Multi-Brain Collaborative Control for Quadruped Robots

Anonymous Author(s)
Affiliation
Address
`email`

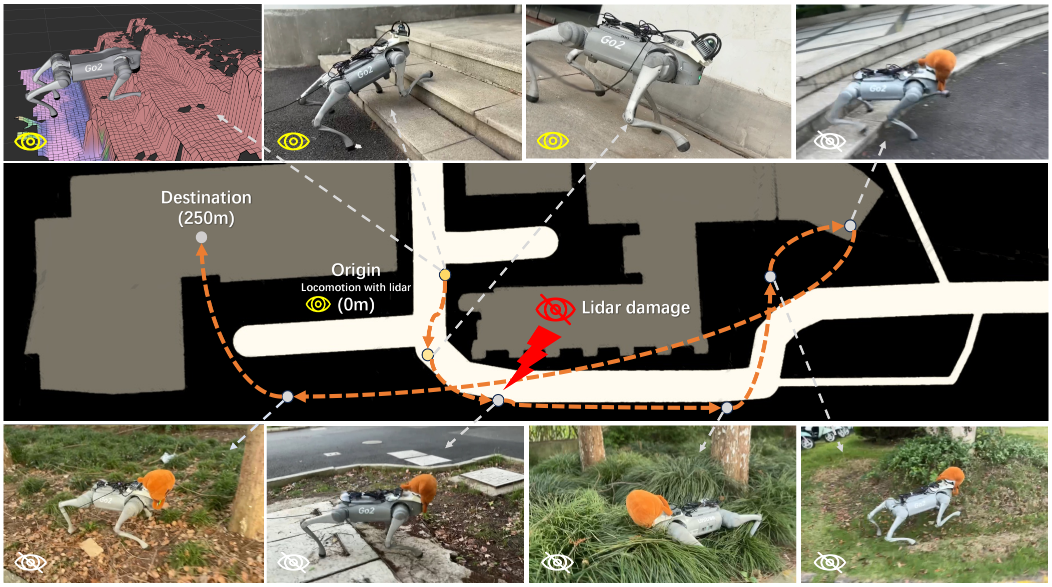

Figure 1: We conducted a long-distance (250m) test on a controller based on multi-brain collaborative. At the beginning of the map, the robot relied on height-map and proprioception to traverse through terrain. During the test, we simulated a scenario where the lidar suddenly malfunctioned (by covering it with a orange bag). The robot did not experience any mode crashes and was still able to handle complex terrains effectively.

Abstract: In the field of locomotion task of quadruped robots, Blind Policy and Perceptive Policy each have their own advantages and limitations. The Blind Policy relies on preset sensor information and algorithms, suitable for known and structured environments, but it lacks adaptability in complex or unknown environments. The Perceptive Policy uses visual sensors to obtain detailed environmental information, allowing it to adapt to complex terrains, but its effectiveness is limited under occluded conditions, especially when perception fails. Unlike the Blind Policy, the Perceptive Policy is not as robust under these conditions. To address these challenges, we propose a Multi-Brain collaborative system that incorporates the concepts of Multi-Agent Reinforcement Learning and introduces collaboration between the Blind Policy and the Perceptive Policy. By applying this multi-policy collaborative model to a quadruped robot, the robot can maintain stable locomotion even when the perceptual system is impaired or observational data is incomplete. Our simulations and real-world experiments demonstrate that this system significantly improves the robot's passability and robustness against perception failures in complex environments, validating the effectiveness of multi-policy collaboration in enhancing robotic motion performance.

Keywords: Quadruped Robots, Perception Fails, Multi-Brain Collaborative

1 Introduction

What happens if a robot suddenly loses its perception? Can it maintain its previous stable motion? In natural environments, the sensory systems of humans and animals can sometimes experience temporary or permanent impairments, such as "dark adaptation" phenomenon when moving from a bright to a dark environment. In these situations, humans and animals can rely on past experiences to immediately switch to a state of motion without sensory input, ensuring safe movement.

For humans, this ability stems from two main sources. First, the human brain has a strong adaptive and memory capacity. When the perceptual system fails for a short period of time, the brain will automatically call upon memories and experiences to compensate for the perceptual deficit. Secondly, the human motor control system has a high degree of redundancy and multisensory integration. For example, when vision fails, the proprioceptive and vestibular systems enhance their role in maintaining balance movement.

