SEA-Nav: Efficient Policy Learning for Safe and Agile Quadruped Navigation in Cluttered Environments

Shiyi Chen[1*], Mingye Yang[2*], Haiyan Mao[1], Jiaqi Zhang[1], Haiyi Liu[1]
Shuheng He[1], Debing Zhang[1], Zihao Qiu[1], Chun Zhang[1†]
Project Website: https://11chens.github.io/sea-nav/

***Abstract*— Efficiently training quadruped robot navigation in densely cluttered environments remains a significant challenge. Existing methods are either limited by a lack of safety and agility in simple obstacle distributions or suffer from slow locomotion in complex environments, often requiring excessively long training phases. To this end, we propose SEA-Nav (Safe, Efficient, and Agile Navigation), a reinforcement learning framework for quadruped navigation. Within diverse and dense obstacle environments, a differentiable control barrier function (CBF)-based shield constraints the navigation policy to output safe velocity commands. An adaptive collision replay mechanism and hazardous exploration rewards are introduced to increase the probability of learning from critical experiences, guiding efficient exploration and exploitation. Finally, kinematic action constraints are incorporated to ensure safe velocity commands, facilitating successful physical deployment. To the best of our knowledge, this is the first approach that achieves highly challenging quadruped navigation in the real world with minute-level training time.**

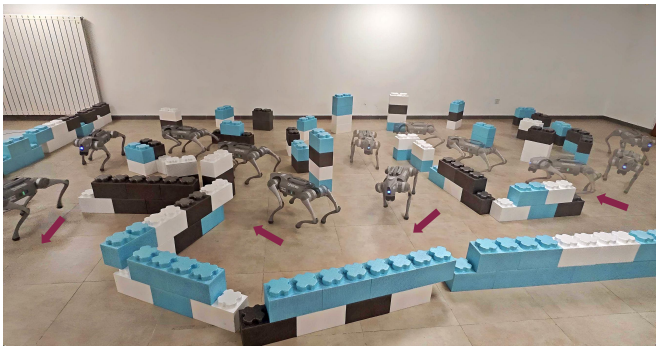

Fig. 1: SEA-Nav is trained in minute-level time and deployed zero-shot in a previously unseen maze. The robot successfully escapes using the out-of-distribution built-in MPC controller and onboard sparse LiDAR.

I. INTRODUCTION

Autonomous navigation in dense, cluttered environments remains a core challenge in robotics. Learning-based methods, especially imitation/self-supervised and imperative-learning paradigms, have shown strong local performance by fitting large-scale experience or learned cost priors [1]–[6]. However, these methods often require expensive labeled data and can fail catastrophically under out-of-distribution (OOD) dense or dynamic obstacles.

To overcome static-dataset limitations, deep reinforcement learning (DRL) has been widely adopted for its maneuverability and reactivity [7]–[15]. Yet RL in dense obstacles is difficult: obstacle penalties are hard to tune (large penalties cause over-conservative behavior, small penalties increase collisions), and low sample efficiency in long-horizon scenes makes it hard to balance goal reaching and aggressive avoidance. As a result, many methods need long training horizons or separate training stages [7], [14], [15], increasing development and validation cost.

To inject explicit safety boundaries into RL, recent work integrates classical safety control methods such as Velocity Obstacles (VO) and Control Barrier Functions (CBF) [16]–[22]. However, existing schemes still compromise either end-to-end credit assignment (when used as post-processing filters) [23]–[25] or stability under multiple constraints, where oscillation and conservative "freezing" can occur [26]–[29]. More fundamentally, they do not solve the key bottleneck of high-quality experience sampling in dense-risk regions.

To address these issues, we propose an end-to-end safe RL navigation framework that combines efficient experience sampling with a differentiable physical barrier. The framework has three core components:

First, we introduce Adaptive Collision-State Initialization (ACSI) to improve sample utilization in high-density scenes. After a collision, the system probabilistically resets the robot to a critical pre-collision state in the local high-risk area. Combined with a success-rate-based reset curriculum, this focuses training on bottleneck regions and rapidly accumulates valuable avoidance experience.

Second, we propose an end-to-end adaptive LSE-CBF safety projection layer. Instead of a rigid post-hoc filter, it provides an analytical geometric inductive bias in the action space. We fuse multiple LiDAR constraints with Log-Sum-Exp (LSE) and add a physical damping term ϵ_d to avoid numerical singularities and discrete-time "ping-pong" oscillations in narrow passages. The closed-form projection remains differentiable, so rewards backpropagate to adapt the safety gain α online.

Third, we introduce a kinematic action regularization loss in the action space to suppress dangerous commands. With high sampling efficiency and differentiable safety constraints,

*Equal contribution.
[1]Tsinghua University, Beijing, China.
[2]Imperial College London, London, UK.
†Corresponding author email: zhangchun@tsinghua.edu.cn

our method achieves zero-shot, safe, and agile deployment in extremely dense environments after only tens of minutes of training on a single RTX 4090 GPU (Fig. 1).

The main contributions of this paper are summarized as follows:

- **Adaptive Collision-State Initialization (ACSI):** A curriculum-guided critical-state replay strategy that addresses the sample-efficiency bottleneck of RL in dense obstacles.
- **End-to-end Adaptive LSE-CBF Layer:** A closed-form differentiable safety layer with LSE fusion and physical damping that suppresses multi-constraint oscillations and enables adaptive avoidance aggressiveness.
- **Efficient Training and Hardware-Safe Physical Deployment:** A safe action regularization loss that improves Sim-to-Real transfer and enables minute-level training (tens of minutes) for zero-shot deployment.

