionless and are inserted directly, so $\bar{h}^\mathrm{b}_i = h^\mathrm{b}_i$ with no separate rollout-scale hyperparameter. The two terminal constituents are likewise dimensionless and use the fixed scales $s_{B,\mathrm{geom}} = s_{B,\mathrm{rec}} = 1$, so their numerical values are unchanged by normalization. Thus, no independent tunable rollout or terminal normalization is hidden from the parameter table.

The control-input margins are made dimensionless using fixed positive per-axis reference scales. Each control-input component is defined as $s_{I,d} := \max(|u_{\min,d}|, |u_{\max,d}|) > 0$. The normalized input constituents are

$$\bar{I}_d^+ = \frac{u_{\max,d} - u_d}{s_{I,d}}, \qquad \bar{I}_d^- = \frac{u_d - u_{\min,d}}{s_{I,d}}.$$

The input bounds used in this work are $u_{\min} = [-1, -1, -1]^\top$ and $u_{\max} = [2, 1, 1]^\top$, which gives $s_I = [2, 1, 1]^\top$. Accordingly, the longitudinal, lateral, and yaw-rate margins are normalized by $2\ \mathrm{m\,s^{-1}}$, $1\ \mathrm{m\,s^{-1}}$, and $1\ \mathrm{rad\,s^{-1}}$, respectively. Because each scale is strictly positive, the normalization preserves the sign of the corresponding input-admissibility constituent while only affecting its numerical weighting within the LSE aggregation.

B.2.7 Safety-Filter and Implementation Parameters

Table 5 summarizes the principal parameters used by the high-level safety-filter configuration. The high-level policy and safety filter operate at 10 Hz, while the frozen low-level locomotion controller operates at 50 Hz. The backup flow uses the zero-surrogate controller with the same first-order control-input dynamics as the safety filter and evaluates its planar displacement with fixed eight-point Gauss–Legendre quadrature. The rollout-count notation distinguishes the 21 total nodes stored by the implementation parameter from the 20 strictly positive-time samples that actually enter the LSE certificate.

C Finite-Rate Realization and Guarantee Scope

C.1 Nominal Surrogate Tracking

Let $u^{\mathrm{nom}}(t)$ denote the nominal physical command. Substituting equation 6 into the augmented command dynamics gives

$$\dot{u} = \dot{u}^{\mathrm{nom}} + \Sigma(u^{\mathrm{nom}} - u), \qquad \dot{e}_u = -\Sigma e_u, \quad e_u := u - u^{\mathrm{nom}}.$$

Hence the applied command exponentially tracks the nominal command whenever the safety projection is inactive. In the 10 Hz implementation,

$$\dot{u}_t^{\mathrm{nom}} \approx \iota_{t-1} \frac{u_t^{\mathrm{nom}} - u_{t-1}^{\mathrm{nom}}}{\Delta t},$$

where $\iota_{t-1} = 0$ when no valid previous nominal command is available and 1 otherwise.

C.2 Finite-Rate Experimental Realization

The reported experiments evaluate the safety filter at the high-level control interval $\Delta t = 0.1\,\mathrm{s}$. The analytical projection in equation 8 is first applied in surrogate-control space to obtain $\hat{u}_t^\star$. During each high-level interval, $\hat{u}_t^\star$ is held constant under the first-order command dynamics

$$\dot{u} = -Au + A\hat{u}_t^\star.$$

Because A is diagonal, the corresponding command-state update is evaluated exactly as

$$u_t^{\mathrm{pre}} = \Lambda u_t^- + (I - \Lambda)\hat{u}_t^\star, \qquad \Lambda := e^{-A\Delta t}. \tag{15}$$

The physical command executed by the environment is then obtained through the component-wise command limits,

$$u_t^{\mathrm{safe}} = \mathrm{clip}\left(u_t^{\mathrm{pre}}, u_{\min}, u_{\max}\right). \tag{16}$$

At the next filter evaluation, this executed command is used as the current control-input state of the augmented model.