In the motion tasks of bipedal and quadrupedal robots, sensory systems may fail due to incomplete information or hardware malfunctions. These robots rely on various sensors to gather environmental data, such as LiDAR, cameras, and ultrasonic sensors. However, the effectiveness of these sensors can be limited in low-light or adverse weather conditions, or they may fail due to physical damage or signal disruptions. Therefore, researching how to maintain stable robot motion under these unfavorable conditions is a challenge in current studies.

In locomotion tasks, blind policies and perceptive policies each have their advantages and limitations [1]. Blind policies rely on sensors and preset algorithms for movement, requiring no visual input [2, 3, 4, 5]. Although they are fast and consume fewer resources, their adaptability in complex or unknown environments is limited, and they have weaker obstacle recognition abilities and generalizability. Perceptive policies use visual sensors to obtain detailed information about the environment, enabling robots to adapt to complex terrains [6, 7, 8]. However, in less than ideal visual conditions or in known and structured environments, perceptive policies may not be as efficient as blind policies. Researching how to effectively merge these two policies to cope with complex and changing environments is an equally challenging research issue.

Addressing the challenges mentioned, this study integrates Multi-Agent Reinforcement Learning (MARL) [9, 10] to propose the concept of Multi-Brain Game Collaboration. We envision a quadruped robot system integrating multiple policies to form a collective "brain" with each policy tailored to different input policies. Specifically, we explore the interaction between a Blind Policy, independent of perceptual input, and a Perceptive Policy that utilizes external information. This model excels in scenarios with incomplete observational data or impaired sensory capabilities, accurately simulating and analyzing the robot's interactions with its environment. This approach enhances decision-making and adaptability in complex environments.

The primary contributions of this research are as follows:

- **A Novel Multi-Brain Game Collaboration System:** This study introduces and successfully implements a multi-brain game collaboration system. In this system, each policy or "brain" independently and collaboratively optimizes decisions for different tasks. This design mimics the division of labor and cooperation in biological neural systems, significantly enhancing decision-making efficiency and precision.

- **Integration of Blind and Perceptive Policies:** The research thoroughly analyzes the combination of Blind Policies with Perceptive Policies. This collaborative policy provides an innovative approach for flexible movement in complex environments.

- **Enhanced Mobility in Complex Environments:** Through the non-zero-sum game [11] between blind and perceptive policies, this policy allows the robot to make accurate and effective motion decisions, even with incomplete information or limited perception.

2 Related work

Multi-Agent Reinforcement Learning In the field of Multi-Agent Reinforcement Learning (MARL), there are generally three learning paradigms: centralized learning, independent learning, and Centralized Training with Decentralized Execution (CTDE) [12]. Among these, CTDE effectively combines the advantages of centralized learning with the flexibility of decentralized execution

MADDPG [13] is a typical representative of the CTDE paradigm, employing an actor-critic framework. However, as an off-policy algorithm, MADDPG requires extensive memory storage to save previous experiences and may not perform as stably in dynamic environments as on-policy algorithms. MATD3 [14], a multi-agent version of TD3, enhances the stability of multi-agent cooperation through double Q-learning and delayed policy updates, but this also increases computational complexity, especially in large-scale multi-agent environments, and is extremely sensitive to hyperparameters, which may require extensive tuning and experimentation in practical applications to achieve optimal performance.

MAPPO [15], for the first time, effectively extends the single-agent PPO algorithm to a multi-agent environment, becoming an on-policy strategy that can handle complex multi-agent collaborations while maintaining the stability and efficiency of policy updates. MAPPO not only retains the advantages of PPO but also successfully addresses the collaboration problems in multi-agent environments. Its application on the SMAC platform demonstrates its high sample efficiency and consistency of policies [16].

Blind Policy & Perceptive Motion Policy In enhancing the adaptability and motion performance of quadruped robots in complex environments, current research explores three primary policies. The first policy, termed the blind policy, relies on the robot's proprioceptive history, primarily utilizing forelimb probing, to estimate terrain [3, 5, 2]. This policy faces limitations in complex or unknown environments due to its weak obstacle recognition and generalization capabilities. The second policy uses a holistic control approach based on external sensory inputs to gather environmental details, helping the robot plan movements and navigate complex terrains [17, 18, 19, 20, 21]. However, this often involves isolated end-to-end network architectures without testing for sensor reliability. The third, a composite policy [22, 23] integrates blind and visual policies into a synergistic mechanism, quickly adapting to sudden failures in external perception systems.

