2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

Resilient Legged Local Navigation: Learning to Traverse with Compromised Perception End-to-End

Chong Zhang[†1], Jin Jin[†1], Jonas Frey[1,2], Nikita Rudin[1], Matías Mattamala[3], Cesar Cadena[1], Marco Hutter[1]

***Abstract*— Autonomous robots must navigate reliably in unknown environments even under compromised exteroceptive perception, or perception failures. Such failures often occur when harsh environments lead to degraded sensing, or when the perception algorithm misinterprets the scene due to limited generalization. In this paper, we model perception failures as invisible obstacles and pits, and train a reinforcement learning (RL) based local navigation policy to guide our legged robot. Unlike previous works relying on heuristics and anomaly detection to update navigational information, we train our navigation policy to reconstruct the environment information in the latent space from corrupted perception and react to perception failures end-to-end. To this end, we incorporate both proprioception and exteroception into our policy inputs, thereby enabling the policy to sense collisions on different body parts and pits, prompting corresponding reactions. We validate our approach in simulation and on the real quadruped robot ANYmal running in real-time (<10 ms CPU inference). In a quantitative comparison with existing heuristic-based locally reactive planners, our policy increases the success rate over 30 % when facing perception failures. Project Page: `https://bit.ly/45NBTuh`.**

I. Introduction

Reliable local navigation is essential for autonomous robots in the wild, which typically involves building a real-time local traversability map [1]–[3] or cost map [4], [5] for path planning. Generally, given an accurate map, it is not difficult for existing navigation planners to guide the robot towards the local goal safely. For legged robots that can robustly traverse various terrains [6], such local planners have become a routine and demonstrated remarkable performance in different tasks [7], [8].

However, real-world scenarios are complex, and near-perfect perception (*i.e.*, the map representation accurately reflects the environment, with negligible noises or drifts) is not always available. Cameras and LiDARs may not work in dark, rainy, and foggy environments. The vision system may not generalize to the experienced data especially when it is based on neural networks [9]. Transparent objects and obscured pits are untraversable and difficult to recognize. All these cases lead to invisible obstacles and pits for the navigation module, which we call ***perception failures***.

[†] Equal Contribution, listed alphabetically
[1] Robotic Systems Lab, ETH Zurich, Zurich, Switzerland
[2] Max Planck Institute for Intelligent Systems, Tübingen, Germany.
[3] Oxford Robotics Institute, the University of Oxford, UK.

This work was supported by the Swiss National Science Foundation (SNSF) through project 188596, the National Centre of Competence in Research Robotics (NCCR Robotics), the European Union's Horizon 2020 research and innovation program under grant agreement No 101016970, No 101070405, and No 852044, and an ETH Zurich Research Grant. Jonas Frey is supported by the Max Planck ETH Center for Learning Systems.