Thus, the reported implementation uses the single-half-space analytical projection of equation 8 followed by the exact finite-rate propagation in equation 15 and physical command clipping in equation 16. The same command limits are included in the composite safety certificate. In the finite-rate implementation, a final component-wise clipping operation also enforces these limits on the executed command. The continuous-time guarantee of Theorem 4.2 therefore remains scoped to the continuous augmented model.

Table 5: Safety-filter and local-navigation parameters used by the verified implementation.

Parameter	Symbol	Value
Control and sensing		
LiDAR update rate	–	$10\,\mathrm{Hz}$
Low-level control rate	–	$50\,\mathrm{Hz}$
LiDAR field of view	–	360°
Number of LiDAR rays	N_h	120
LiDAR range limits	$\rho_{\min}, \rho_{\max}$	$0.05,\ 5.0\,\mathrm{m}$
LiDAR body offset	p_L^B	$[0.28945, 0]^\top\,\mathrm{m}$
Robot length	L	$0.70\,\mathrm{m}$
Robot width	w_{rob}	$0.31\,\mathrm{m}$
Configured safety distance	d_{safe}	$0.30\,\mathrm{m}$
Effective footprint clearance	d_{eff}	$0.23945\,\mathrm{m}$
Longitudinal CBF semi-axis	r_x	$0.58945\,\mathrm{m}$
Lateral CBF semi-axis	r_y	$0.39445\,\mathrm{m}$
Control-input augmentation		
Control-dynamics gain	A	$\mathrm{diag}(8,8,8)\,\mathrm{s}^{-1}$
Nominal tracking gain	Σ	$\mathrm{diag}(8,8,8)\,\mathrm{s}^{-1}$
Current HOCBF gain	k_0	$2.0\,\mathrm{s}^{-1}$
Composite CBF gain	α_{aug}	$0.8\,\mathrm{s}^{-1}$
Longitudinal speed command limits	v_x	$[-1.0, 2.0]\,\mathrm{m\,s}^{-1}$
Lateral speed command limits	v_y	$[-1.0, 1.0]\,\mathrm{m\,s}^{-1}$
Yaw-rate command limits	ω_z	$[-1.0, 1.0]\,\mathrm{rad\,s}^{-1}$
Backup prediction and terminal set		
Backup surrogate input	$\hat{u}_{\mathrm{b}}$	0
Backup horizon	T	$4.0\,\mathrm{s}$
Total rollout nodes	N_{b}	21 (includes $\tau_0 = 0$)
Positive-time LSE samples	K	20
Positive sample times	τ_k	$0.2k\,\mathrm{s},\ k = 1{:}20$
Fixed terminal margin	d_{b}	$0.60\,\mathrm{m}$
Terminal longitudinal semi-axis	R_x	$1.18945\,\mathrm{m}$
Terminal lateral semi-axis	R_y	$0.99445\,\mathrm{m}$
Yaw anisotropy coefficient	c_θ	0.180015
Residual-motion decay rate	λ_A	$8.0\,\mathrm{s}^{-1}$
LSE aggregation		
Recoverability LSE sharpness	κ_B	50
Input LSE sharpness	κ_U	20
Outer LSE sharpness	κ_O	20
HOCBF normalization scale	s_ψ	$1.0\,\mathrm{s}^{-1}$
Terminal geometric scale	$s_{B,\mathrm{geom}}$	1.0
Terminal recoverability scale	$s_{B,\mathrm{rec}}$	1.0

C.3 Guarantee Scope

Theorem 4.2 is a continuous-time result for the reduced safety model. Two sampling gaps remain outside that theorem: (i) the backup trajectory is checked only at the selected τ_k, and (ii) the safety filter is recomputed only at the $10\,\mathrm{Hz}$ high-level updates. In the reported finite-rate implementation, the same physical command limits included in the composite certificate are also enforced by a final component-wise clipping operation. This ensures that the executed command remains within its prescribed bounds, but the continuous-time theorem does not guarantee that the clipped command satisfies the composite CBF condition between high-level updates.

The LiDAR certificate is also local to the current scan. During one filter computation, observed endpoints are fixed while the reduced pose and command states evolve relative to them. The result therefore assumes that this local model and the zero-surrogate backup adequately represent the high-level motion over the prediction horizon. It does not certify full-body tracking error, unobserved or occluded geometry, moving obstacles absent from the model, or arbitrary LiDAR noise. Conse-

quently, the formal guarantee applies to the stated reduced safety model and measured scan-defined obstacles, not to every full quadruped state.