In sim2real applications for vision-based motion controllers using reinforcement learning, two main approaches are prevalent: end-to-end training with depth or RGB images, effective in quadrupedal robots, and using elevation maps [24, 25] or height scans from a Global Reference Frame. The latter provides precise terrain information, enhancing adaptability and performance in complex environments. Compared to traditional images, elevation maps mitigate poor visual conditions, improving navigation and decision-making [26, 27]. Furthermore, LiDAR offers high precision and reliability under low light or visual occlusion, with its point cloud data converted into elevation maps providing rich 3D terrain details, crucial for obstacle detection and terrain analysis.

To the best of the authors' knowledge, there has been no research combining multi-agent reinforcement learning algorithms such as MAPPO to achieve non-zero-sum games between blind policies and perceptive policies. Our approach can accurately simulate and analyze the complex interactions between the robot and the environment, even under conditions of incomplete observation data or sensory loss, thereby enhancing the robot's motion performance in various environments.

3 Method

3.1 Task Formulation

In the locomotion task of a quadruped robot, we define a process that combines a blind policy and an external perception-based policy to handle complex environments. Specifically, the quadruped robot can flexibly navigate various obstacles such as highlands, gaps, obstacles, and stairs when

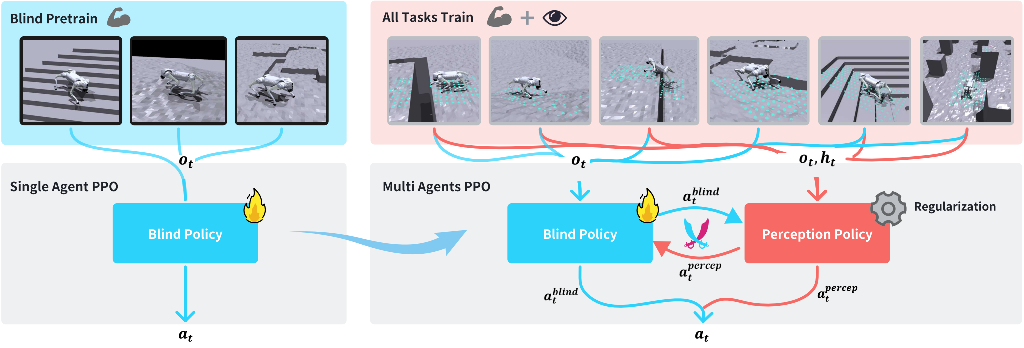

Figure 2: Two-stage multi-brain game collaborative training framework.

external perception (e.g., LiDAR elevation maps) is functioning properly. However, when external perception suddenly fails, the quadruped robot, although unable to navigate terrains such as gaps, should still retain the capability to traverse complex terrains like stairs and ramps.

We have designed a two-stage training approach, as illustrated in Figure 2. In the first stage, a training mode without external perception is used, involving only a blind policy. In the second stage, a multi-agent approach is employed, incorporating external perception and simultaneously training both the blind policy and a perceptive policy with perception capabilities. The collaboration between these two policies is guided by a terrain reconstruction error regularization term. This ensures that our robot can effectively traverse terrains both with and without perception.

3.2 Base Set

Theorem In complex 3D environments, quadruped robots must maintain stable navigation and locomotion even when external perception capabilities fail. To achieve this objective, we describe the locomotion problem of quadruped robots using a Partially Observable Markov Decision Process (POMDP) [28, 29].