II. RELATED WORK

A. Learning-Based Robot Navigation

Early studies primarily relied on imitation learning and self-supervised learning (e.g., behavior cloning [30]–[33], environmental cost prediction [34]–[36]) for end-to-end control or to assist traditional planners. Although these methods perform well in specific static scenarios, they are prone to collision failures when encountering OOD dense or dynamic obstacles. DRL has been widely applied to navigation tasks, demonstrating high maneuverability. However, pure RL faces two severe challenges in dense obstacles. First, the sample efficiency is extremely low [4], [6], [37]. Collisions typically terminate the episode directly, causing the robot to spend most of its exploration time in collision-free free space, making it difficult to accumulate high-value "extreme avoidance" experiences. Second, pure RL lacks physical safety constraints, often leading to conservative behaviors under high collision penalties [15]. These issues often translate into long training schedules before deployment across diverse environments. Moreover, several approaches rely on multi-stage or multi-module training [7], [14], [15], [23], which further increases engineering and validation overhead. Designing an efficient experience sampling mechanism while constraining physical safety from the action space remains a key challenge.

Recent reactive local-avoidance methods improve short-horizon responsiveness in cluttered scenes, but they still require careful integration with policy learning to maintain global efficiency and safety consistency [38].

B. Safe RL and Barrier Functions

To provide explicit safety guarantees, prior works introduced VO as a post-processing safety shield. While VO can enforce hard kinematic constraints via optimization, it acts as a non-differentiable external filter that truncates gradient backpropagation. Moreover, its velocity-cone formulation causes extreme conservatism in dense obstacle scenarios [12]. Other studies attempted "soft guidance" by incorporating VO into the reward function [16], [17], but this compromises strict analytical geometric constraints in the action space.

In contrast, CBF formulations map spatial boundaries directly into control inputs via system dynamics, providing mathematically rigorous forward invariant set guarantees [18], [19], [22]. More importantly, CBF formulations offer the potential for closed-form analytical solutions, paving the way for fully differentiable safety layers. Despite this potential, traditional multi-constraint CBFs rely on the non-differentiable min operator, which triggers severe "ping-pong" control oscillations during boundary switching [21]. Furthermore, static Class-$\mathcal{K}$ parameters (α) fail to adapt to varying spatial densities, often leading to "Freezing Robot" deadlocks in narrow passages [26]. Therefore, integrating adaptive parameters with smooth fusion mechanisms into a fully differentiable, optimizer-free analytical layer remains a critical challenge.

III. METHOD

We design a single-stage RL framework. With efficient exploration and exploitation in diverse, highly occupied maps, our method completes navigation in dense environments with minute-level training time.

A. Key Challenges & Overview

Exploration-Exploitation Challenge in Dense Environments: In densely cluttered environments, commonly used collision and reaching rewards are hard to balance. Dense obstacles bring excessive collision penalties, causing conservative movement. Furthermore, if an episode terminates immediately upon collision or reaching the goal, the proportion of critical experiences drastically decreases. The robot struggles to learn obstacle avoidance in narrow, high-risk environments and goal-reaching in long-horizon tasks. Thus, we design ACSI to repeatedly replay critical trajectories before collisions, and set a goal-stay mechanism to deliberately prolong the learning of sparse experiences.

Safe RL Constraints: Without physical priors, pure RL training is inefficient and unsafe. Moreover, deploying safety barriers exclusively at test-time introduces a severe train-test mismatch. Unaware of these post-hoc interventions during training, the policy often generates actions that conflict with the shield at deployment, leading to uncooperative and suboptimal behaviors. A trainable barrier can shape the navigation policy during training, helping it learn safety awareness faster. Our solution relies on a Differentiable Barrier Layer based on Adaptive Multi-Constraint CBF.

Robot Kinematic Constraints: High-speed sharp turns are unsafe and can cause the robot to fall during real-world deployment. Common action smoothing imposes hard constraints that severely restrict the action space. Thus, we design an appropriate action loss to constrain unsafe action generation while ensuring exploration.

B. System Pipeline & MDP Formulation

This paper proposes a single-stage reinforcement learning framework that couples the Proximal Policy Optimization

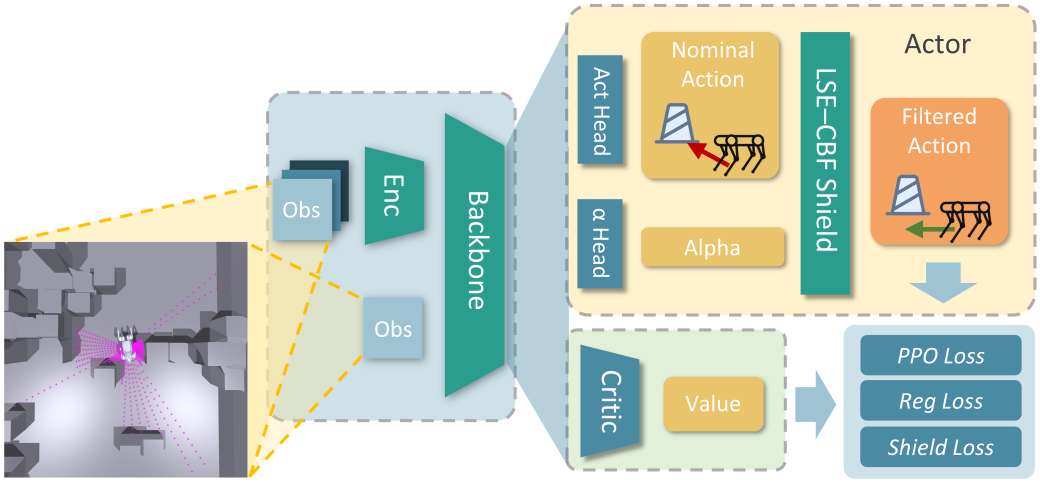

Fig. 2: Overview of the proposed SEA-Nav pipeline. LiDAR rays provide exteroceptive observations that are encoded and fused by the Encoder and Backbone into shared features. The Actor maps shared features to a navigation action head and a safety-gain α head, producing a nominal velocity command and an adaptive gain; the LSE-CBF Shield then solves for a safe velocity command. The Critic directly predicts state value from shared features. The Actor and Critic are optimized jointly with PPO, shield intervention, and kinematic regularization losses.