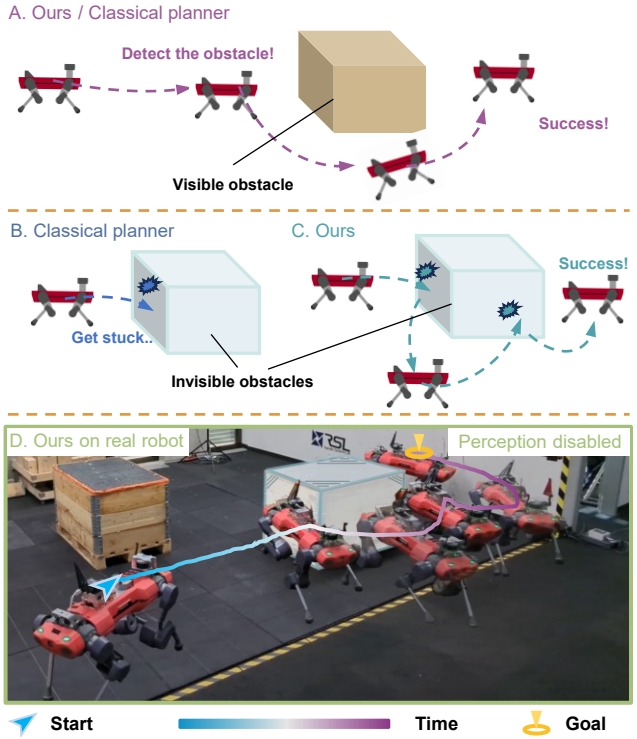

Fig. 1. Impact of failed perception (invisible obstacle) on navigation. (A) Without perception failures, both classical planners and our navigation policy can function well. (B) Under failed perception, the classical planner gets stuck. (C) Our proposed policy can react to perception failures and reach the target. (D) Our proposed policy functions on a real ANYmal robot being fully blind to obstacles.

Existing navigation methods struggle with perception failures, as they typically rely solely on exteroceptive sensors to interpret the environment. Most of them construct an explicit map for path planning based on traversability [1], [8] or geometry information [10], [11] of the surroundings, and some generate twist commands based on raw observations [12], [13]. When perception failure happens, these methods can have undesired behaviors, as illustrated in Fig. 1. Few recent works recognize these risks and achieve notable initial success by using anomaly detection and heuristic rules for replanning [14], [15]. However, such manually-designed rules cannot scale well to diverse situations.

To tackle perception failures, we intuitively need certain feedback to recognize the discrepancies between perceived and real environments. In this work, we propose to incorporate locomotion-level observations into navigation, contrasting existing methods that typically decouple navigation from locomotion. These observations include both proprioception and exteroception used by the locomotion module, inspired by Fu et al. [14] using proprioception for collision detection

and Ji et al. [15] using exteroception for anomaly detection.

We use end-to-end reinforcement learning (RL) to learn local navigation skills for legged robots. The learned navigation policy generates velocity commands to a pre-existing model-free low-level locomotion policy, and takes low-level observations as part of its inputs. We employ an asymmetric actor-critic design where the actor has corrupted exteroception and memory allowing it to infer the environment, while the critic can access privileged information including the external forces and the perfect map. Compared to heuristic-based local planners [3], our learned policy exhibits comparable performance under near-perfect perception, and far outperforms the baseline planner under heavily corrupted perception. We also show that legged robots can navigate in the real world even being fully blind.

Our main contributions are:

1) The methodology to learn local navigation strategies that can react to and recover from perception failures;
2) The design of simulation environments and reward functions to train our policy;
3) Experimental validation of our method both in simulation and real world.

II. Related Works

A. Planning-Based Local Navigation

Classical local planners for legged robots typically use heuristics (for sampling [16], [17], optimization [3], or rules [14]) or model-based techniques [18], [19] to guide the robot towards a local goal. A representation of the environment must be built from perception, based on geometric information [10], [17], [20]–[24] or semantics [1], [25], [26]. These planners assume near-perfect perception and plan the path with minimal cost based on the provided representation. However, if the perception system fails to identify an obstacle, these planners cannot handle it correctly. In this paper, we assume that perfect perception cannot be guaranteed, and therefore resign to learning a robust local planner that can handle these failure cases while exploiting the provided environment representation when it is reliable.

B. RL-Based Local Navigation

RL-based local navigation has recently emerged as it is powerful for behavior learning [27]–[30], computationally efficient and suitable for mapless navigation [12], [13], [31]–[33]. However, the lack of a map and the reliance on vision make them vulnerable to local minima and perception failures. For example, Faust et al. [27] and Khaled et al. [34] train map-based policies, but assume a near-perfect prebuilt map. Joanne et al. [35] uses a context map complimentary to the vision, but cannot handle visual perception failures.

In this paper, we use a traversability map [10] that can be imperfect, *i.e.*, the map representation can be inconsistent with the environment. We also utilize the proprioceptive information (*e.g.*, joint torques) and the scanned terrain geometry (*i.e.*, the exteroceptive data of the locomotion module) to sense and react to perception failures.

