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

Learning Collision-free Object Goal Pushing for Quadruped Robots with Safe Corridors

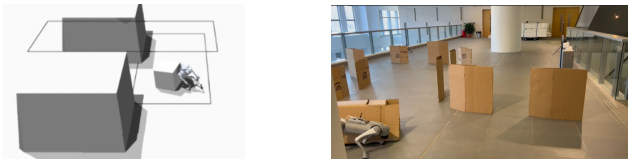
Gabriel Lai[1], Yi Wong[1], Chung Yui Yeung[1], Shaohang Xu[1*], Zhi Chen[2], Chin Pang Ho[1]

***Abstract*— While recent advancements in reinforcement learning have enabled quadrupedal robots to perform non-prehensile manipulation tasks like pushing, existing methods have largely overlooked the critical challenge of obstacle avoidance. In this paper, we address this significant limitation by introducing a novel reinforcement learning (RL) framework that controls a quadrupedal robot to push large objects in cluttered, real-world environments. In particular, obstacle avoidance is integrated as a primary objective directly into the policy training process. To achieve this, we propose to represent the traversable space with a low-dimensional safe corridor, a method that is both computationally efficient and highly effective. This approach avoids the need for complex and resource-intensive training pipelines typically required for processing high-dimensional sensor data. We validate our policy through extensive experiments in both simulation and the real world. The implementation code will be released to benefit the research community.**

I. Introduction

Legged robots, particularly quadrupedal platforms, have demonstrated remarkable mobility and maneuverability across diverse and challenging terrains [1]. However, their ability to actively interact with the environment remains a significant challenge. A common approach to address this limitation is to equip quadrupedal robots with a manipulator, enabling them to perform grasping-based interactions. The control of such systems is typically achieved through either model-based [2], [3] or model-free [4] whole-body control methods. While these grasping-based approaches are effective for interacting with objects of known geometry and manageable mass, they present substantial challenges when dealing with large or heavy objects due to the mechanical constraints of the manipulator and the inherent instability of the robot.

Recently, non-prehensile manipulation has garnered increasing research interest as a viable alternative for interacting with large-scale objects [5]. Specifically, pushing-based manipulation allows quadrupedal robots to displace large objects without the need for a bulky and heavy robotic arm [6]. This approach simplifies the robot's mechanical structure but introduces significant control complexities. While recent advancements in reinforcement learning have shown some progress in this area [7], [8], existing studies have primarily focused on pushing tasks in obstacle-free environments. In real-world applications, however, robots must operate in cluttered and dynamic spaces. Consequently, robust obstacle avoidance is a critical, yet often overlooked, requirement for successful non-prehensile manipulation.



(a) (b)

Fig. 1. Experimental results of our method on the Unitree Go2 robot. (a) Fine-Grained Locomotion Control: Complex object pushing setting in simulation. (b) Corresponding Environment in Reality

In this paper, we address a novel and challenging task: controlling a quadrupedal robot to push an object to a target location in a cluttered environment (as shown in Fig. 1). Unlike prior work, our research places a significant emphasis on incorporating obstacle avoidance as a primary objective during non-prehensile manipulation. To this end, we propose a novel RL-based policy learning framework that leverages the concept of a safe corridor. A safe corridor, a well-established concept in trajectory planning, defines the traversable region around the robot [9]. In traditional robot learning methods, obstacle avoidance for robots is often handled by directly feeding high-dimensional sensor data, such as point clouds from a LiDAR [10] or images from a camera [11], into the policy network. However, in simulation environments like Isaac Sim, processing such data is computationally expensive, demanding significant GPU memory. Furthermore, it often necessitates complex training techniques, such as knowledge distillation or multi-stage curriculum learning [12]. In contrast, a safe corridor can be represented by a low-dimensional vector (e.g., a simple rectangular corridor requires only a 4-dimensional vector), providing a compact and efficient representation of the traversable space. The key contributions of this paper are summarized as follows:

- We propose a novel reinforcement learning framework for controlling a quadrupedal robot to perform pushing-based manipulation in cluttered environments. Crucially, we treat obstacle avoidance as a core task requirement, which distinguishes our work from previous studies.
- We introduce the use of a safe corridor as a key component in the observation space and reward function design for policy training. This representation naturally

* Corresponding Author
[1] Department of Data Science, City University of Hong Kong
[2] Department of Decisions, Operations and Technology, CUHK Business School, The Chinese University of Hong Kong
The implementation code will be released at https://github.com/gabrilai3/ICRA2026-Object-Push-Safe-Corridor.

serves as an effective constraint for obstacle avoidance, seamlessly integrating into the framework for quadrupedal navigation and locomotion learning. The policy is a low-level controller designed to seamlessly ingest outputs from existing high-level convex corridor planners.
- Our RL framework is computationally efficient, enabling training on a lightweight platform (e.g., NVIDIA RTX 4070 TiS) in approximately one hour, with direct deployability on a real robot. This approach circumvents the need for complex and expensive training setups to process high-dimensional sensor data. We validate our method through extensive experiments in both simulation and the real world, demonstrating robust performance in a variety of cluttered environments. The implementation code will be made publicly available to benefit the robotics community.