(PPO) algorithm with a differentiable physical barrier. As shown in Fig. 2.

Our Actor-Critic architecture is designed to be fully end-to-end differentiable, and all neural networks within the architecture (including the Encoder, Backbone, respective Heads, and Critic) are composed of Multi-Layer Perceptrons (MLPs). In the Actor network, a sequence of historical observations $O_{hist} = \{o_{t-H}, \dots, o_{t-1}\}$ (where the history length $H = 10$ frames) first passes through the Encoder to extract a low-dimensional latent feature vector z_t. This latent representation is then concatenated with the current observation o_t to form the joint state representation $x_t = [o_t, z_t]$, which is fed into a shared Backbone. The shared feature vector f_t extracted by the Backbone is subsequently split into two parallel network heads: the **Navigation Action Head** outputs the desired nominal velocity command $\bar{u}_t = [\bar{v}_x, \bar{v}_y, \bar{\omega}_z]^T$; the **Safety Gain Head** (α Head) dynamically outputs the barrier gain parameter α_t (ensuring $\alpha_t > 0$ via a Softplus activation function). The outputs of both heads ($\bar{u}_t$ and α_t) are fed into the LSE-CBF barrier layer to analytically compute the final safe command $u_{s,t} = [v_{x,s}, v_{y,s}, \omega_{z,s}]^T$, which drives the low-level locomotion controller. The Critic network, on the other hand, bypasses the safety barrier; it takes the joint state representation x_t and passes it through an MLP to directly estimate the state value $V(x_t)$.

The Markov Decision Process (MDP) for the system interacting with the environment is formulated as follows:

- **Observation Space:** The current observation $o_t \in \mathbb{R}^d$ includes the robot's base linear velocity $v^B \in \mathbb{R}^3$, base angular velocity $\omega^B \in \mathbb{R}^3$, projected gravity vector $g^B \in \mathbb{R}^3$, 2D local goal position $p^B_{goal} \in \mathbb{R}^2$ relative to the base frame, and a 2D LiDAR range scan $\rho \in \mathbb{R}^{41}$. The rays ρ_i cover an angular range of $[-\frac{2\pi}{3}, \frac{2\pi}{3}]$ rad with a resolution of $\frac{\pi}{30}$ rad, and a sensing range of 0.1 m to 3.0 m. To align with real-world hardware deployment, the update frequency of exteroception (LiDAR ρ and goal point p^B_{goal}) is set to 10 Hz, while proprioception (v^B, ω^B, g^B) is maintained at 50 Hz.
- **Action Space:** The high-level navigation policy outputs the safe body velocity command $u_{s,t} \in \mathbb{R}^3$, which is sent to a pre-trained low-level locomotion controller at 50 Hz to be converted into target joint torques $\tau \in \mathbb{R}^{12}$.
- **Reward Design:** Our reward formulation centers on three key components: a clearance reward r_{clear} to encourage traversing cluttered regions, a velocity reward r_{velo} for goal-directed progress, and a stuck penalty r_{stuck} to facilitate escape from local minima. Due to space constraints, the comprehensive formulation is detailed in Appendix V.

C. Adaptive Collision-State Initialization (ACSI)

To break the exploration bottleneck in high-density obstacle environments, we propose an adaptive collision replay mechanism. In traditional training, immediate termination upon collision wastes substantial computational resources on obstacle-free, smooth areas. In our mechanism, the system records the interaction history; once a collision occurs, the environment is not immediately reset to the initial state. Instead, it extracts the historical state from a short period before the collision and, with a certain probability, resets the robot to the local high-risk area prior to the collision, retaining its true pose and velocity at that time.

This "critical state reloading" mechanism allows the most challenging extreme obstacle avoidance scenarios in navigation to be trained repeatedly. To balance the accumulation of obstacle avoidance experience and global goal reaching, we created a curriculum reset based on the success rate. The reset probability P_{reset} dynamically increases as the

probability of reaching the goal increases:

$$P_{reset} = P_{min} + (P_{max} - P_{min}) \cdot \text{clip}\left(L_{goal}, 0, 1\right) \quad (1)$$

Where: P_{min} and P_{max} are the minimum and maximum reset probabilities. The curriculum update is $L_{goal} \leftarrow L_{goal} + \mathbf{1}[d < d_{up}] - \mathbf{1}[d > d_{down}]$.

This curriculum guides the robot to prioritize moving towards the goal in the early stages of training, while forcing it to focus on improving extreme obstacle avoidance capabilities in high-risk areas during the later stages.

D. Differentiable Adaptive LSE-CBF Layer

To enhance the safety of the policy network in complex environments, this module introduces a fully differentiable multi-constraint CBF layer.

1) LSE Aggregation for Smooth Safety Constraints: In our LiDAR-based navigation task, the safety set $\mathcal{C}$ is defined by the intersection of constraints from $N = 41$ discrete rays. A direct definition uses $h(x) = \min_i h_i(x)$, where $h_i(x)$ is the distance residue between the i-th ray measurement and the safety radius. The non-differentiable min causes gradient jumps during constraint switching. For instance, in a narrow corridor, a marginal shift in proximity (e.g., from a $0.50\,\text{m}$ left wall to a $0.51\,\text{m}$ right wall) forces the active constraint's spatial gradient to abruptly flip $180°$, triggering directional chattering. To address this, we adopt the Log-Sum-Exp (LSE) formulation—a standard smooth approximation in safe control [20], [21]—to fuse all discrete constraints into a global composite safety function $h(x)$:

$$h(x) = -\frac{1}{k} \ln \left(\sum_{i=1}^{N} \exp(-k \cdot h_i(x)) \right) \quad (2)$$

where $k > 0$ is the smoothing coefficient. This formulation ensures that $h(x)$ is continuously differentiable, providing a smooth gradient landscape for both control execution and neural network backpropagation.