Assumption C.1 (Finite-rate experimental realization)**.** Over each high-level interval, the reduced pose and first-order command dynamics are assumed to remain representative of the high-level closed-loop motion, and the frozen 50 Hz locomotion controller is assumed to track the resulting high-level command sufficiently well for that reduced model to remain applicable.

D SAFTER–SAC TRAINING DETAILS

D.1 CRITIC AND ACTOR UPDATES

The actor uses a reparameterized tanh-Gaussian policy in normalized action coordinates $\tilde{u}_t^{\mathrm{nom}} \in [-1,1]^m$. The corresponding physical nominal command is $u_t^{\mathrm{nom}} = c_u + S_u \tilde{u}_t^{\mathrm{nom}}$ as defined in Appendix B.1. The complete finite-rate safety map consists of the nominal-surrogate construction, the analytical half-space projection in equation 8, the finite-rate command update in equation 15, and the final component-wise clipping in equation 16. It produces $u_t^{\mathrm{safe}} = \mathcal{F}_\eta(\xi_t, u_t^{\mathrm{nom}})$, which is mapped back to normalized critic coordinates as $\tilde{u}_t^{\mathrm{safe}} = S_u^{-1}(u_t^{\mathrm{safe}} - c_u)$.

For replay sample $(o_t, \tilde{u}_t^{\mathrm{safe}}, r_t, o_{t+1}, \chi_t)$, where χ_t is the true-termination indicator, the next nominal action is filtered with the same forward map without gradient propagation to obtain $\tilde{u}_{t+1}^{\mathrm{safe}}$. With twin critics $Q_{\vartheta_1}, Q_{\vartheta_2}$ and target networks $Q_{\bar{\vartheta}_1}, Q_{\bar{\vartheta}_2}$, the SAC target is

$$\mathcal{Y}_t = r_t + \gamma(1-\chi_t)\left[\min_{i=1,2} Q_{\bar{\vartheta}_i}(o_{t+1}, \tilde{u}_{t+1}^{\mathrm{safe}}) - \alpha_{\mathrm{ent}} \log \pi_\phi(\tilde{u}_{t+1}^{\mathrm{nom}} \mid o_{t+1})\right],$$

and the critics minimize

$$J_Q = \mathbb{E}_{\mathcal{D}}\left[\sum_{i=1}^{2}\left(Q_{\vartheta_i}(o_t, \tilde{u}_t^{\mathrm{safe}}) - \mathcal{Y}_t\right)^2\right].$$

Thus the critics evaluate the action that actually generated the transition.

For the actor update, the critic parameters are fixed and

$$J_\pi(\phi) = \mathbb{E}\left[\alpha_{\mathrm{ent}} \log \pi_\phi(\tilde{u}_t^{\mathrm{nom}} \mid o_t) - \min_{i=1,2} Q_{\vartheta_i}(o_t, \tilde{u}_t^{\mathrm{safe}})\right].$$

Gradients propagate through

$$\tilde{u}_t^{\mathrm{nom}} \mapsto u_t^{\mathrm{nom}} \mapsto \hat{u}_t^{\mathrm{nom}} \mapsto \hat{u}_t^{\star} \mapsto u_t^{\mathrm{pre}} \mapsto u_t^{\mathrm{safe}} \mapsto \tilde{u}_t^{\mathrm{safe}},$$

where $\hat{u}_t^{\star}$ is the analytical half-space projection and u_t^{pre} is the finite-rate command before clipping.

Holding the current filter state and command history fixed, define the local projection Jacobian

$$J_{\mathrm{proj},t} := \frac{\partial \hat{u}_t^{\star}}{\partial \hat{u}_t^{\mathrm{nom}}}.$$

Away from the switching boundary,

$$J_{\mathrm{proj},t} = \begin{cases} I_m, & \Omega_t \geq 0, \\ I_m - \dfrac{a_t^\top a_t}{a_t a_t^\top}, & \Omega_t < 0. \end{cases}$$

Thus, the analytical projection removes the component of the actor gradient normal to the active safety half-space.