The POMDP framework effectively models decision-making scenarios where information is incomplete, defining key elements such as states, actions, observations, and rewards. In this model, the environment at time step t is represented by a complete state x_t. Based on the agent's policy, an action a_t is performed, resulting in a state transition to x_{t+1} with a probability $P(x_{t+1} \mid x_t, a_t)$. The agent then receives a reward r_t and a partial observation o_{t+1}. The aim of reinforcement learning here is to identify a policy π that maximizes the expected discounted sum of future rewards:

$$J(\pi) = \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t r_t \right]$$

Action Space & State Space The action spaces for the blind policy and perceptive policy are respectively $a_t^{blind} \in \mathbb{R}^{12}$ and $a_t^{percep} \in \mathbb{R}^{12}$, representing the offset from the default position for each joint. The critic networks for both policies observe the global state $s_t^{critic} = [o_t, v_t, e_t, h_t, a_t^{percep}, a_t^{blind}]^T$, which includes proprioceptive observations o_t, estimated linear velocities $\hat{v}_t$, and latent variables e_t such as body mass, center of mass position, friction coefficients, and motor strength. These global observations are crucial for the second phase of training, helping the critic network make balanced decisions during the interactions between the two policies and preventing training collapse due to excessive competition.

For the actor networks, the state space for the blind policy includes proprioceptive observations o_t, estimated linear velocity $\hat{v}_t$, and latent variables e_t. Additionally, aligning with the multi-agent game theory approach, the state space for the blind policy also incorporates the output from the Perceptive Policy a_t^{percep}, expressed as $s_t^{blind} = [o_t, \hat{v}_t, e_t, a_t^{percep}]^T$. Similarly, the state space for the Perceptive Policy is $s_t^{percep} = [o_t, h_t, a_t^{blind}]^T$, where h_t represents the local elevation map centered around the robot.

During the first phase of training for the Blind policy, we employed the Regularized Online Adaptation (ROA) method [30] to estimate the explicit observations $\hat{v}_t$ and the latent variables e_t. In this phase, a_t^{percep} was set to zero. In the second phase of training, the final action $a_t = a_t^{percep} + a_t^{blind}$.

3.3 First Stage

In the first stage of training, we primarily developed a proprioceptive motion system for the quadruped robot, aimed at enabling the robot to traverse various complex terrains such as uneven slopes, stairs, and discrete terrains without direct visual or elevation map input to the policy. During this phase, the output action item for the perception policy was set as a 12-dimensional zero vector, ensuring that the blind policy operates without interference from other agents' outputs. Our blind policy, inspired by the ROA training framework, uses only current proprioceptive inputs and the action outputs of other agents at time t to estimate the robot's real-time privileged information, such as speed. This approach does not require a long temporal window, allowing the network to estimate the robot's state based solely on its current status and actions. Additionally, the training utilized an asymmetric Actor-Critic structure to better evaluate the quality of the actions output by the Actor.

For the robot's elevation map, we trained a Variational Autoencoder (VAE) model primarily to memorize the terrains encountered by the blind policy and to compute regularization terms for action constraints in the subsequent training phase.

3.4 Second Stage

In the second stage of learning, we introduced a multi-agent learning approach, utilizing a non-zero-sum game strategy to optimize the external perception controllers for quadruped robots. Unlike traditional single-policy approaches such as parkour, this method allows for adaptation when one controller fails, as other controllers can detect and adjust their actions, enhancing the system's robustness. Within the multi-agent framework, the gradients for each controller are updated independently, facilitating task separation and allowing each controller to focus on specific tasks, thereby improving the overall adaptability of the system. Additionally, this model supports "hot-swapping" of the perception system, enabling the robot to move based on sensory data when available and to continue proprioceptive movement without malfunction when perception is unexpectedly lost.

The primary implementation policy is as follows: initially, load the pre-trained model of the single-agent blind policy and activate these models in the second phase to utilize the actual outputs from the perceptive policy. Inputs to the perceptive policy include proprioceptive data, outputs from the blind policy, and elevation map information, primarily adjusted for terrain. The robot's final actions are a combination of perceptive and blind actions. This framework ensures that during training, the perceptive and blind policies interact and collaborate to optimize movement. All networks use the CTDE approach with MAPPO [15] updates, where each agent's Critic network shares all environmental information, including the inputs and outputs of other agents, during training, while each operates independently during execution. The loss calculations and updates for the blind policy remain as in the first phase, while the perceptive policy's loss includes surrogate loss, value loss, entropy loss, and a Reconstruction Error Regularizer. The purpose of the regularization term is to encourage the Percep policy to minimize actions when encountering terrain similar to those handled by the blind policy, promoting cooperation between the two policies and reducing excessive competition.