2) Damped Analytical Safety Projection: Given the smoothed safety margin $h(x)$, the CBF layer modulates the Actor's nominal action $\bar{u}(x)$ to satisfy the forward invariance condition $\dot{h}(x, u) \geq -\alpha h(x)$. This requirement yields the standard CBF-QP [20], [39]:

$$\min_{u_s} \frac{1}{2} ||u_s - \bar{u}(x)||^2 \quad \text{s.t.} \quad \langle L_g h(x), u_s \rangle + \alpha h(x) \geq 0 \quad (3)$$

Here, $L_g h(x) = \nabla h(x)$ is the Lie derivative representing the gradient direction of the safety margin, and $\alpha > 0$ is the Class-$\mathcal{K}$ parameter.

While LSE resolves directional chattering by smoothly blending gradients, it introduces a challenge in perfectly symmetric environments. In narrow passages where opposing hazard gradients (e.g., from left and right walls) equally blend and cancel each other out, the composite gradient vanishes ($||L_g h(x)||^2 \to 0$). The closed-form solution of (3) in [20] then yields a vanishing denominator, causing the correction magnitude to blow up and leading to numerical divergence. To resolve this, we introduce a physical damping term ϵ_d in the closed-form solution, resulting in the modified safety command $u_s(x)$:

$$u_s(x) = \bar{u}(x) + \underbrace{\max\left\{0, \frac{-(\langle L_g h(x), \bar{u}(x) \rangle + \alpha \cdot h(x))}{||L_g h(x)||^2 + \epsilon_d}\right\}}_{\text{Correction Magnitude } \eta} \cdot \underbrace{L_g h(x)}_{\text{Direction}} \quad (4)$$

where the $\max\{0, \cdot\}$ operator acts as a logical switch: it leaves safe actions unaltered ($\eta = 0$), while generating a strictly positive magnitude η to project unsafe actions along the safety gradient $L_g h(x)$. Crucially, the damping term ϵ_d effectively caps the maximum correction magnitude, preventing velocity explosion when gradients vanish in highly constrained spaces.

3) End-to-End Differentiability as Inductive Bias: Due to discrete-time execution and sensor noise, this module does not mathematically guarantee absolute zero-collision. Instead, it serves as a **differentiable inductive bias**. Since (4) is composed entirely of differentiable operations, gradients from task rewards can backpropagate through the projection seamlessly. This allows the network to autonomously learn to increase α (be aggressive) in open areas and decrease α (be conservative) in narrow spaces, effectively internalizing physical safety constraints into the policy itself.

E. Loss Function Design

To foster a synergistic relationship between the policy network and the LSE-CBF layer, we introduce a **Shield Intervention Loss** (L_{shield}). This loss minimizes the discrepancy between the nominal command $\bar{u}_t$ and the shielded command $u_{s,t}$, while discouraging overly small α that would trigger near-complete shield intervention and introduce dynamics risks:

$$L_{shield} = ||u_{s,t} - \bar{u}_t||^2 + [\alpha_{min} - \alpha_t]_+^2 \quad (5)$$

where $[x]_+ = \max\{0, x\}$ and α_{min} is a lower bound for the adaptive gain.

Furthermore, since the execution performance of quadruped locomotion is highly sensitive to the navigation commands, we aggregate physical constraints and network smoothness into a unified **Kinematic Regularization Loss** (L_{reg}). This regularization comprises two sub-components: a range penalty (L_{range}) and a smoothness penalty (L_{smooth}).

First, to prevent destructive behaviors, the velocity command range loss penalizes outputs that exceed the hardware safety limits:

$$L_{range} = \sum_{j \in \{x,y,\omega\}} \left(u_{s,t}^j - \text{clip}(u_{s,t}^j, u_j^{\min}, u_j^{\max}) \right)^2 \quad (6)$$

where $u^{\min}, u^{\max} \in \mathbb{R}^3$ are the element-wise lower and upper bounds of the robot's kinematic capabilities.

Second, to promote smooth transitions in both action and value predictions, we adopt a Lipschitz continuity constraint [40]:

$$L_{smooth} = \lambda_\pi D\left(\pi_\theta(x_t), \pi_\theta(\bar{x}_t)\right) + \lambda_V D\left(V_\phi(x_t), V_\phi(\bar{x}_t)\right) \quad (7)$$

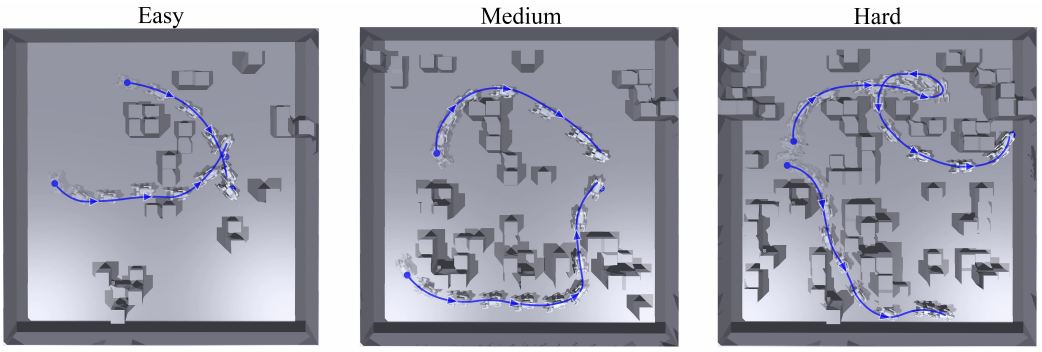

Fig. 3: SEA-Nav trajectories in three navigation scenarios of increasing difficulty. Each subfigure shows two distinct start-goal trials. SEA-Nav successfully traverses narrow passages and performs timely maneuvering adjustments to avoid entrapment in cluttered environments.