For the implemented finite-rate map, the corresponding local physical-action Jacobian is

$$J_{F,t} = D_{\mathrm{clip},t} B_\Delta J_{\mathrm{proj},t} A^{-1}\left(\frac{\iota_{t-1}}{\Delta t} I_m + \Sigma\right), \tag{17}$$

where $B_\Delta = I - e^{-A\Delta t}$ and $D_{\mathrm{clip},t}$ denotes the local derivative of the component-wise clipping operation. The surrounding mappings between normalized and physical action coordinates are handled by automatic differentiation.

The Identity-Jacobian ablation preserves the same numerical forward action but replaces only $J_{\mathrm{proj},t}$ by I_m during the actor backward pass. All other mappings in the safety-filter chain retain their ordinary derivatives.

D.2 Entropy Temperature and Critic Parameterization

Entropy regularization remains defined on the nominal stochastic policy, because its tanh-Gaussian log-density is tractable whereas the deterministic filtered action does not generally have a tractable induced density. The implementation uses automatic entropy tuning with $\vartheta_\alpha := \log \alpha_{\text{ent}}$ and

$$J_\alpha(\vartheta_\alpha) = \mathbb{E}\big[-\vartheta_\alpha\big(\log \pi_\phi(\tilde{u}_t^{\text{nom}} \mid o_t) + \mathcal{H}_{\text{target}}\big)\big] ,$$

using $\mathcal{H}_{\text{target}} = -3$ for the three-dimensional command. The log-probability is detached in this temperature update.

The critics are parameterized as $Q_\vartheta(o_t, \tilde{u}_t^{\text{safe}})$. Because the correction reward depends on both the nominal and projected surrogate commands, distinct nominal actions can produce the same executed action while incurring different correction costs. Moreover, nominal-command history is included in the subsequent observation. The critic can therefore alias such nominal-action choices and should be viewed as a practical observation-conditioned approximation rather than an exact Markov action-value representation. Accordingly, "exact" in the Jacobian discussion refers to the local analytical projection derivative, not to an exact policy gradient for the complete objective.

D.3 Time-Limit Truncation

The Bellman mask uses $\chi_t = 1$ for true task termination and $\chi_t = 0$ for time-limit truncation. For a truncation, replay stores the final observation before reset as o_{t+1}, so the SAC target continues to bootstrap through the artificial time limit.

D.4 High-Level Reward Function

Table 6: High-level SAC reward coefficients used for all reported trained checkpoints. The reward is evaluated at 10 Hz. The safety-filter correction coefficient is set to zero for Vanilla SAC and to -0.05 for the safety-filtered methods.

Reward component	Symbol	Coefficient
Goal progress	w_{prog}	$+4.00$
Goal reached	w_{goal}	$+30.00$
Time penalty	w_{time}	-0.01
Timeout penalty	w_{timeout}	-5.00
Forward alignment	w_{align}	$+0.20$
Reverse motion	w_{rev}	-0.50
Yaw tracking	w_{yaw}	-0.05
Command smoothness	w_{smooth}	-0.01
Safety-filter correction	w_{corr}	$-0.05^\dagger$
Collision / unsafe terminal event	w_{coll}	-20.00

† Set to 0 for Vanilla SAC.

The high-level navigation reward is evaluated once per policy step at 10 Hz ($\Delta t = 0.1$ s). All SAC-based methods use the same task-reward formulation and coefficients, except that the safety-filter correction term is disabled for Vanilla SAC because no safety filter modifies its action.

Let $\mathbf{p}_t \in \mathbb{R}^2$ denote the planar robot position, $\mathbf{p}_G \in \mathbb{R}^2$ the goal position, and let $d_t := \|\mathbf{p}_G - \mathbf{p}_t\|_2$ denote the current goal distance. The total high-level reward is

$$\begin{aligned} r_t = {} & w_{\text{prog}} r_t^{\text{prog}} + w_{\text{goal}} r_t^{\text{goal}} + w_{\text{time}} r_t^{\text{time}} + w_{\text{timeout}} r_t^{\text{timeout}} + w_{\text{align}} r_t^{\text{align}} + w_{\text{rev}} r_t^{\text{rev}} \\ & + w_{\text{yaw}} r_t^{\text{yaw}} + w_{\text{smooth}} r_t^{\text{smooth}} + w_{\text{corr}} r_t^{\text{corr}} + w_{\text{coll}} r_t^{\text{coll}}, \end{aligned}$$

where the signed coefficients are listed in Table 6.