II. Related Work

A. Mobile Manipulation Control

Mobile manipulators are a traditional solution for object transportation. Model-based methods, such as trajectory optimization and model predictive control (MPC), are commonly employed for their control [2], [3], [13]–[16]. For example, Mittal et al. [15] introduced a two-stage hierarchical framework for mobile manipulators to interact with articulated objects in unknown environments, demonstrating that an MPC whole-body planner is particularly suitable for real-world tasks like opening kitchen cabinets. Similarly, Dadiotis et al. [16] presented an experimental evaluation of a whole-body MPC framework on a dual-arm quadruped, showcasing its ability to perform complex loco-manipulation tasks without requiring an instantaneous whole-body controller. However, the real-time performance of these methods is often limited by their high computational cost. In recent years, model-free methods have successfully addressed this limitation for mobile manipulators [4], [17]. For instance, Fu et al. [4] proposed a reinforcement learning framework to train a single, unified policy for the whole-body control of a legged manipulator, enabling coordinated and dynamic movements for both locomotion and manipulation tasks. Liu et al. [17] proposed a two-level policy for legged mobile manipulators, trained in simulation and transferred to the real world. The low-level policy controls all degrees of freedom to track body and end-effector commands, while the high-level policy uses visual inputs to generate these commands, enabling the robot to autonomously pick up a variety of objects in different environments.

Traditional mobile manipulation typically involves grasping lightweight objects. The effectiveness of these methods for larger or heavier objects remains an open question. Consequently, non-prehensile pushing motion control has gained increasing research interest. Traditionally, non-prehensile pushing motion control has relied on model-based approaches [5], [6]. Hogan et al. [5] used a combination of integer programming and machine learning to handle the complex dynamics and hybrid nature of contact, enabling an industrial robotic arm to track a pre-planned trajectory while reasoning about multiple contact modes in real-time. Moura et al. [6] presented a trajectory optimization method that uses a Mathematical Program with Complementarity Constraints (MPCC) to handle both sticking and sliding contact modes for planar non-prehensile manipulation. These methods, however, are highly dependent on accurate robot and environmental dynamic models.

Recently, with the success of reinforcement learning in legged robotics, several learning-based pushing methods have been proposed [7], [8]. Jeon et al. [7] presented a learning-based, hierarchical control system that enables a quadrupedal robot to perform whole-body manipulation of large and heavy objects. By using a deep latent variable embedding, the system implicitly learns an object's properties from past interactions, allowing the robot to successfully push, pull, or carry objects that are too big or heavy to grasp. Dadiotis et al. [8] developed a learning-based controller for a mobile manipulator to perform dynamic, non-prehensile object pushing to a goal. Using a constrained reinforcement learning framework, the policy learns to handle unknown objects of various masses, sizes, and shapes, dynamically switching contact points and pushing directions to move and reorient the object while preventing it from toppling.

In the aforementioned studies, the robot operates in an obstacle-free environment. In this work, we address a more challenging scenario where the robot must navigate through a cluttered environment while performing object transportation.

B. Collision Avoidance for Quadruped Robots

Collision avoidance is a core function in the control of quadrupedal robots. Model-based methods, such as MPC, have been utilized to handle collision avoidance constraints [18], [19]. However, these model-based approaches suffer from a lack of robustness due to their strong reliance on model assumptions.

Recently, model-free methods have demonstrated significant success in achieving robust locomotion for quadrupedal robots, leading to their application in collision-avoidance locomotion control [20]. For instance, He et al. [11] introduced a learning-based control framework that enables quadrupedal robots to perform agile, high-speed locomotion while ensuring collision avoidance. The system uses a depth camera to train a control-theoretic reach-avoid value network to switch between an agile policy for efficient movement and a recovery policy for preventing collisions, allowing for safe navigation in cluttered environments with static and dynamic obstacles. Wang et al. [10] proposed an end-to-end reinforcement learning framework that allows legged robots to perform agile, collision-free locomotion by directly processing raw, spatio-temporal LiDAR point clouds. This approach enables the robot to learn robust omnidirectional avoidance behaviors against both static and dynamic obstacles in complex 3D environments.

A major drawback of these learning-based methods is their high computational demand and long training times,

as they often require processing raw sensor inputs (e.g., LiDAR, depth cameras). Furthermore, these approaches typically involve multi-stage training pipelines, which further prolong the training process. In this paper, we propose an alternative approach. Instead of using raw sensor inputs, we utilize a safe corridor as the observation, enabling a more efficient collision avoidance strategy based on constrained reinforcement learning.

III. Main Method

This section outlines the preliminaries and applied framework in robotic training and deployment. To seamlessly bridge the gap between simulation and reality, with consideration of physical limitations and unseen distribution in observed data, the framework is characterized as follows:

1) Sequential decision-making formulations.
2) Carefully designed observations and task-specified reward functions.
3) Deep RL framework with Constrained Proximal Policy Optimization (CPPO)
4) Domain randomization on observed data and environment configuration parameters in training.

Here, we primarily utilize navigation reward functions in the context of pushing object, where we navigate the legged robot to a dynamic goal and the object to a static goal, in a constrained environment.

A. Learning Framework

We formulate collision-free object pushing as an MDP and train a stochastic policy using constrained PPO (CPPO). We adopt an asymmetric actor-critic: the actor receives only deployable observations, while the critic additionally uses privileged state information during training to improve value estimation. Safety and physical limits are encoded as costs (Table VI) and handled via Constraints-as-Terminations (CaT) [21], which stochastically terminates rollouts upon constraint violations to encourage constraint satisfaction without manual penalty-weight tuning. Training details are summarized in Table X.

The actor network is an MLP with hidden layers [512, 256, 128] (ELU activations) producing action means; the critic is an MLP with hidden layers [512, 256, 128] (ELU) predicting state value. The scan encoder uses [128, 64, 32] (ELU). Observation inputs are normalized online via an empirical normalization (running mean/std); other MLPs are standard linear layers with ELU nonlinearity.

B. Observation and Action Space

The observation could be structured as follows:

- Common locomotion observations.
- Common position-based navigation observations.
- Task-specific safe corridor observations.

The observations shown in table I encompassed the necessary information for an actor to derive actions in deployment, and are made available to both the actor and critic networks.

To account for the temporal dynamic, the full observation to actor includes 10 historical observations.