TABLE I: Different difficulty levels are evaluated, with the mean and standard deviation reported over 3 runs.

Method	Easy			Medium			Hard		
	SR (%) ↑	CR (%) ↓	TR (%) ↓	SR (%) ↑	CR (%) ↓	TR (%) ↓	SR (%) ↑	CR (%) ↓	TR (%) ↓
SEA-Nav (Ours)	**100.00** (±0.00)	**0.00** (±0.00)	**0.00** (±0.00)	**97.00** (±0.82)	**1.00** (±0.82)	**2.00** (±0.00)	**90.00** (±1.63)	**5.00** (±0.82)	**5.00** (±0.82)
SEA-Nav w/o ACSI	**100.00** (±0.00)	**0.00** (±0.00)	**0.00** (±0.00)	93.00 (±0.82)	2.33 (±0.47)	4.67 (±0.47)	83.00 (±2.45)	8.00 (±2.16)	9.00 (±1.63)
SEA-Nav w/o Shield	97.67 (±0.47)	1.33 (±0.47)	1.00 (±0.00)	90.00 (±0.82)	2.67 (±0.94)	7.33 (±1.70)	74.33 (±3.30)	11.67 (±1.70)	14.00 (±2.16)
SEA-Nav w/o L_{reg}	96.33 (±0.47)	2.67 (±0.47)	1.00 (±0.00)	82.33 (±2.05)	6.00 (±0.82)	11.67 (±1.25)	57.00 (±4.08)	18.00 (±2.16)	25.00 (±2.16)
ABS [7]	95.00 (±0.00)	4.67 (±0.47)	0.33 (±0.47)	75.33 (±1.25)	9.33 (±0.47)	15.33 (±1.70)	45.33 (±3.40)	22.00 (±1.41)	32.67 (±2.05)
OCR [23]	95.33 (±0.47)	3.33 (±0.47)	1.33 (±0.47)	67.33 (±2.49)	31.33 (±2.87)	32.67 (±2.49)	56.00 (±3.27)	18.33 (±2.49)	25.67 (±2.49)
SEASAN [14]	95.67 (±0.47)	2.67 (±0.47)	1.33 (±0.47)	71.33 (±3.09)	10.33 (±1.89)	28.67 (±2.16)	77.67 (±4.62)	14.33 (±1.89)	22.33 (±4.19)

where $\bar{x}_t = x_t + \beta \cdot (x_{t+1} - x_t)$ is an interpolated state with the coefficient $\beta \sim \mathcal{U}(-1, 1)$, and $D(\cdot, \cdot)$ denotes the mean squared error (MSE). Here, π_θ and V_ϕ represent the policy and value networks taking the joint state representation x_t as input, while λ_π and λ_V are their corresponding weighting factors. This constraint improves deployment safety by suppressing abrupt changes in action and value outputs, thereby reducing the risk of falls and motor overheating during Sim-to-Real locomotion. The total kinematic regularization is thus given by $L_{reg} = L_{range} + L_{smooth}$.

Finally, these auxiliary losses are integrated with the standard PPO objective [41]. The total loss function used to update the entire end-to-end network is formulated as:

$$L_{total} = L_{PPO} + \lambda_{shield} L_{shield} + \lambda_{reg} L_{reg} \tag{8}$$

where λ_{shield} and λ_{reg} are the weighting hyperparameters for each respective loss term.

F. Training in Simulation

Using the Isaac Gym platform [42], we first trained a standard velocity tracking controller based on [43]. For the navigation task, we designed 10×10 rooms heavily occupied by diverse obstacles, as illustrated in Fig. 2. During training, the episode ends only after the robot stays near the goal for a period, increasing the proportion of goal-reaching experiences. We also expanded the collision bodies to enhance collision perception and lower the policy's safety margin.

IV. EXPERIMENTS

A. Simulation Experiments

We designed rooms with varying occupancy difficulties: Easy, Medium, and Hard (Fig. 3). We conducted 100 experiments for each mode with randomized start points, end points, and initial headings. We utilized three metrics: Success Rate (SR), Collision Rate (CR), and Timeout Rate (TR). Timeout (failing to reach within 30s) reflects whether the robot is stuck in local minima.

Our training design achieves a strong balance among agility, safety, and task completion, especially in hard environments. The ablation results in Table I show that ACSI improves success by repeatedly revisiting high-risk states, which increases near-obstacle experience and reduces collision-induced failures. Removing the differentiable Shield degrades both safety and task completion, indicating that shaping the action space with the LSE-CBF layer provides more efficient avoidance learning and a stronger safety bias. The kinematic regularization further guides the policy toward hardware-feasible velocity commands, improving stability and deployment robustness.

B. Case Study

We further analyze the contributions of ACSI, the Shield, and the kinematic regularization in a challenging scenario with continuous obstacle avoidance and local minima. Fig. 4 visualize the trajectories and velocity profiles. With the

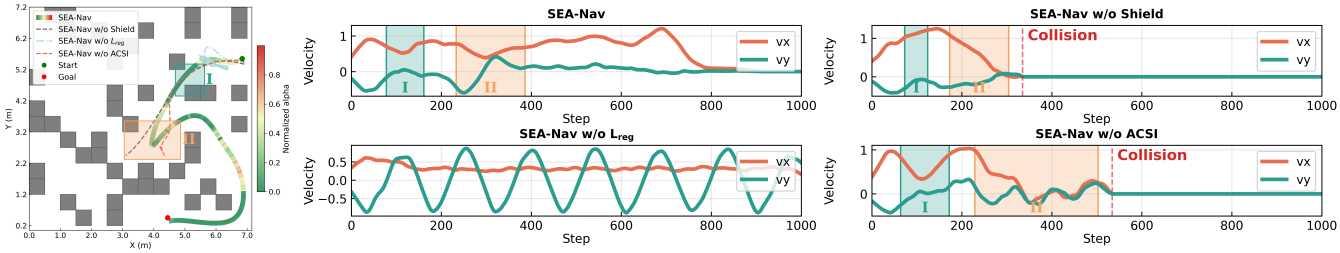

Fig. 4: (Left) Trajectory plot; (right) velocity profiles. SEA-Nav maintains larger obstacle clearance and smoother speed variations. The safety gain α decreases in hazardous regions so the CBF Shield dominates, and increases in safe regions where the nominal navigation command takes the lead.