Goal progress and completion. Progress toward the goal is measured using the change in Euclidean goal distance, defined by $r_t^{\text{prog}} = d_{t-1} - d_t$. Additionally, the goal-completion term is defined by $r_t^{\text{goal}} = \mathbb{I}[d_t \le d_{\text{goal}}]$, where $d_{\text{goal}} = 0.35$ m.

Time and timeout penalties. A constant per-step cost encourages shorter trajectories, denoted by $r_t^{\text{time}} = 1$. Let $\mathcal{T}_t$ denote a time-limit truncation that occurs without a preceding goal, collision, or fall termination. The timeout term is defined as $r_t^{\text{timeout}} = \mathbb{I}[\mathcal{T}_t]$. As described in Appendix D.3, a timeout is treated as a truncation rather than a true Bellman terminal state.

Forward alignment and reverse motion. Let β_t denote the signed bearing of the goal in the robot body frame, and let $v_{x,t}^{\text{meas}}$ denote the measured body-frame forward velocity. The forward-alignment term is

$$r_t^{\text{align}} = \frac{\text{ReLU}(v_{x,t}^{\text{meas}})}{v_{x,\max}} \cos \beta_t.$$

Motion in the reverse direction is penalized using

$$r_t^{\text{rev}} = \left(\frac{\text{ReLU}(-v_{x,t}^{\text{meas}})}{v_{x,\max}} \right)^2.$$

Yaw tracking. A reference yaw rate is constructed from the current goal-bearing error, with the reward function defined by $\omega_t^{\text{ref}} = \text{clip}\left(k_\beta \beta_t, -\omega_{z,\max}, \omega_{z,\max}\right)$, where $k_\beta = 1.5$ is the parameter used for the implementation. With measured body yaw rate $\omega_{z,t}^{\text{meas}}$, the normalized yaw-tracking cost is

$$r_t^{\text{yaw}} = \left(\frac{\omega_{z,t}^{\text{meas}} - \omega_t^{\text{ref}}}{\omega_{z,\max}} \right)^2.$$

Command smoothness. Let u_t^{safe} denote the executed high-level command after safety filtering. Rapid changes in the executed command are penalized according to

$$r_t^{\text{smooth}} = \left\| u_t^{\text{safe}} - u_{t-1}^{\text{safe}} \right\|_2^2.$$

Safety-filter intervention. The intervention term discourages the nominal policy from repeatedly requesting commands that require large safety corrections. Let $s_u > 0$ denote the component-wise action scaling used by the implementation, let $\oslash$ denote element-wise division, and let $\hat{u}_t^{\text{filt}}$ denote the surrogate command returned by the active safety filter. For SAFTER, $\hat{u}_t^{\text{filt}} = \hat{u}_t^\star$ from equation 8. For all safety-filtered SAC variants, the correction cost is

$$r_t^{\text{corr}} = \left\| \left(\hat{u}_t^{\text{filt}} - \hat{u}_t^{\text{nom}} \right) \oslash s_u \right\|_2^2.$$

The same coefficient $w_{\text{corr}} = -0.05$ is used for all safety-filtered SAC methods. For Vanilla SAC, no safety-filter intervention occurs and $w_{\text{corr}} = 0$.

Unsafe termination. Let $\mathcal{C}_t$ denote the configured collision-reward event. The corresponding terminal penalty is defined as $r_t^{\text{coll}} = \mathbb{I}[\mathcal{C}_t]$. The implementation does not include a separate fall-reward term. A fall remains an episode-termination condition and may be included in the configured unsafe terminal event used by r_t^{coll}, but it receives no additional independent fall penalty.