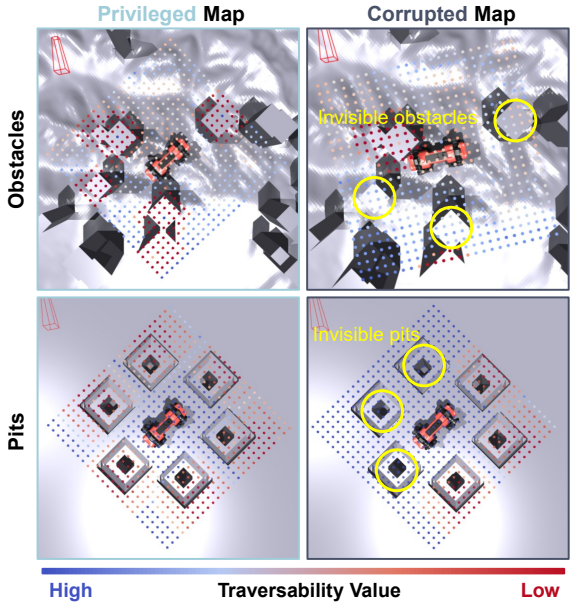

Fig. 2. An illustration of the privileged map and the corrupted map. The map (visualized by colored dots) provides the terrain traversability around the robot. Given an visibility ratio, certain part of the obstacles or pits are invisible to the robot, making their part on the map recognized as highly traversable.

C. Perception Failures

Some recent works tackle perception failures in the context of navigation [14], [15], [36]–[38]. Antonante et al. [37] propose risk estimation of perception failures for autonomous driving. Some works [38]–[40] propose to incorporate anomaly detection into multi-sensor fusion. Sun et al. [36] and Ji et al. [15] use perception data to predict potential failures. Fu et al. [14] study how proprioceptive data can predict collisions and thereby labeling transparent obstacles in the prebuilt map for navigation planning. In this paper, we explore how legged robots can automatically discover local navigation skills to overcome perception failures in a data-driven manner without heuristic rules.

Another way around is to improve the perception system. Examples are [1] where the perception module is continuously updated via self-supervision during deployment, and [41] where the exteroception can be updated by semantic perceptual feedbacks. However, such methods cannot support real-time reaction to perception failures, as they need to first update their perception module. Specific hardware setups [42], [43] may also enhance the system resilience, while this paper focuses on the algorithmic improvements.

III. Method

A. Overview

The objective of our method is to guide the robot to a local target within the given time. To simulate perception failures, we mask part of the obstacles and pits on the terrain to be invisible to the robot's exteroceptive sensing, as shown in Fig. 2. Besides reaching the target in time, the robot should also reduce base collisions and avoid falls.

An overview of our system is in Fig. 3. Given a pre-established model-free low-level locomotion policy [6], we