3.5 VAE & Perception Cooperation Constraint Regularization

In the first stage, we primarily trained the quadruped robot to navigate slopes, steps, and discrete obstacles without relying on external perception. These terrains were chosen because they enable the robot to learn fundamental locomotion skills and develop robust capabilities. Steps, in particular, significantly improve the robot's ability to lift its legs and react to tripping, thereby enhancing overall mobility. We believe these terrains exemplify the types of environments a robot can navigate without perception in real-world scenarios. We designed a Variational Autoencoder (VAE) to encode and decode these features, with the VAE being updated using MSE and KL divergence during the first stage.

In the second stage, we introduced more challenging terrains, such as highlands, gaps, and pillars, which are difficult for the robot to navigate using only the Blind Policy trained in the first stage. Therefore, it must rely on the Percep policy with external perception input for compensation. However, the complexity introduced by Multi-Agent systems can lead to policies converging to local optima, with the blind policy and Percep policy potentially competing against each other, hindering coordinated control. To address this, we introduced a perception cooperation constraint regularization term based on elevation maps. This term helps ensure that if the current elevation map reconstruction error, as produced by the VAE, is below a threshold, indicating familiarity with the terrain, the regularization term increases with the Percep policy's output, limiting its action. If the reconstruction error exceeds the threshold, indicating unfamiliar terrain, the regularization term is set to zero, encouraging the Percep policy to compensate.

Specifically, in the second stage, the robot's current elevation map h_{ij} is input into the VAE, which reconstructs the elevation map $\hat{h}_{ij}$. The reconstruction error is then calculated as $E_i = \frac{1}{n}\sum_{j=1}^{n}(\hat{h}_{ij} - h_{ij})^2$, where i represents the i-th sample in the batch, j represents the index of the dimensions of the elevation map and action, and n represents the dimension of the elevation map. Based on the reconstruction error and the threshold, we define the penalty factor:

$$\mathbb{I}_i = \begin{cases} 0 & \text{if } E_i > \tau \\ 1 & \text{if } E_i \leq \tau \end{cases}$$

This means that when the reconstruction error exceeds the threshold, the regularization term is set to 1, otherwise it is 0. The perception cooperation constraint regularization term is then introduced as:

$$\mathcal{P}_i = \frac{1}{m}\sum_{i=1}^{m}\mathbb{I}_i\sum_{j=1}^{k}a_{ij}^2$$

where k represents the dimension of the action, and m represents the batch size. Finally, the total loss function consists of the surrogate loss, value function loss, policy entropy, and the action regularization term:

$$\mathcal{L} = \mathcal{L}_{\text{surrogate}} + \lambda_v\mathcal{L}_{\text{value}} - \lambda_e\mathcal{H}(\pi) + \lambda_a\mathcal{P}_i$$

4 Experimental Results

4.1 Experiment Setup

We used the Unitree Go2 robot as our experimental subject, which features 12 degrees of freedom in its legs. Utilizing a single NVIDIA RTX 4090 GPU, we simultaneously trained 4096 domain-randomized Go2 robot environments in Isaac Gym. During training, we employed PD position controllers for each joint, with both the Blind Policy and Perception Policy running at a frequency of 50 Hz. The elevation map update rate was set to 10 Hz, and the robot's control signal delay was 20 ms. Additional domain randomization parameters and training specifics are detailed in the appendix.

The training terrain comprised six types: ramps, stairs, discrete obstacles, highlands, gaps, and pillar terrain. The first three terrains are relatively easier for the robot to navigate, while the last three require more reliance on external perception for anticipation. We primarily measured the robot's performance in both simulated and real-world settings under two conditions:

- The robot's ability to navigate the tough terrains with the aid of perception.
- The robot's capability to traverse the first three terrains when perception is suddenly lost.

4.2 Simulation Experiment

Terrain Passability Experiment: We first tested the survival rate of our policy across three tough terrains with varying levels of difficulty. For each terrain and difficulty level, we conducted 100 environment samples, calculated the success rate four times, and averaged the results. The success rate for the Gap and Pit terrains was defined as the robot successfully crossing or climbing over the obstacle, while for the Pillar terrain, it was defined as the proportion of environments the robot navigated without collisions. As shown in Table 1 , our policy achieved high success rates across various tough terrains. The highest difficulty level for each terrain was beyond the scope of our curriculum settings, demonstrating the robustness of our algorithm.