TABLE II: Real-world deployment results. AS: average speed (m/s); SEA-Nav-b: built-in MPC controller and onboard sparse LiDAR.

Method	Cluttered Room			Dynamic Obstacle			Obstacle Course			S-Blend Track		
	SR (%) ↑	CR (%) ↓	AS (m/s) ↑	SR (%) ↑	CR (%) ↓	AS (m/s) ↑	SR (%) ↑	CR (%) ↓	AS (m/s) ↑	SR (%) ↑	CR (%) ↓	AS (m/s) ↑
SEA-Nav	**100**	10	1.6	90	20	1.5	**100**	10	**1.2**	**100**	10	**1.4**
SEA-Nav-b	**100**	10	0.9	**100**	20	0.8	90	10	0.7	**100**	10	0.5
ABS [7]	70	50	**2.1**	60	10	**1.7**	0	100	0.6	80	20	0.8
OCR [23]	90	20	1.7	90	10	1.5	30	100	1.0	90	20	1.3
REASAN [14]	90	20	1.3	90	10	1.3	40	100	0.9	90	30	0.9
SLAM	**100**	**0**	0.5	80	40	0.4	**100**	**0**	0.4	80	**0**	0.3

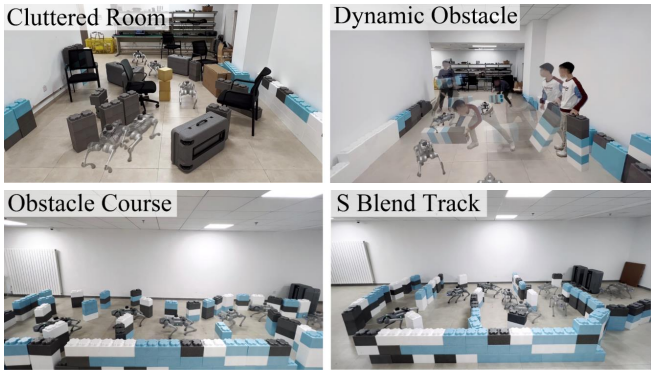

Fig. 5: Real-world experimental environments. Ten trials are conducted in each environment.

full method, the adaptive gain α decreases when the robot enters narrow passages, increasing shield intervention and enabling safe passage; the turning phase also exhibits smooth, bounded velocity transitions. The stuck-aware reward helps the robot escape local minima, while the regularization loss suppresses aggressive velocity spikes that would otherwise lead to unsafe impacts. In contrast, removing regularization yields abrupt velocity changes and collisions, and removing ACSI or the Shield reduces exposure to high-risk states or removes explicit safety shaping, resulting in avoidable collisions in tight regions.

C. Hardware Setup

We evaluated our method on the Unitree Go2 quadruped robot across different obstacle environments. We designed two deployment schemes for SEA-Nav:

1) Using the robot's onboard sparse LiDAR L1 for perception and the built-in MPC controller (max 1 m/s).
2) Using a higher-precision RPLIDAR A2 for perception and our trained agile policy [43] as the controller.

Different from prior lidar-based approaches [13], [14], [23], which require high-precision ray measurements and do not support the stock Unitree L1 LiDAR, our first scheme uses the native sensor and controller to enable a plug-and-play, low-cost deployment path with strong real-world viability. We summarize the point-cloud differences between the two LiDAR setups in Appendix V (Fig. 6). And we also compared with the Unitree built-in SLAM algorithm.

D. Results Analysis

Table II summarizes the quantitative results. Our method achieves safer obstacle avoidance in highly difficult navigation environments compared to SOTA methods. It performs exceptionally well in continuous turning and local planning. In contrast, while some SOTA methods move faster, they either get stuck or crash at dangerous speeds when facing narrow corners. The SLAM method has a high success rate but relies on lengthy mapping and moves slowly, failing to handle dynamic obstacles.

V. CONCLUSION

In this paper, we proposed SEA-Nav, a highly efficient RL navigation framework for quadruped robots that integrates ACSI experience replay and an end-to-end differentiable LSE-CBF shield. It achieves agile and safe deployment with minute-level training.

Limitations: The current algorithm only supports flat-ground navigation and lacks detection capabilities for slopes or stairs. It can overcome simple local optima but is still prone to getting stuck in complex mazes and dead ends.

Future Work: We plan to incorporate global navigation algorithms or memory mechanisms to resolve complex local optima and expand terrain adaptability.

APPENDIX

Reward Function

TABLE III: Reward terms and weights (extracted from the implementation). Let d denote the goal distance; $\mathbf{v} = [v_x, v_y]$ the base linear velocity in the horizontal plane; and $\boldsymbol{\omega} = [\omega_x, \omega_y, \omega_z]$ the base angular velocity. $\mathbf{1}[\cdot]$ is the indicator function. θ is the heading error to the goal. ϕ is the angular offset to the most open ray direction, and c_{front} is the minimum ray clearance in a frontal cone. ρ_i is the i-th ray distance. For collisions, $\mathbf{f}_k$ is the contact force magnitude on body group k, and β_k are body-dependent coefficients (omitted for brevity). For stuck detection, we track recent planar base positions $\{\mathbf{p}_t\}_{t=1}^{T}$ and define $\Delta p_{\max} = \max_t \|\mathbf{p}_t - \mathbf{p}_1\|_2$.