TABLE I
Actor Observation Components and Scaling Factors

Observation Component	Notation	Scale	Dim
Robot Angular Velocity	ω_{r}	0.25	3
Robot Orientation (x-axis)	$\phi_{x,\mathrm{r}}$	2.0	3
Sign-Invariant Quaternion	$w \cdot (x \;\; y \;\; z)$	2.0	3
Front Feet Positions (L, R)	$\mathbf{p}_{\mathrm{FL}}, \mathbf{p}_{\mathrm{FR}}$	1.0	4
Projected Gravity	$\mathbf{g}_{\mathrm{proj,\,r}}$	5.0	3
Motor angle	$\mathbf{q} - \mathbf{q}_0$	1.0	12
Motor Velocity	$\dot{\mathbf{q}}$	0.05	12
Last Action	$\mathbf{a}_{t-1}$	0.25	12
Distance Robot-Object	$\lVert\mathbf{p}_{\mathrm{obj}} - \mathbf{p}_{\mathrm{r}}\rVert$	2.0	1
Distance Robot-Goal	$\lVert\mathbf{p}_{\mathrm{goal}} - \mathbf{p}_{\mathrm{r}}\rVert$	1.0	1
Distance Object-Goal	$\lVert\mathbf{p}_{\mathrm{goal}} - \mathbf{p}_{\mathrm{obj}}\rVert$	1.0	1
Object Position	$\frac{\mathbf{p}_{\mathrm{obj}} - \mathbf{p}_{\mathrm{r}}}{\lVert\mathbf{p}_{\mathrm{obj}} - \mathbf{p}_{\mathrm{r}}\rVert}$	5.0	2
Goal Position	$\frac{\mathbf{p}_{\mathrm{goal}} - \mathbf{p}_{\mathrm{r}}}{\lVert\mathbf{p}_{\mathrm{goal}} - \mathbf{p}_{\mathrm{r}}\rVert}$	5.0	2
Goal Position (Obj frame)	$\frac{\mathbf{p}_{\mathrm{goal}} - \mathbf{p}_{\mathrm{obj}}}{\lVert\mathbf{p}_{\mathrm{goal}} - \mathbf{p}_{\mathrm{obj}}\rVert}$	5.0	2
Corridor Vertex Positions	$v_{\mathcal{C}}$	*masked	8

At each command resampling step we construct an oriented rectangular corridor $\mathcal{C} \subset \mathbb{R}^2$ aligned with the current pushing direction $\hat{d}$ (object→goal). We place an intermediate sub-goal $\mathbf{p}_{\mathrm{s}} = \mathbf{p}_{\mathrm{obj}} + \alpha\,(\mathbf{p}_{\mathrm{goal}} - \mathbf{p}_{\mathrm{obj}})$, where $\mathbf{p}_{\mathrm{obj}}$, $\mathbf{p}_{\mathrm{goal}}$ are the object and goal positions and $\alpha \sim \mathcal{U}[0.5 - \zeta,\, 1]$ with curriculum parameter $\zeta \in \{0, 0.25, 0.4\}$ for achieving $\{20\%-, 20\% \sim 30\%, 30\%+\}$ of r_1 in Table V respectively. The three reference points $(\mathbf{p}_{\mathrm{r}}, \mathbf{p}_{\mathrm{obj}}, \mathbf{p}_{\mathrm{s}})$ (robot, object, sub-goal) are projected onto $\hat{d}$ and its perpendicular $\hat{d}^{\perp}$; the axis-aligned bounding box of these projections is expanded by a randomized margin $m \sim \mathcal{U}[m_{\min}, m_{\max}]$ to form the rectangle whose four vertices define $\mathcal{C}$. As training progresses, ζ is annealed downward following a curriculum: smaller ζ permits sub-goals closer to the box, which reduces the expected longitudinal extent of $\mathcal{C}$ along $\hat{d}$. The reduced distance between the corridor boundary and the box demands higher precision, as any angular error or overshoot triggers both reward penalties and safety-constraint violations. Simultaneously, the curriculum introduces greater variation in the margin range, thereby enhancing the generalization of the model across diverse corridor geometries. To this end, in later Section IV-A.2, we compare the success rate of a model trained with the corridor observation (*masked $= 1$) and one did not (*masked $= 0$).

We also provide the critic network with privileged observations in the training stage. These observations (shown in table II) are unavailable to the actor network as they are usually unobtainable in deployment.

The actions are given as relative angles to the default motor position. The actuation is implemented by a Proportional-Derivative (PD) controller. It takes hyper-parameters k_{p}, k_{d} as the proportional and derivative gains, respectively.

C. Reward and Cost Functions

Following the observation structure, the reward and cost functions present a similar structure,

- Common locomotion rewards.

TABLE II

CRITIC OBSERVATION COMPONENTS AND SCALING FACTORS

Observation Component	Notation	Scale	Dim
Robot Linear Velocity	$\mathbf{v}_{\text{r}}$	2.0	3
Object Linear Velocity	$\mathbf{v}_{\text{obj}}$	2.0	3
Object Angular Velocity	ω_{obj}	0.25	3
Object Orientation (x-axis)	$\phi_{x,\text{object}}$	2.0	3
Orientation Alignment	$\phi_{x,\text{object}}{}^{\top}\phi_{x,\text{r}}$	2.0	1
Contact States	$\mathbb{I}(\text{contact})$	0.05	12
Object Mass	m_{obj}	2.0	1
Object Center of Mass	CoM_{obj}	1.0	3
Object Dimensions	l_{obj}	1.0	3
Corridor Edge Dist (r/o/g)	d_{edge}	1.0	12
Corridor Edge Relative Pos (r/o/g)	$\mathbf{d}_{\text{edge}}$	5.0	24
Robot Base Height	$\mathbf{p}_{z,\text{r}}$	2.0	1

- Distance-based navigation rewards.
- Binary safe corridor costs and other physical limitations.