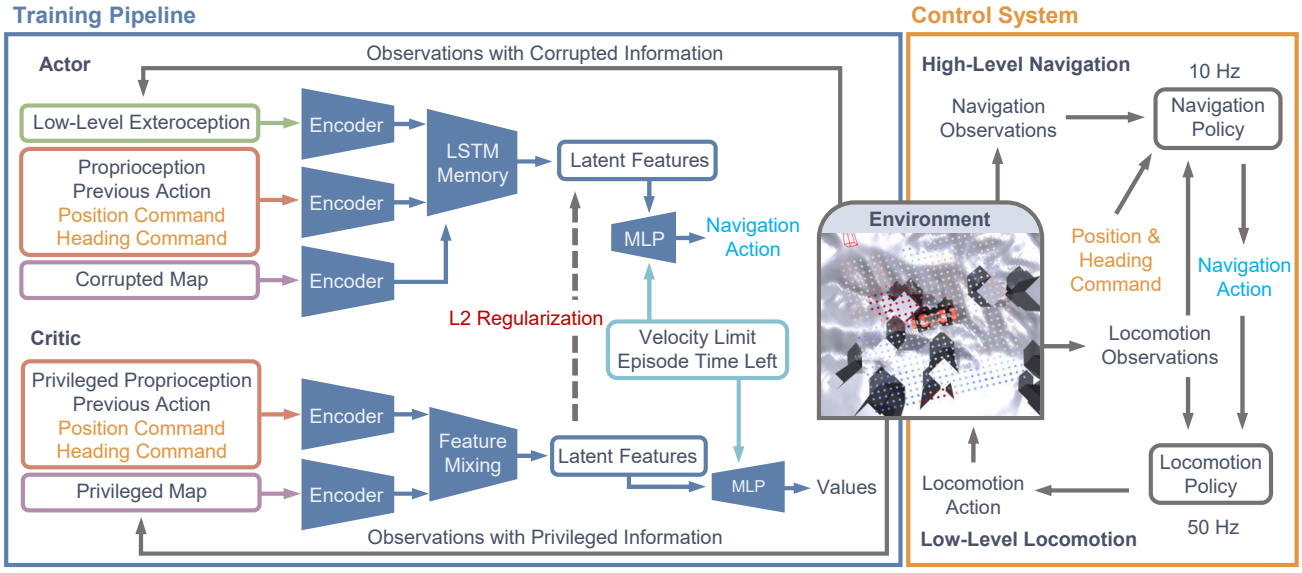

Fig. 3. An overview of our learning system. **Left** The actor-critic design of the navigation policy. **Right:** Our high-level navigation policy generates velocity commands tracked by a low-level locomotion policy. **Middle:** The simulation environment. The colored points around the robot are the visualized traversability map (as explained in Fig. 2), which makes the exteroceptive observation of the high-level policy.

train a navigation policy that generates velocity commands to be tracked in a hierarchical RL structure. We use the PPO algorithm [44] to automatically discover end-to-end navigation strategies capable to cope with perception failures.

B. Observations

There are four kinds of observations for our navigation policy: the low-level exteroception, the proprioception, the map, and other navigation-related states.

Low-Level Exteroception: The low-level exteroception is the sampled height scan h_t used for the locomotion policy, representing the local terrain perception. Around each foot, there are 52 scanned points, making h_t a 208-dim vector. We refer the readers to [6] for details.

Proprioception: The proprioception consists of the base linear velocity v_t, the base angular velocity ω_t, the last five frames of the base acceleration captured at 20 ms intervals β_t^{history}, the projected gravity on the base g_t, the joint positions q_t, the joint velocities $\dot{q}_t$, and the last eight frames of joint torques captured at 2.5 ms intervals τ_t^{history}. All of the aforementioned observations are obtained from the state estimator [45], and we use the raw IMU data for acceleration measurement. Additional privileged proprioception is provided to the critic consisting of the horizontal external force directions on the base and thighs F_t^{ext}, represented by unit vectors when forces are applied or zero vectors otherwise.

Traversability Map: We use the traversability map of a 3 m-wide square area around the robot with 4 cm resolution, using the method in [10]. Such a traversability map is down-sampled to a resolution of 12 cm as the map observations m_t. The map is illustrated in Fig. 2.

Navigation-Related States: Other navigation-related states include the previous action output a_{t-1}, the position command p_t^c (*i.e.*, the target position in the base frame), the heading command ψ_t^c (*i.e.*, the target yaw angle in the base frame), the left time of the episode $T-t$ following [28] where T is the episode length, and the velocity limit $\bar{v}$ allowing users to condition the robot velocity.

C. Asymmetric Actor-Critic with Memory

We apply asymmetric actor-critic following [46]. The critic network takes privileged proprioception without the joint torques, the privileged map, and other navigation-related states as the inputs. The actor network receives the corrupted map along with the corrupted low-level exteroception, the available proprioception from the state estimator, and other navigation-related states.