Term	Purpose	Compact expression	Weight
r_{term}	episode termination cost	$\mathbf{1}[\text{terminated}]$	-100
r_{reach}	tight goal reaching	$\frac{1}{1+2d^2}\,\mathbf{1}[d<0.5]$,	$+10$
r_{velo}	move toward goal	$\cos\theta\, v_x + \frac{1}{1+2d^2}$	$+15$
r_{clear}	clearance-aware motion	$\mathbf{1}[d>1]\,\cos\phi\, v_x + \mathbf{1}[d\le 1]\,\frac{1}{1+2d^2}$	$+15$
r_{stuck}	penalize getting stuck	$\mathbf{1}[d>1]\,\mathbf{1}[\Delta p_{\max}<0.1]\,\mathbf{1}[v_x>0.0]\,\mathbf{1}[\lvert\omega_z\rvert<1.0]$	-5
r_{coll}	penalize collisions	$\left(1+4(\lVert\mathbf{v}\rVert_2^2+\omega_z^2)\right)\sum_k \beta_k\,\mathbf{1}[\lVert\mathbf{f}_k\rVert>0.1]$	-4
r_{ω}	reduce angular motion	$\lVert[\omega_x,\omega_y]\rVert_2$	-0.05

LiDAR Point Cloud Comparison

We compare the native Unitree LiDAR L1 and the external RPLIDAR A2 used in our two deployment schemes. The L1 provides a sparse, low-cost 3D point cloud that is directly supported by the onboard controller, enabling plug-and-play deployment. The A2 provides a denser 2D scan that improves near-obstacle geometry for higher-speed navigation. Fig. 6 illustrates the qualitative difference between the two sensing streams.

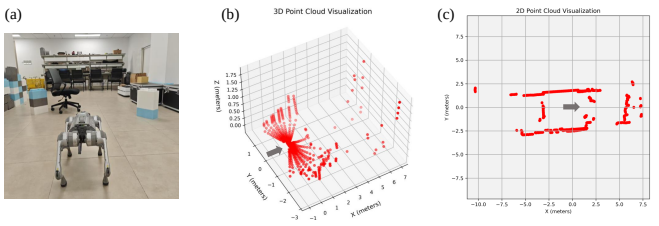

Fig. 6: Comparison of LIDAR point clouds. The robot's position and orientation are indicated by the gray arrow. (a): The robot's actual environment. (b): Sparse point cloud from the Unitree Go2 robot's onboard LIDAR, consisting of approximately 1,000 3D points per frame. (c): Denser point cloud from the RPLIDAR A2, consisting of approximately 2,000 2D points per frame.

Domain Randomization

TABLE IV: Domain-randomization parameters.

Term	Value
Rays delay	$\mathcal{U}(40, 80)$ ms
Gravity noise	$\mathcal{U}(-0.05, 0.05)$
Linear velocity noise	$\mathcal{U}(-0.1, 0.1)$ m/s
Angular velocity noise	$\mathcal{U}(-0.1, 0.1)$ rad/s
Friction coefficient factor	$\mathcal{U}(-0.2, 1.25)$
Mass perturbation	$\mathcal{U}(-1.5, 1.5)$ kg

Hyper Parameters

TABLE V: Hyper parameters for training.

Term	Value
P_{min}	0.1
P_{max}	0.5
d_{up}	0.5 m
d_{down}	2.0 m
u^{min}	[-0.5, 0.8, 1.0]
u^{max}	[1.7, 0.8, 1.0]
ϵ_d	1.0
λ_{shield}	0.1
α_{min}	0.1
λ_{reg}	1.0
λ_{π}	0.05
λ_V	0.005
Episode Duration	60 s
Entropy Coefficient	0.003

REFERENCES

[1] D. Shah, A. Sridhar, A. Bhorkar, N. Hirose, and S. Levine, "Gnm: A general navigation model to drive any robot," *arXiv preprint arXiv:2210.03370*, 2022.

[2] G. Kahn, P. Abbeel, and S. Levine, "Badgr: An autonomous self-supervised learning-based navigation system," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 1312–1319, 2021.

[3] A. Loquercio, E. Kaufmann, R. Ranftl, M. Müller, V. Koltun, and D. Scaramuzza, "Learning high-speed flight in the wild," *Science Robotics*, vol. 6, no. 59, p. eabg5810, 2021.

[4] F. Yang, C. Wang, C. Cadena, and M. Hutter, "iPlanner: Imperative Path Planning," in *Proceedings of Robotics: Science and Systems*, Daegu, Republic of Korea, July 2023.

[5] P. Roth, J. Nubert, F. Yang, M. Mittal, and M. Hutter, "Viplanner: Visual semantic imperative learning for local navigation," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 5243–5249.

[6] K.-H. Zeng, Z. Zhang, K. Ehsani, R. Hendrix, J. Salvador, A. Herrasti, R. Girshick, A. Kembhavi, and L. Weihs, "Poliformer: Scaling on-policy rl with transformers results in masterful navigators," *arXiv*, 2024.

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

[8] D. Hoeller, N. Rudin, D. Sako, and M. Hutter, "Anymal parkour: Learning agile navigation for quadrupedal robots," *Science Robotics*, vol. 9, no. 88, p. eadi7566, 2024.

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

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

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

[12] Z. Xu, X. Han, H. Shen, H. Jin, and K. Shimada, "Navrl: Learning safe flight in dynamic environments," *arXiv preprint arXiv:2409.15634*, 2024.

[13] Z. Wang, T. Ma, Y. Jia, X. Yang, J. Zhou, W. Ouyang, Q. Zhang, and J. Liang, "Omni-perception: Omnidirectional collision avoidance for legged locomotion in dynamic environments," 2025. [Online]. Available: https://arxiv.org/abs/2505.19214

[14] Q. Yuan, Z. Cao, M. Cao, and K. Li, "Reasan: Learning reactive safe navigation for legged robots," 2025. [Online]. Available: https://arxiv.org/abs/2512.09537

[15] Y. Zhong, C. Zhang, T. He, and G. Shi, "Bridging adaptivity and safety: Learning agile collision-free locomotion across varied physics," *arXiv preprint arXiv:2501.04276*, 2025.