D.5 SAC Network Architecture and Optimization

All SAC-based methods implemented in our framework use the same actor–critic architecture and optimization settings summarized in Table 7. The actor is a reparameterized tanh-squashed diagonal Gaussian policy with three hidden layers of 512, 256, and 128 units using ELU activations. Its output contains three action means and three log-standard deviations, with the latter clamped to $[-5, 2]$. The critic uses twin Q-networks with the same hidden-layer dimensions and ELU activations, together with separate Polyak-averaged target networks.

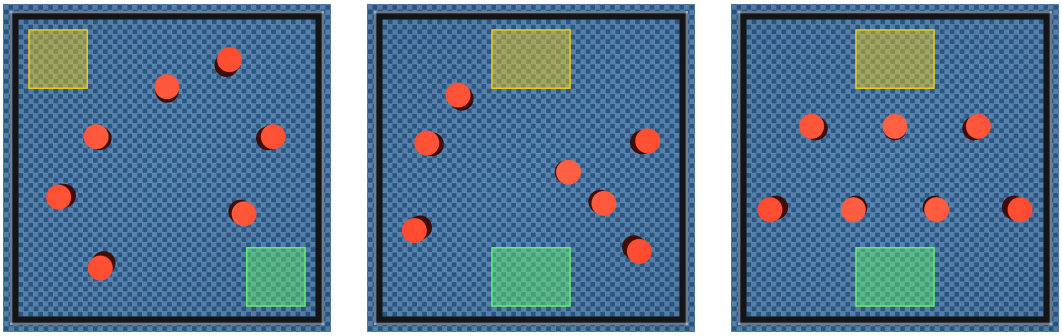
E Experimental Protocols and Additional Results

E.1 Training and Evaluation Maps

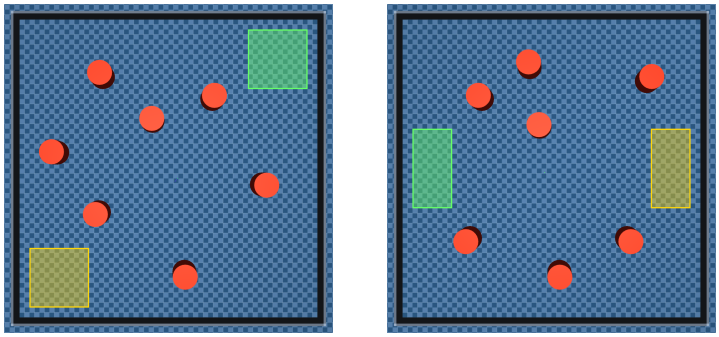
The quadruped experiments use a fixed $15\ \text{m} \times 15\ \text{m}$ workspace with cylindrical obstacles of radius $0.6\ \text{m}$. The training bank contains the five hand-crafted maps shown in Figure 4, with 200 predefined

Table 7: Optimization parameters used by the SAC-based methods.

Parameter	Value
Discount factor γ	0.99
Polyak coefficient τ	0.005
Actor learning rate	3×10^{-4}
Critic learning rate	3×10^{-4}
Temperature learning rate	3×10^{-4}
Optimizer	Adam
Replay-buffer capacity	2×10^{6} transitions
Minibatch size	1024
Learning starts	20,000 environment transitions
Updates per environment step	1
Target entropy	-3



(a) Training Map 1 (b) Training Map 2 (c) Training Map 3

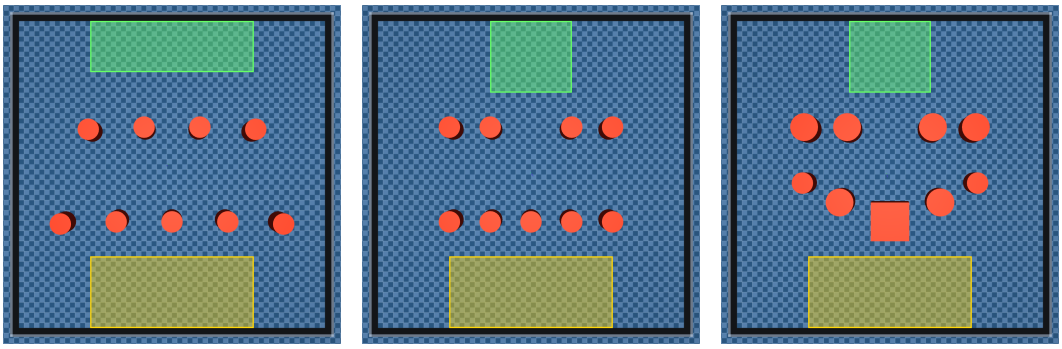


(d) Training Map 4 (e) Training Map 5

Figure 4: Five hand-crafted training maps with fixed obstacle configurations and predefined initial and goal regions. Green and yellow regions indicate the initial and goal regions, respectively, while obstacles are shown in red.