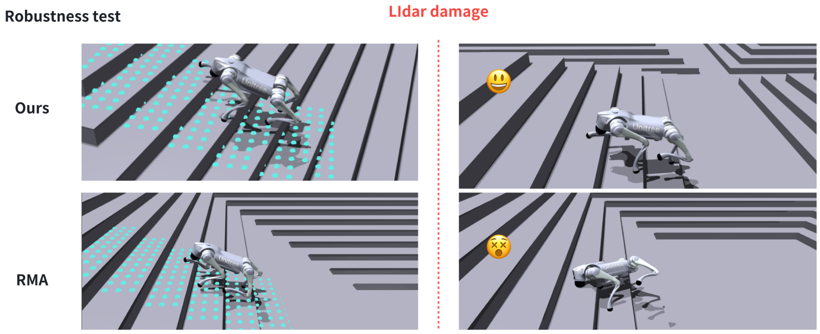

Figure 3: Robustness testing In simulation, the perception-based RMA mode collapses when the height map is corrupted while our policy works well.

Gap	Success Rate	Pit	Success Rate	Pillar	Success Rate
0.35m	**99.3%**	0.30m	**97.6%**	obstacle size=0.4 ; distance=1.6	**86.7%**
0.45m	**98.3%**	0.40m	**97.6%**	obstacle size=0.5 ; distance=1.5	**80.4%**
0.55m	**91.3%**	0.50m	**85.0%**	obstacle size=0.6 ; distance=1.4	**65.0%**
0.65m	**44.3%**	0.55m	**49.3%**	obstacle size=0.7 ; distance=1.3	**60.7%**

Table 1: Success Rates in Tough Terrains

Comparison Experiment: We compared our collision estimation and response policy with several baselines and ablations as follows:

- **Baseline**: Training directly with proprioception and height map.
- **RMA**: Employing an Adaptation Module to estimate all privileged observations, but directly inputting the elevation map into proprioception.
- **Ours w/o Regularizer**: Training without Perception Cooperation Constraint Regularization.

As shown in Table 2, our method demonstrates the most robust performance under external perception failure, especially when climbing stairs. Other strategies failed to learn to handle obstacles without perception during training, resulting in tripping over obstacles. In contrast, our method can easily climb steps, and the MXD indicates that our method can also achieve higher speeds (1 m/s to 1.6 m/s). Figure 3. shows the effect of our run in simulation.

4.3 Physical Experiments

Navigating Complex Terrains with Sensory Input Our policy substantially enhanced the quadruped robot's capability to navigate vertical challenges, such as wooden boxes and low walls. In our experiments, the robot was tasked with climbing a 32 cm high wooden box. It adeptly lifted its front legs preemptively and elevated its body to surmount the box, as shown in Figure 4. This sequence of movements, successfully culminating in the robot climbing over the box, exemplifies

Method	Up Stair Success	Down Stair Success	Discrete Success	Stair XMD	Discrete XMD
Ours	97%	100%	90%	19.97	17.04
RMA	0%	100%	81%	8.2	12.38
Baseline	0%	100%	76%	7.8	11.53
Ours w/o VAE	87%	100%	90%	16.42	14.99

Table 2: we primarily compared the success rates of different methods on stairs and discrete terrains, as well as the Mean X-Displacement (MXD) for each environment. For this experiment, all elevation map inputs were set to zero, and we tested 1048 environments over 1000 steps. The stairs had a width of 0.31 and a height of 0.13, while the maximum height of the discrete terrain was 0.15. Failure conditions were defined as either the roll or pitch exceeding 1.3, or the robot's foot getting stuck and unable to move forward.

the efficacy of our integrated elevation map and perceptual policies in enabling the robot to tackle climbing obstacles.

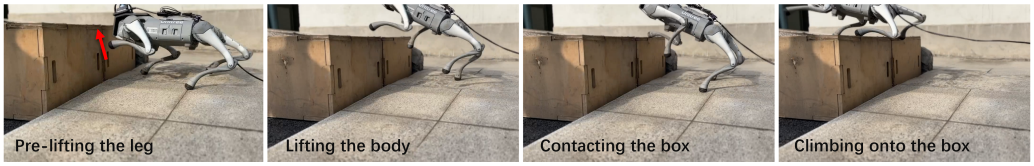

Figure 4: Robot Climbing a Wooden Box Using Our Policy.