1) Common Locomotion: To learn the specific task of object pushing in a corridor, the model has to first learn to take stable control in different situations. To this end, we include the following reward functions that are trivial for robotic learning.

TABLE III

LOCOMOTION REWARD FUNCTIONS

Reward Description	Notation
Stumble	$r_{\text{stumble\_up}} = -\mathbb{I}(\text{stumble})$
Collision	$r_{\text{collision\_up}} = -\mathbb{I}(\text{collision})$
Has Contact	$r_{\text{has\_contact}} = \sum_{i\in\text{feet}} \mathbb{I}(\text{contact}_i)$
Base Height	$r_{\text{base\_height\_up}} = -(h_{\text{r}} - h_{\text{target}})^2$
Action Rate	$r_{\text{a}} = -\|\mathbf{a}_t - \mathbf{a}_{t-1}\|^2$
Action Smoothness	$r_{\text{as}} = -\|\mathbf{a}_t - 2\mathbf{a}_{t-1} + \mathbf{a}_{t-2}\|^2$
Upward Orientation	$r_{\text{upward}} = \mathbf{z}_{\text{base}} \cdot \mathbf{z}_{\text{world}}$
Foot Clearance	$r_{\text{foot\_clearance\_up}} = -\sum_{i\in\text{feet}}(h_i - h_{\text{target}})^2$

The rewards shown in table III aim to maintain target robot height and orientation to prevent falling, and stabilize actions by penalty on changing actions.

2) Navigation: We simplify the object pushing task as two simultaneously occurring navigation reward processes:

a. Robot-Obj navigation with orientation alignment.

b. Obj-Goal navigation with orientation alignment.

In this work, the orientation of the object is defined as the direction of velocity, instead of yaw from its initial position. Thus, the manipulation did not consider the yaw as the goal, which poses one of the limitations of this work to practice.

The rewards shown in table IV guide the robots to push the objects. Here, only r_1, r_3 are directly related navigation reward that minimize the distances of object to goal and robot to object. Other rewards are auxiliary for pushing task, that align the motion of robot and object to be in the direction of object to goal while maintaining the robot forward direction perpendicular to the motion, and hence learning to use body side for pushing. Thus, it comes to another ablation study in Section IV-A.2 that we compare the success rate of a model trained with full rewards ($r_1 \sim r_{11}$) and one only with the basic navigation reward (r_1, r_3). These reward terms contribute to the final reward by take a weighted sum, where the scales are outlined in table V.

TABLE IV

NAVIGATION REWARD FUNCTIONS

Reward	Notation
$\text{Navigation}_{\text{obj}}$	$r_1 = \exp(-\|\mathbf{p}_{\text{goal}} - \mathbf{p}_{\text{obj}}\|)$
$\text{Motion}_{\text{obj}}$	$r_2 = \exp\left(2 \cdot \frac{(\mathbf{p}_{\text{goal}} - \mathbf{p}_{\text{obj}})^{\top}}{\|\mathbf{p}_{\text{goal}} - \mathbf{p}_{\text{obj}}\|} \frac{\mathbf{v}_{\text{obj}}}{\|\mathbf{v}_{\text{obj}}\|} - 2\right)$
$\text{Navigation}_{\text{r}}$	$r_3 = \exp(-\|\mathbf{p}_{\text{obj}} - \mathbf{p}_{\text{r}}\|)$
Motion_{r}	$r_4 = \exp\left(2 \cdot \frac{(\mathbf{p}_{\text{goal}} - \mathbf{p}_{\text{obj}})^{\top}}{\|\mathbf{p}_{\text{goal}} - \mathbf{p}_{\text{obj}}\|} \frac{\mathbf{v}_{\text{r}}}{\|\mathbf{v}_{\text{r}}\|} - 2\right)$
Motion 1	$r_5 = \exp\left(-2\left\|\phi_{x,\text{r}}{}^{\top}\frac{\mathbf{v}_{\text{obj}}}{\|\mathbf{v}_{\text{obj}}\|}\right\|\right)$
Motion 2	$r_6 = \exp\left(-2\left\|\phi_{x,\text{r}}{}^{\top}\frac{\mathbf{v}_{\text{r}}}{\|\mathbf{v}_{\text{r}}\|}\right\|\right)$
Motion 3	$r_7 = \exp\left(2 \cdot \frac{\mathbf{v}_{\text{obj}}}{\|\mathbf{v}_{\text{obj}}\|}{}^{\top}\frac{\mathbf{v}_{\text{r}}}{\|\mathbf{v}_{\text{r}}\|} - 2\right)$
Pos 1	$r_8 = \exp\left(-2\left\|\phi_{x,\text{r}}{}^{\top}\frac{\mathbf{p}_{\text{goal}} - \mathbf{p}_{\text{obj}}}{\|\mathbf{p}_{\text{goal}} - \mathbf{p}_{\text{obj}}\|}\right\|\right)$
Pos 2	$r_9 = \exp\left(2 \cdot \frac{\mathbf{p}_{\text{goal}} - \mathbf{p}_{\text{r}}}{\|\mathbf{p}_{\text{goal}} - \mathbf{p}_{\text{r}}\|}{}^{\top}\frac{\mathbf{p}_{\text{goal}} - \mathbf{p}_{\text{obj}}}{\|\mathbf{p}_{\text{goal}} - \mathbf{p}_{\text{obj}}\|} - 2\right)$
Pos 3	$r_{10} = \exp\left(2 \cdot \frac{\mathbf{p}_{\text{obj}} - \mathbf{p}_{\text{r}}}{\|\mathbf{p}_{\text{obj}} - \mathbf{p}_{\text{r}}\|}{}^{\top}\frac{\mathbf{p}_{\text{goal}} - \mathbf{p}_{\text{obj}}}{\|\mathbf{p}_{\text{goal}} - \mathbf{p}_{\text{obj}}\|} - 2\right)$
Pos 4	$r_{11} = \exp\left(2 \cdot \frac{\mathbf{p}_{\text{obj}} - \mathbf{p}_{\text{r}}}{\|\mathbf{p}_{\text{obj}} - \mathbf{p}_{\text{r}}\|}{}^{\top}\frac{\mathbf{p}_{\text{goal}} - \mathbf{p}_{\text{r}}}{\|\mathbf{p}_{\text{goal}} - \mathbf{p}_{\text{r}}\|} - 2\right)$