[16] Z. Xie and P. Dames, "Drl-vo: Learning to navigate through crowded dynamic scenes using velocity obstacles," *IEEE Transactions on Robotics*, vol. 39, no. 4, p. 2700–2719, Aug. 2023. [Online]. Available: http://dx.doi.org/10.1109/TRO.2023.3257549

[17] R. Han, S. Chen, S. Wang, Z. Zhang, R. Gao, Q. Hao, and J. Pan, "Reinforcement learned distributed multi-robot navigation with reciprocal velocity obstacle shaped rewards," 2022. [Online]. Available: https://arxiv.org/abs/2203.10229

[18] A. Thirugnanam, J. Zeng, and K. Sreenath, "Safety-critical control and planning for obstacle avoidance between polytopes with control barrier functions," in *2022 International Conference on Robotics and Automation (ICRA)*, 2022, pp. 286–292.

[19] E. Sabouni, H. M. S. Ahmad, V. Giammarino, C. G. Cassandras, I. C. Paschalidis, and W. Li, "Reinforcement learning-based receding horizon control using adaptive control barrier functions for safety-critical systems," 2025. [Online]. Available: https://arxiv.org/abs/2403.17338

[20] T. G. Molnar and A. D. Ames, "Composing control barrier functions for complex safety specifications," *IEEE Control Systems Letters*, vol. 7, pp. 3615–3620, 2023.

[21] C. Wang, X. Wang, Y. Dong, L. Song, and X. Guan, "Multi-constraint safe reinforcement learning via closed-form solution for log-sum-exp approximation of control barrier functions," 2025. [Online]. Available: https://arxiv.org/abs/2505.00671

[22] P. Mestres, S. S. Mousavi, P. Ong, L. Yang, E. Das, J. W. Burdick, and A. D. Ames, "Explicit control barrier function-based safety filters and their resource-aware computation," 2025. [Online]. Available: https://arxiv.org/abs/2512.10118

[23] A. Lin, S. Peng, and S. Bansal, "One filter to deploy them all: Robust safety for quadrupedal navigation in unknown environments," *arXiv preprint arXiv:2412.09989*, 2024.

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

[25] R. Cheng, G. Orosz, R. M. Murray, and J. W. Burdick, "End-to-end safe reinforcement learning through barrier functions for safety-critical continuous control tasks," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 33, no. 01, 2019, pp. 3387–3395.

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

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

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

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

[30] A. Sadat, S. Casas, M. Ren, X. Wu, P. Dhawan, and R. Urtasun, "Perceive, predict, and plan: Safe motion planning through interpretable semantic representations," in *European Conference on Computer Vision*. Springer, 2020, pp. 414–430.

[31] M. Pfeiffer, M. Schaeuble, J. Nieto, R. Siegwart, and C. Cadena, "From perception to decision: A data-driven approach to end-to-end motion planning for autonomous ground robots," in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 1527–1533.

[32] S. Lv, Y. Gao, J. Che, and Q. Quan, "Autonomous drone racing: time-optimal spatial iterative learning control within a virtual tube," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 3197–3203.

[33] D. Shah, A. Sridhar, N. Dashora, K. Stachowicz, K. Black, N. Hirose, and S. Levine, "Vint: A foundation model for visual navigation," *arXiv preprint arXiv:2306.14846*, 2023.

[34] M. Elnoor, K. Weerakoon, A. J. Sathyamoorthy, T. Guan, V. Rajagopal, and D. Manocha, "Amco: Adaptive multimodal coupling of vision and proprioception for quadruped robot navigation in outdoor environments," 2024. [Online]. Available: https://arxiv.org/abs/2403.13235

[35] F. Gao, Z. Chai, and S. Tian, "Local semantic map-based navigation system for quadruped inspection robot in complex terrains," in *2025 8th International Conference on Advanced Algorithms and Control Engineering (ICAACE)*, 2025, pp. 570–575.

[36] Q. Zhu, Z. Sun, S. Xia, G. Liu, K. Ma, L. Pei, Z. Gong, and C. Jin, "Learning-based traversability costmap for autonomous off-road navigation," 2024. [Online]. Available: https://arxiv.org/abs/2406.08187

[37] J. Ye, D. Batra, A. Das, and E. Wijmans, "Auxiliary tasks and exploration enable objectnav," 2021.

[38] V. Reijgwart, M. Pantic, R. Siegwart, and L. Ott, "Waverider: Leveraging hierarchical, multi-resolution maps for efficient and reactive obstacle avoidance," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 13 157–13 163.

[39] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs for safety critical systems," *IEEE Transactions on Automatic Control*, vol. 62, no. 8, p. 3861–3876, Aug. 2017. [Online]. Available: http://dx.doi.org/10.1109/TAC.2016.2638961

[40] T. Kobayashi, "L2c2: Locally lipschitz continuous constraint towards stable and smooth reinforcement learning," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 4032–4039.

[41] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[42] N. Rudin, D. Hoeller, P. Reist, and M. Hutter, "Learning to walk in minutes using massively parallel deep reinforcement learning," in *Conference on Robot Learning*. PMLR, 2022, pp. 91–100.

[43] S. Chen, Z. Wan, S. Yan, C. Zhang, W. Zhang, Q. Li, D. Zhang, and F. U. D. Farrukh, "Slr: Learning quadruped locomotion without privileged information," in *Proceedings of The 8th Conference on Robot Learning*, vol. 270. PMLR, 06–09 Nov 2025, pp. 3212–3224. [Online]. Available: https://proceedings.mlr.press/v270/chen25e.html