In the obstacle avoidance trials, the robot encountered various obstacles including rocks, wooden boxes, and human figures. Leveraging our policy, it quickly recognized a human-shaped obstacle through its elevation map, then adeptly adjusted its trajectory, sidestepping to bypass the obstacle efficiently and safely, as depicted in Figure 5. This performance underscores our method's effectiveness, particularly noting that despite the absence of y-direction velocity training, the robot adeptly maneuvered in the y-direction, showcasing the robustness and adaptability of our approach.

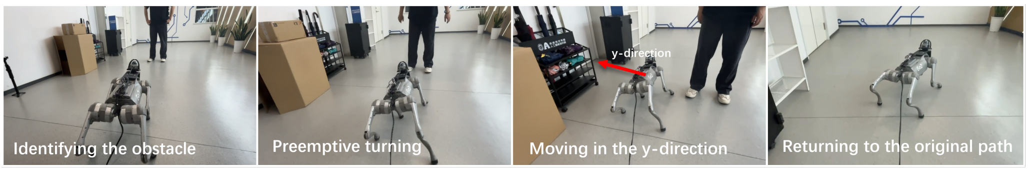

Figure 5: Robot Avoiding a Person Using Our Policy.

Long-Distance Test with Outdoor Terrain Perception Failure Initially, with effective LiDAR elevation map inputs, the robot used a comprehensive policy for movement, efficiently climbing 16 cm stairs and handling slopes. Subsequently, we deliberately covered the LiDAR, disabling the elevation map input, and conducted a long-distance test on unstructured terrains. We tested the robot over a 250m path that included dense grass, irregular terrain, soft and slippery grasslands, gentle slopes, and stair terrains, where the robot successfully navigated through all (see Figure 1).

5 Conclusion, Limitations and Future Directions

We propose the concept of Multi-Brain Collaborative Control based on Multi-Agent systems, establishing a training framework that achieves both perceptive motion and robust obstacle traversal in the event of perception failure. We tested our system in both simulations and real-world experiments, demonstrating the effectiveness and robustness of our algorithm. However, currently, our robot's elevation maps are derived from LiDAR, which heavily depends on the frequency and stability of the odometry, and involves significant computational overhead. This greatly affects the stability and sustainability of our perceptive policy. Additionally, our perceptive algorithm is still quite sensitive to environmental noise. In the future, we aim to replace LiDAR with lighter-weight perception devices such as cameras and construct local elevation maps without relying on odometry. We will also explore how to apply our algorithm to control various legged robots.

References

[1] M. T. Shahria, M. S. H. Sunny, M. I. I. Zarif, J. Ghommam, S. I. Ahamed, and M. H. Rahman. A comprehensive review of vision-based robotic applications: Current state, components, approaches, barriers, and potential solutions. *Robotics*, 11(6):139, 2022.

[2] I. M. A. Nahrendra, B. Yu, and H. Myung. Dreamwaq: Learning robust quadrupedal locomotion with implicit terrain imagination via deep reinforcement learning. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5078–5084. IEEE, 2023.

[3] J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter. Learning quadrupedal locomotion over challenging terrain. *Science robotics*, 5(47):eabc5986, 2020.

[4] Y. Cheng, H. Liu, G. Pan, L. Ye, H. Liu, and B. Liang. Quadruped robot traversing 3d complex environments with limited perception, 2024.

[5] J. Siekmann, K. Green, J. Warila, A. Fern, and J. Hurst. Blind bipedal stair traversal via sim-to-real reinforcement learning. *arXiv preprint arXiv:2105.08328*, 2021.

[6] Y. D. Yasuda, L. E. G. Martins, and F. A. Cappabianco. Autonomous visual navigation for mobile robots: A systematic literature review. *ACM Computing Surveys (CSUR)*, 53(1):1–34, 2020.

[7] P. Fankhauser, M. Bjelonic, C. D. Bellicoso, T. Miki, and M. Hutter. Robust rough-terrain locomotion with a quadrupedal robot. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5761–5768. IEEE, 2018.

[8] P. Fankhauser. *Perceptive locomotion for legged robots in rough terrain*. PhD thesis, ETH Zurich, 2018.