TABLE V

SCALES ASSOCIATED WITH EACH NUMBERED REWARD FUNCTION

Reward	r_1	r_2	r_3	r_4	$r_5 \sim r_{11}$
Scale	5.0	1.0	2.0	1.0	2.0

3) Corridor and Others: We implemented collision avoidance by constraining our models to manipulate the robot body and object only existing in the corridor area. We assume that the safe corridor is an obstacle-free terrain area. This could be provided by high-level planner algorithms [22].

In the context of safety concerns in real-world application, it naturally takes a binary cost of inside/outside of the corridor. This implementation enforces the model to strictly avoid collisions. To address other physical limitations, such as output power, rotational velocity, and angles of motors, we define costs as summation of violation from target threshold values, respectively [23]. These constraints are defined in table VI

TABLE VI

COST FUNCTIONS

Cost Description	Definition
Safe Corridor Cost	$c_{\text{safe\_corridor}} = \mathbb{I}(\mathbf{p}_{\text{r}} \notin \mathcal{C})$
Obj Safe Corridor Cost	$c_{\text{obj\_safe\_corridor}} = \mathbb{I}(\mathbf{p}_{\text{obj}} \notin \mathcal{C})$
Motor Angle Cost	$c_{\text{pos\_limit}} = \sum_i \mathbb{I}(\|q_i\| > q_{\max})$
Motor Vel Cost	$c_{\text{vel\_limit}} = \sum_i \mathbb{I}(\|\dot{q}_i\| > \dot{q}_{\max})$
Motor Power Cost	$c_{\text{torque\_limit}} = \sum_i \mathbb{I}(\|\tau_i\| > \tau_{\max})$

Each edge of $\mathcal{C}$ contributes a signed distance and an outward unit direction to the observation (for both robot and object). We utilize half-plane method [22] that the cross-product of direction from the point to edges with the edges itself directly addresses this binary logic.

TABLE VII

SCALES AND THRESHOLDS ASSOCIATED WITH EACH COST FUNCTION

Constraint	Scale	Activation Threshold	Targeted Value
$c_{\text{safe\_corridor}}$	1.0	0.0	–
$c_{\text{box\_safe\_corridor}}$	2.0	0.0	–
$c_{\text{pos\_limit}}$	1.0	0.0	0.9
$c_{\text{vel\_limit}}$	0.1	0.0	1.0
$c_{\text{torque\_limit}}$	1.0	0.0	1.0

D. Domain Randomization

To enhance the robustness and generalization of the learned policy, domain randomization was utilized during simulation to improve sim-to-real adaptability. The model comprises two main components: the object and the robot, with domain randomization applied differently to each component, as shown in table VIII.

TABLE VIII

DOMAIN RANDOMIZATION RANGES

Parameters	Min	Max	Unit
Obj Dimensions	0.2	0.6	m
Obj Mass	2	10	kg
Obj CoM Displacement	-0.05	0.05	m
Body Mass Deviation	-1	3	kg
Motor Strength Factor	0.9	1.1	-
Friction Factor	0.2	1.25	-
Gravity Shifts	-1	1	$\text{m} \cdot \text{s}^{-2}$
Motor Offset	-0.02	0.02	Rad
Restitution	0	1	-
CoM Displacement	-0.05	0.05	m
k_{p}	36	44	-
k_{d}	0.9	1.1	-
Actuation Time Lag	0	0.015	s
Disturbance Force	0	30	N
Push Velocity	0	3	$\text{m} \cdot \text{s}^{-1}$
Corridor Size	0.2	0.6	m

To simulate uncertain environments where unexpected external forces may act on the robot, we introduce periodic external actuation to the system at 8-second intervals. We mimic a push force on the robot by inserting an external velocity in a predefined range of velocities, sampled from a uniform random distribution (as detailed in table VIII).

To enhance robustness against perturbations in sensory data obtained from real-world observations, uniform random noise is added to the observations provided to the actor. The magnitudes of the noise are specified in table IX.

IV. EXPERIMENTS

A. Simulation

1) Training Details: We summarize the NP3O/PPO training hyperparameters and rollout setup in Table X.

The main contribution of the work lies in the application of safe corridor representation to provide sufficient information for collision-free navigation. Although navigation functionality is not the novelty of this work, it is challenging to navigate the robot and object to the target positions without collision. To empirically validate our framework, we evaluate

TABLE IX

ACTOR OBSERVATION NOISE SCALES

Observation Component	Noise Scale
Robot Angular Velocity	0.2
Robot Orientation (x-axis)	0.01
Front Feet Positions (L, R)	0
Projected Gravity	0.05
Motor angle	0.01
Motor Velocity	1.5
Last Action	0
Distance Robot-Object	0.05
Distance Robot-Goal	0.05
Distance Object-Goal	0.05
Object Position	0.05
Goal Position	0.05
Goal Position (Obj frame)	0.05
Corridor Vertex Positions	0.05

TABLE X

TRAINING HYPERPARAMETERS

Parameter	Value
PPO (NP3O)	
Discount γ	0.99
GAE λ	0.95
Clip range ϵ	0.2
Entropy coefficient	0.01
Value loss coefficient	1.0
Cost value loss coefficient	1.0
Cost violation loss coefficient	1.0
Num. learning epochs	5
Num. mini-batches	4
Max gradient norm	1.0
Optimization	
Optimizer	Adam
Learning rate	1×10^{-3}
Weight decay	0
LR schedule	Adaptive KL
Target KL d_{targ}	0.01
Rollout	
Num. parallel envs	4096
Rollout length	24 steps
Samples per update	98 304
Mini-batch size	24 576
Init. action noise σ	1.0
Simulation	
Physics Δt	0.005 s
Control decimation	4
Policy Δt	0.02 s (50 Hz)
Episode length	20 s (1000 steps)

the trained model in the following metric under different terrain configurations:

- First arrival time (lower better).
- Arrival rate (higher better).
- Success rate (higher better).