Both networks have a layer that mixes all features from different observations in the latent space. The critic network uses an MLP given that all information is observable from the privileged information. On the other hand, for the actor network, we use an LSTM layer to recurrently infer the partially observable environment following [33].

D. Implicit State Inference via Regularization

As the actor has no access to the ground truth perception, we would like to reconstruct the real information from the physical interactions and the corrupted perception. However, our cases do not fit in the popular techniques such as teacher-student [6], [47], [48] and supervised reconstruction [6]. First, in the teacher-student scheme, the privileged teacher can observe and avoid all of the obstacles, which differs from the expected student's behavior of first touching the unseen obstacle and then sidestepping it. Second, the supervised reconstruction scheme involves training decoders to reconstruct the raw privileged information. Yet, in a large map with many unperceived areas, the robot can only reconstruct the very limited part with which it has interacted, making the training of the decoder tricky and hard to generalize.

In this paper, we propose to use regularization to achieve implicit state inference. We assume that the critic's latent features contain all essential and relevant information for

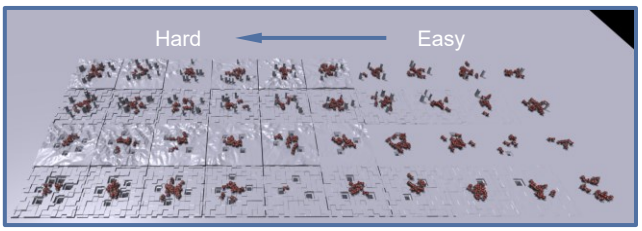

Fig. 4. We employ a terrain curriculum to facilitate learning. The terrains from top to bottom are obstacles on hilly, obstacles on boxes, pits on hilly, and pits on boxes. The difficulty increases from right to left.

decision-making and that trying to replicate this information can help the actor understand the scene. Therefore, we regularize the LSTM layer's outputs in the actor network towards the latent feature embeddings of the critic network, as is illustrated in Fig. 3.

E. Rewards

There are three types of reward terms involved: the task performance terms, the event penalties, and the regularization terms. All terms are summed up to make the reward function.

The task performance terms consist of two position tracking terms, one heading tracking term, one standing still term. They are based on the ones in [28] and dominate the rewards. Given a short duration T_r, these rewards are only received when $t > T - T_r$ regardless of what happens before, thereby avoiding local minima introduced by human priors. They are all in the form of

$$r_{\text{task}} = \frac{c_1}{1 + \|\frac{\text{error}}{c_2}\|^2}, \tag{1}$$

where c_1 and c_2 are positive constants, and the error is the distance-to-target for position tracking, the heading difference for heading tracking, and the velocity scalar for standing still. We have a small c_2 for a rigid position tracking term with a large c_1 so that the position tracking can be accurate, and a large c_2 for a soft position tracking term with a small c_1 so that the rewards are less sparse. The heading tracking term and the standing still term are only activated when the distance-to-target is below a threshold besides $t > T - T_r$.

Regarding event penalties, we have constant negative rewards to penalize the robot when it falls and terminates, or has a collision on the base or thighs. We also constantly penalize the robot for each timestep unless it is close to the target.

The regularization terms include the ones for energy consumption, action changes and magnitude, velocity tracking errors of the locomotion policy, the overspeeding above the limit, yaw rates, the directional discrepancy between the velocity and the heading, and the velocity commands being too small when the target is distant.

F. Terrain Curriculum

We employ curriculum learning to tackle challenging scenarios following [49] and [28]. We have discrete boxes in [49] and realistic hilly terrains in [50] as the basic terrains to avoid overfitting the flat terrain, and add obstacles and pits on them, as shown in Fig. 4. With increasing difficulty, the roughness and step height of the terrain increases as well as the number of obstacles and pits. A robot gets promoted to higher difficulty levels if it reaches the target position within the episode, and gets demoted otherwise.