[9] D. Huh and P. Mohapatra. Multi-agent reinforcement learning: A comprehensive survey. *arXiv preprint arXiv:2312.10256*, 2023.

[10] S. Gronauer and K. Diepold. Multi-agent deep reinforcement learning: a survey. *Artificial Intelligence Review*, 55(2):895–943, 2022.

[11] D. B. Gillies. Solutions to general non-zero-sum games. *Contributions to the Theory of Games*, 4(40):47–85, 1959.

[12] C. Zhu, M. Dastani, and S. Wang. A survey of multi-agent reinforcement learning with communication. *arXiv preprint arXiv:2203.08975*, 2022.

[13] R. Lowe, Y. I. Wu, A. Tamar, J. Harb, O. Pieter Abbeel, and I. Mordatch. Multi-agent actor-critic for mixed cooperative-competitive environments. *Advances in neural information processing systems*, 30, 2017.

[14] J. Ackermann, V. Gabler, T. Osa, and M. Sugiyama. Reducing overestimation bias in multi-agent domains using double centralized critics. *arXiv preprint arXiv:1910.01465*, 2019.

[15] C. Yu, A. Velu, E. Vinitsky, J. Gao, Y. Wang, A. Bayen, and Y. Wu. The surprising effectiveness of ppo in cooperative multi-agent games. *Advances in Neural Information Processing Systems*, 35:24611–24624, 2022.

[16] J. Hu, S. Hu, and S.-w. Liao. Policy regularization via noisy advantage values for cooperative multi-agent actor-critic methods. *arXiv preprint arXiv:2106.14334*, 2021.

[17] T. Miki, J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter. Learning robust perceptive locomotion for quadrupedal robots in the wild. *Science Robotics*, 7(62):eabk2822, 2022.

[18] W. Yu, D. Jain, A. Escontrela, A. Iscen, P. Xu, E. Coumans, S. Ha, J. Tan, and T. Zhang. Visual-locomotion: Learning to walk on complex terrains with vision. In *5th Annual Conference on Robot Learning*, 2021.

[19] C. Mastalli, I. Havoutis, M. Focchi, D. G. Caldwell, and C. Semini. Motion planning for quadrupedal locomotion: Coupled planning, terrain mapping, and whole-body control. *IEEE Transactions on Robotics*, 36(6):1635–1648, 2020.

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

[21] Z. Zhuang, Z. Fu, J. Wang, C. Atkeson, S. Schwertfeger, C. Finn, and H. Zhao. Robot parkour learning. *arXiv preprint arXiv:2309.05665*, 2023.

[22] H. Duan, B. Pandit, M. S. Gadde, B. J. van Marum, J. Dao, C. Kim, and A. Fern. Learning vision-based bipedal locomotion for challenging terrain. *arXiv preprint arXiv:2309.14594*, 2023.

[23] Z. Fu, A. Kumar, A. Agarwal, H. Qi, J. Malik, and D. Pathak. Coupling vision and proprioception for navigation of legged robots. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17273–17283, 2022.

[24] P. Fankhauser, M. Bloesch, C. Gehring, M. Hutter, and R. Siegwart. Robot-centric elevation mapping with uncertainty estimates. In *Mobile Service Robotics*, pages 433–440. World Scientific, 2014.

[25] T. Miki, L. Wellhausen, R. Grandia, F. Jenelten, T. Homberger, and M. Hutter. Elevation mapping for locomotion and navigation using gpu. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2273–2280. IEEE, 2022.

[26] P. Fankhauser, M. Bloesch, and M. Hutter. Probabilistic terrain mapping for mobile robots with uncertain localization. *IEEE Robotics and Automation Letters*, 3(4):3019–3026, 2018.

[27] M. Stölzle, T. Miki, L. Gerdes, M. Azkarate, and M. Hutter. Reconstructing occluded elevation information in terrain maps with self-supervised learning. *IEEE Robotics and Automation Letters*, 7(2):1697–1704, 2022.

[28] G. Shani, J. Pineau, and R. Kaplow. A survey of point-based pomdp solvers. *Autonomous Agents and Multi-Agent Systems*, 27:1–51, 2013.

[29] M. T. Spaan and N. Spaan. A point-based pomdp algorithm for robot planning. In *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. 2004*, volume 3, pages 2399–2404. IEEE, 2004.

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