The success rate is defined as reaching goal position and remaining in the corridor for the robot and object throughout the testing episode.

2) Ablation Study: We perform the evaluation with different configurations. In general, we configure a mixture of random sloped and rough terrain with proportion of 0.1/0.9 respectively. The objects are randomly initialized in the distance 1.5m from the robot, and the target positions are randomly initialized with distances from the objects,

TABLE XI
EVALUATION METRICS IN 216 TRIALS

Metric	Size	Level 1			Level 2		
		Full	**NoC**	**NoA**	**Full**	**NoC**	**NoA**
Time (s)	0.3	4.83	5.94	5.79	4.40	5.01	4.54
	0.4	4.87	5.62	4.03	4.53	5.25	3.82
	0.5	4.92	5.67	-	4.56	5.30	4.06
Arrival (%)	0.3	93.0	78.7	2.31	88.0	68.1	1.85
	0.4	91.2	77.8	1.85	85.2	73.6	2.31
	0.5	94.0	78.2	0	91.2	70.8	0.93
Success (%)	0.3	87.0	72.2	0.93	81.9	19.0	1.39
	0.4	85.2	76.9	1.85	78.7	23.6	1.39
	0.5	90.3	77.8	0	84.7	37.5	0.93

where the distances are uniformly distributed in $[1.0,\ 4.0]$m. We generate a randomly oriented and sized rectangle that includes the initial position of the robot and object in each trial. This corridor is further extended by a size of $\{0.3,\ 0.4,\ 0.5\}$m, denoted as Size in table XI. Particularly, we define two levels to evaluate the models. The first level, denoted as Level 1, takes 95% of the goal positions included in the safe corridor. The second level, denoted as Level 2, takes 10% of the goal positions included in the safe corridor.

We compare three models trained under the same framework but with different observations and rewards functions. In table XI, we denote the model trained with all observations and rewards as 'Full'. We denote model trained without corridor observations (*masked $= 0$, refer to prior subsequent section) as 'NoC'. We denote model trained without auxiliary reward functions ($r_2,\ r_4 \sim r_{11}$, refer to prior section) as 'NoA'.

Notably, for all models, we trained with same cost functions, thus the models are all expected to follow constraints of staying inside of the corridor.

From table XI, the arrival rate and success rate of 'NoA' are exceptionally low, and hence it may not provide meaningful insight. One could conclude that it is necessary to include the auxiliary reward functions for the object pushing task.

From table XI, in the two levels, 'NoC' attained comparable performances as 'Full' in first arrival time and arrival rate. However, the success rate of 'NoC' is comparable to 'Full' only in Level 1, which includes most goal positions in trials. The success rates of 'NoC' are significantly lower than those of 'Full' in Level 2, while maintaining similar arrival rates. This indicates that the corridor observations will not affect the object pushing task, that it could navigate the object to the given destination, but it takes paths that violate the corridor constraints.

3) Custom Environments: To assist in elaborating the simulation-to-real results, we configured the same terrains used in the deployment to visualize the safe corridor in the simulation. As we took simplified map configuration, the safe corridor vertices in these environments are given as rectangle vertices and is given to the robot along the local goals by a path planner.

In simulation, objects in three custom environments are defined as rigid cubes with length 0.6m and mass 6kg.

The environments are created with default physical property but different customized obstacle positions.

In Fig. 2 (a), 2 (d) and 2 (g), they show the initial position of the robot, object and safe corridor vertices for each terrain. Each terrain is configured with different obstacle positions, and the task for the robot is to push the cube to a given goal and maintain both the robot itself and the cube inside the corridor. The corridor is defined as a rectangle and will be updated upon reaching each local goal. Fig. 2 (b), 2 (e) and 2 (h) show the arrival of goal for each terrain. Fig. 2 (c), 2 (f) and 2 (i) show the full path of robot and cube, and corridor plots for each terrain. In all terrains, the robots successfully pushed the cubes to given destinations while obeying the safe corridor constraints.

These results demonstrated the feasibility of applying safe corridor costs in complex navigation settings, such as object pushing. In the next section, to further validate the effectiveness of our framework, we deploy the model trained with full observations and rewards with the second terrain, as shown in Fig. 2 (d), 2 (e) and 2 (f).

B. Deployments

To adapt to practical scenarios, extensive real-world deployment experiments were conducted using the Unitree Go2 quadruped robot. These experiments focused on the task of object pushing and were performed in a controlled testing environment designed to mimic real-world conditions by strategically placing obstacles. For accurate positioning, the robot was equipped with a Nooploop UWB linktrack system, which provided precise real-time position data for both the robot and the objects involved in the task. We obtain the robot and object's position by assigning a tag to each of them.

Fig. 3 shows the sequence of states in deployment that the robot is pushing a rectangular object from (a) to (e). Fig. 4 shows the sequence of states in simulation that the robot is pushing a cubic object from (a) to (e).