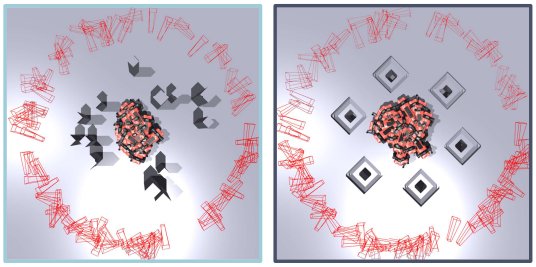

Fig. 5. Test terrains in simulation. The red boxes are the targets. 100 robots are tested together.

IV. Experimental Setup

A. Environments

We verify our methodology on the quadruped ANYmal robot both in simulation and in the real world. We use the Isaac Gym [51] for simulation. To simulate perception failure scenarios, we control the ratio of visible obstacles and pits (also called "visibility") during training and evaluation, where 100 % indicates that all obstacles or pits are correctly reflected within the traversability map and the height scan.

B. Training Settings

For training, we consider the following settings:

1) **Ours:** As is presented in Sec. III, we use all the observations we list in the methodology, and also all the techniques including the memory and the implicit state inference via regularization. 50 % of the obstacles and pits are visible *during training*.
2) **Oracle:** This setting is the same as **Ours** except that 100 % of the obstacles and pits are visible *during training*.
3) **No Proprioception (NoPro):** This setting differs from **Ours** in that we set the base acceleration, the projected gravity, and the joint states (positions, velocities, and torques) in proprioception to zero. These observations are typically not used in traditional navigation modules.
4) **No Low-Level Exteroception (NoExt):** This setting differs from **Ours** in that we mask the low-level exteroception with zeros.
5) **No LSTM Memory (NoMem):** This setting differs from **Ours** in that we replace the LSTM layer in the actor network with an MLP layer.
6) **No Regularization (NoReg):** This setting differs from **Ours** in that we cancel the regularization proposed in Sec. III-D for implicit state inference.

Each setting is run 3 times with different random seeds for statistics, which supports the significance of our performance by P Values. Besides these trained policies, we also use a heuristic-based local planner in [3], a representative for classicial local planners, as a baseline (abbreviated as **Planner**).

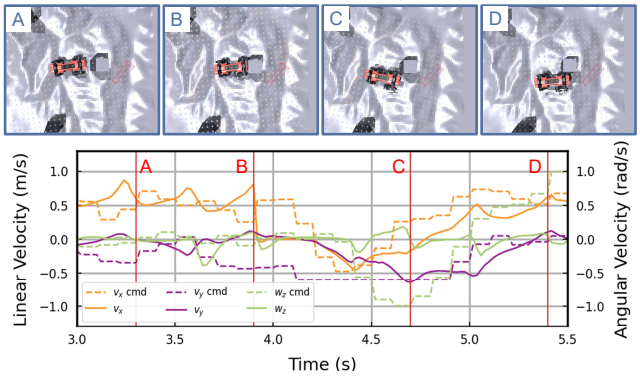

Fig. 6. Emergent behaviors of obstacle avoidance after a collision in simulation. The policy can react to the collision and make a sidestep.

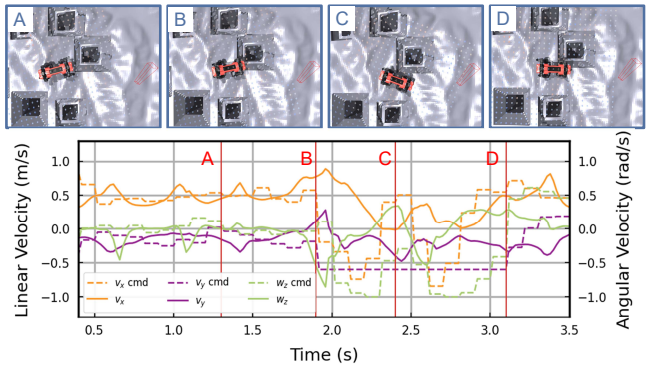

Fig. 7. Emergent behaviors of recovery from an invisible pit in simulation. The policy can react to the missed step and drag the robot out of the pit.