initial–goal scenarios for each map. These maps define a structured training distribution with varied start–goal orientations and obstacle arrangements. Because the policy receives only a local planar LiDAR observation with a 5 m maximum range, the complete obstacle configuration is generally not available from the initial state and local navigation decisions are made as new geometry enters the sensing region.

The three unseen environments in Figure 5 are constructed to introduce progressively larger geometric shifts relative to the training distribution. The *Low-Shift* environment preserves the general row-based obstacle structure of the training maps while increasing the obstacle count and reversing the initial and goal regions. The *Medium-Shift* environment retains a similar obstacle count but reduces the free-space gaps between obstacle rows, producing narrower navigation passages than those encountered during training. The *High-Shift* environment changes both obstacle geometry



(a) Low-Shift (b) Medium-Shift (c) High-Shift

Figure 5: Static unseen environments used for distribution-shift evaluation. Green and yellow regions indicate the admissible initial and goal regions, respectively, while obstacles are shown in red. The environments introduce increasingly different obstacle configurations relative to the hand-crafted training maps.

Table 8: Navigation efficiency on the unseen static environments with the runtime safety filters enabled. Path Length (PL) and Navigation Time (NT) are computed over successful episodes and reported as mean ± standard deviation. Dashes indicate that no successful episodes were available for computing PL and NT.

	Low-Shift		**Medium-Shift**		**High-Shift**	
Method	**PL (m)** ↓	**NT (s)** ↓	**PL (m)** ↓	**NT (s)** ↓	**PL (m)** ↓	**NT (s)** ↓
SEA–Nav[1]	11.57±0.17	13.70±0.17	10.95±0.30	14.21±0.27	−−	−−
CBF–RL[2]	14.86±0.78	32.02±4.02	14.93±0.76	34.93±4.3	15.72±1.19	38.99±8.45
Vanilla SAC[3]	12.76±0.41	26.75±1.54	12.38±0.68	26.31±2.13	12.20±0.70	25.97±1.77
CBF–QP[4]	12.05±0.38	25.08±1.34	11.50±0.45	23.77±1.55	11.82±1.11	25.42±4.02
CBF–CF[4]	12.26±0.36	26.85±1.06	11.72±0.42	24.84±1.22	11.72±0.35	25.93±2.06
bCBF–LSE–QP[5]	13.06±0.37	32.74±1.11	12.20±0.45	29.50±1.67	12.33±0.54	29.91±2.07
SAFTER[5]	12.73±0.32	30.66±1.18	11.87±0.38	28.10±1.40	12.03±0.32	29.43±1.34

Refs.: [1] (Chen et al., 2026); [2] (Yang et al., 2025); [3] (Haarnoja et al., 2018); [4] (Rabiee & Hoagg, 2024); [5] **Ours**.

and spatial arrangement by introducing different obstacle radii, a square obstacle, and tighter inter-obstacle clearances than the Medium-Shift map. For all evaluation environments, the admissible initial and goal regions maintain the minimum wall-clearance requirements used by the evaluation protocol.

E.2 Path Length and Navigation Time of Filter-Enabled Evaluations on Unseen Environments

Table 8 reports path length and navigation time only over successful episodes for the filter-enabled unseen-environment evaluations. Path lengths are broadly comparable across the SAC/CBF methods, while SAFTER generally requires more navigation time than the current-state baselines and less than bCBF–LSE–QP or CBF-RL. Thus the success-rate gains under geometric shift should not be interpreted as uniformly shorter or faster successful trajectories; the efficiency metrics reflect the additional conservatism associated with recoverability-aware filtering. Because PL and NT are conditioned on success, they should be interpreted together with the success rates in Table 3.