Deployment challenges often arise from sim-to-real gaps, such as unmodeled friction. To address this, we employed rigorous domain randomization and observation noise during training. In our experiments, the controller successfully managed an unseen $0.8\,\text{m} \times 0.25\,\text{m} \times 0.35\,\text{m}$ deformable rectangular object ($6\,\text{kg}$) on a non-smooth, obstructed surface. The success of these trials validates our robust training approach. Furthermore, an average computation time of $0.0098\,\text{s}$ on the Unitree Go2 demonstrates the system's capacity for high-frequency control on lightweight hardware during complex tasks.

V. LIMITATIONS AND FUTURE WORK

It happens to fail in turning corner case that the two consecutive local goal paths being in perpendicular direction. The preliminary explanation for it is that the alignment reward can cause the robot to lose effective pushing contact.

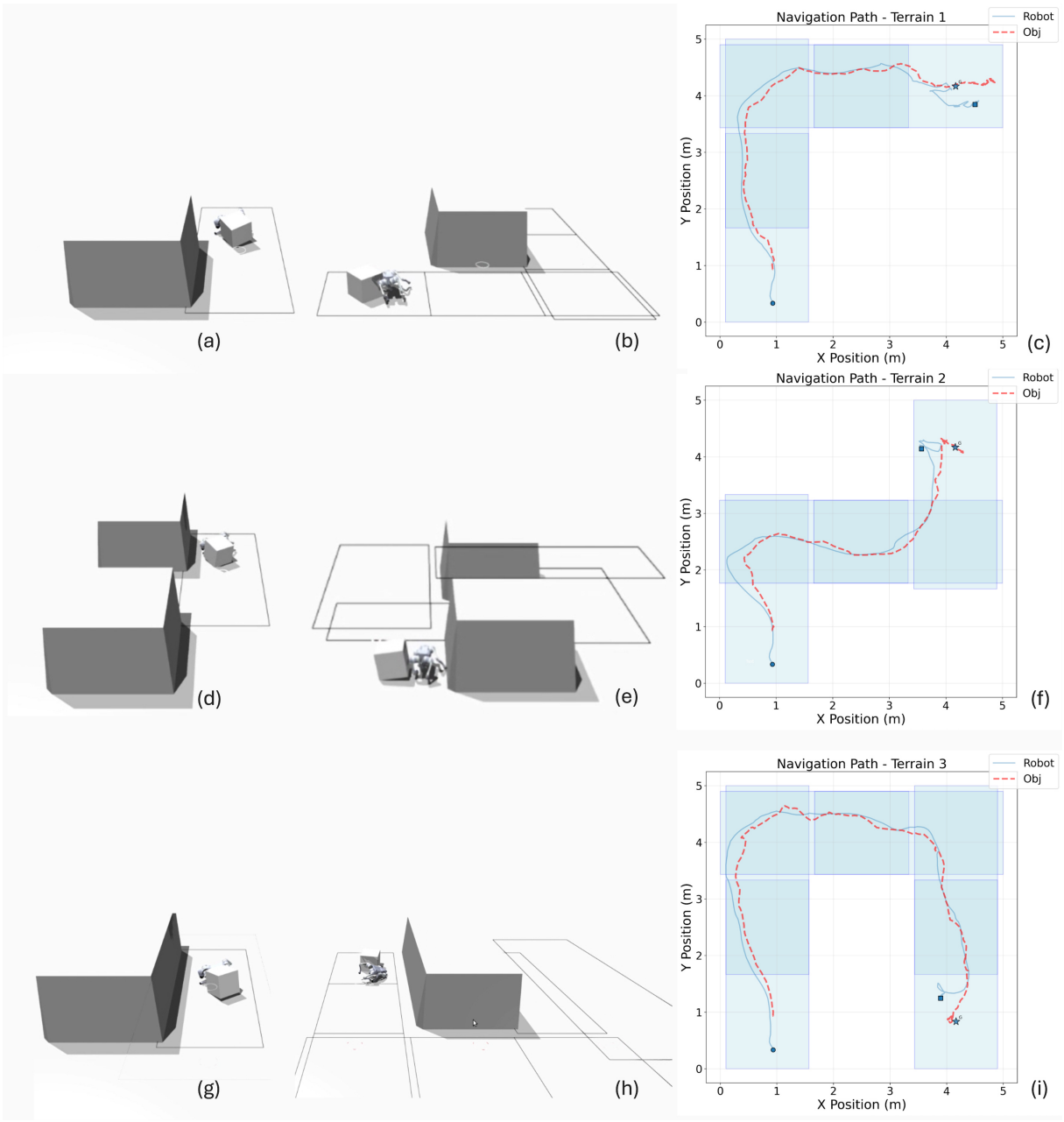
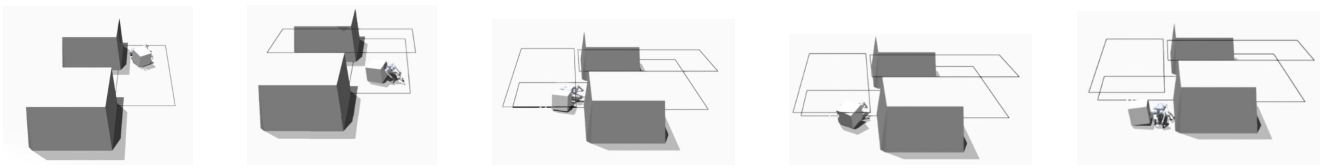


Fig. 2. Simulation Initial (a, d, g)/Goal (b, e, h)/Path (c, f, i) Figures in Different Custom Terrains

(a) (b) (c) (d) (e)

Fig. 3. Sequences of States in Deployment (Ascending order, lateral view)



(a) (b) (c) (d) (e)

Fig. 4. Sequences of States in Simulation (Ascending order, rear-up view)

The current formulation optimizes object translation and uses object velocity as an orientation proxy, the final yaw of objects are not explicitly enforced. We also rely on external UWB for localization and assume rectangular, plannerprovided, obstaclefree corridors, limiting realworld autonomy. Future work includes adding explicit pose objectives (object yaw), perceptiondriven online corridor construction (e.g., LiDAR), onboard state estimation, and broader testing across object shapes, compliance, and friction.

VI. Conclusion

This work presented an RL framework for collision-free quadrupedal object pushing by leveraging safe corridors as a compact, low-dimensional representation of traversable space. By integrating these corridors directly into the observation and reward structures, we achieved efficient, constraint-aware manipulation without high-dimensional sensor processing. Extensive simulation and real-world experiments on a Unitree Go2 demonstrated the framework's robustness against disturbances and its ability to navigate cluttered environments. This approach provides a scalable foundation for future research into complex, non-prehensile mobile manipulation for legged robots.

ACKNOWLEDGMENT

This work was supported by CityUHK Start-Up Grant (Project No. 9610481).

References

[1] S. Xu, L. Zhu, H.-T. Zhang, and C. P. Ho, "Robust convex model predictive control for quadruped locomotion under uncertainties," *IEEE Transactions on Robotics*, vol. 39, no. 6, pp. 4837–4854, 2023.

[2] I. Dadiotis, A. Laurenzi, and N. Tsagarakis, "Trajectory optimization for quadruped mobile manipulators that carry heavy payload," in *2022 IEEE-RAS 21st International Conference on Humanoid Robots (Humanoids)*. IEEE, 2022, pp. 291–298.

[3] M. P. Polverini, A. Laurenzi, E. M. Hoffman, F. Ruscelli, and N. G. Tsagarakis, "Multi-contact heavy object pushing with a centaur-type humanoid robot: Planning and control for a real demonstrator," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 859–866, 2020.

[4] Z. Fu, X. Cheng, and D. Pathak, "Deep whole-body control: learning a unified policy for manipulation and locomotion," in *Conference on Robot Learning*. PMLR, 2023, pp. 138–149.

[5] F. R. Hogan and A. Rodriguez, "Reactive planar non-prehensile manipulation with hybrid model predictive control," *The International Journal of Robotics Research*, vol. 39, no. 7, pp. 755–773, 2020.

[6] J. Moura, T. Stouraitis, and S. Vijayakumar, "Non-prehensile planar manipulation via trajectory optimization with complementarity constraints," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 970–976.

[7] S. Jeon, M. Jung, S. Choi, B. Kim, and J. Hwangbo, "Learning whole-body manipulation for quadrupedal robot," *IEEE Robotics and Automation Letters*, vol. 9, no. 1, pp. 699–706, 2023.

[8] I. Dadiotis, M. Mittal, N. Tsagarakis, and M. Hutter, "Dynamic object goal pushing with mobile manipulators through model-free constrained reinforcement learning," *arXiv preprint arXiv:2502.01546*, 2025.

[9] S. Liu, M. Watterson, K. Mohta, K. Sun, S. Bhattacharya, C. J. Taylor, and V. Kumar, "Planning dynamically feasible trajectories for quadrotors using safe flight corridors in 3-d complex environments," *IEEE Robotics and Automation Letters*, vol. 2, no. 3, pp. 1688–1695, 2017.

[10] 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," *arXiv preprint arXiv:2505.19214*, 2025.

[11] T. He, C. Zhang, W. Xiao, G. He, C. Liu, and G. Shi, "Agile but safe: Learning collision-free high-speed legged locomotion," *arXiv preprint arXiv:2401.17583*, 2024.

[12] X. Cheng, K. Shi, A. Agarwal, and D. Pathak, "Extreme parkour with legged robots," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 11 443–11 450.

[13] J.-P. Sleiman, F. Farshidian, M. V. Minniti, and M. Hutter, "A unified mpc framework for whole-body dynamic locomotion and manipulation," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 4688–4695, 2021.

[14] M. V. Minniti, F. Farshidian, R. Grandia, and M. Hutter, "Whole-body mpc for a dynamically stable mobile manipulator," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3687–3694, 2019.

[15] M. Mittal, D. Hoeller, F. Farshidian, M. Hutter, and A. Garg, "Articulated object interaction in unknown scenes with whole-body mobile manipulation," in *2022 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 2022, pp. 1647–1654.

[16] I. Dadiotis, A. Laurenzi, and N. Tsagarakis, "Whole-body mpc for highly redundant legged manipulators: experimental evaluation with a 37 dof dual-arm quadruped," in *2023 IEEE-RAS 22nd International Conference on Humanoid Robots (Humanoids)*. IEEE, 2023, pp. 1–8.

[17] M. Liu, Z. Chen, X. Cheng, Y. Ji, R.-Z. Qiu, R. Yang, and X. Wang, "Visual whole-body control for legged loco-manipulation," *arXiv preprint arXiv:2403.16967*, 2024.

[18] 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.

[19] I. Ibrahim, F. Farshidian, J. Preisig, P. Franklin, P. Rocco, and M. Hutter, "Whole-body mpc and dynamic occlusion avoidance: A maximum likelihood visibility approach," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 221–227.

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

[21] E. Chane-Sane, P.-A. Leziart, T. Flayols, O. Stasse, P. Souères, and N. Mansard, "Cat: Constraints as terminations for legged locomotion reinforcement learning," in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2024, pp. 13 303–13 310.

[22] Q. Wang, Z. Wang, M. Wang, J. Ji, Z. Han, T. Wu, R. Jin, Y. Gao, C. Xu, and F. Gao, "Fast iterative region inflation for computing large 2-d/3-d convex regions of obstacle-free space," *IEEE Transactions on Robotics*, 2025.

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