C. Evaluation and Metrics

In simulation, we build two test terrains (one for obstacles and one for pits) for quantitative evaluation of policies, as shown in Fig. 5. The visibility of the obstacles and pits is selected from $\{100\,\%, 50\,\%, 0\,\%\}$, making 6 simulated test environments: Obstacles-100 %, Obstacles-50 %, Obstacles-0 %, Pits-100 %, Pits-50 %, Pits-0 %.

We use the following metrics for quantitative comparison:

1) **Success Rate.** We count the proportion of robots that can reach the target within 18 s which is the episode length.
2) **Average Base Collisions.** We count the base collisions per 0.1 s for each robot and sum these values over time. This metric is only applicable to obstacles.
3) **Average Time Cost.** We count the average time cost for each robot to reach the target. For failed episodes, the time cost is calculated as 30 s (the qualitative results hold for any value $\geq$18 s).

D. Randomized Collision Bodies for Obstacles

To trigger different collision cases *during training* and enhance our policy's generalization ability, we randomize the collison body of the base on obstacle terrains. By doing so, we can learn policies that can react to collisions on different parts of the body in different directions, as demonstrated in Sec. V-D and video attachment.

V. Results and Analyses

A. Emergent Behaviors in Simulation

Our learned policy demonstrates reactive behaviors against perception failures on rough terrains in simulation, as shown in Fig. 6 and Fig. 7. When colliding with an unseen obstacle, our policy gives a backward longitudinal velocity and a large lateral velocity to sidestep the obstacle. When the robot missteps into an invisible pit, the policy generates velocity commands directing away from the pit until the robot regains stability.

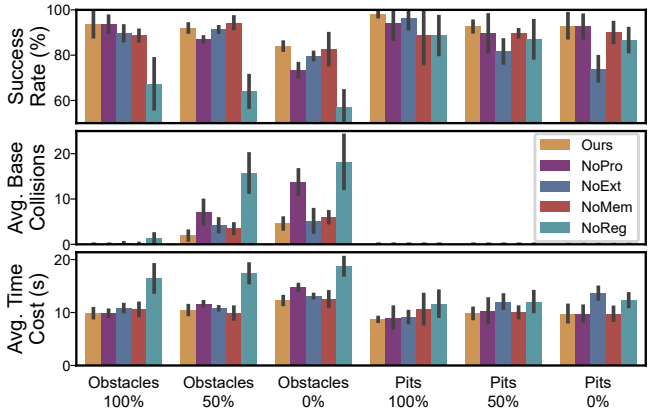

Fig. 8. Results of ablation studies. **Ours** outperforms others in the metrics.

B. Comparison Results

We compare the proposed **Ours** with the baselines **Oracle** and **Planner** in simulation. The results are presented in Table I, with the mean, std, and P values of the metrics. A P value smaller than 0.05 indicates that **Ours** is significantly better than **Oracle** regarding the metric in the environment.

According to the results, all of the policies perform well when the visibility is 100 %, and the **Planner** achieves a perfect 100 % success rate. However, as the visibility decreases, *i.e.*, when perception failures increase, **Ours** drop performance much slower than the other two, and significantly outperform them. **Ours** can have $>80\,\%$ success rates even when the perception is fully corrupted, and generalize well to 0 % and 100 % visibility despite being trained with 50 % visibility. These results indicate that the navigation policy cannot learn to react to perception failures without being exposed to them, and the locomotion policy cannot overcome perception failures by its own.

It is also worth mentioning that, the **Planner** receives a 8 m $\times$ 8 m map, but our policies only receives a 3 m $\times$ 3 m one, which makes our policies more vulnerable to local minima and fail in rare cases even with perfect perception. However, expanding the map observation is non-trivial as it increases the dimensionality and can make the training data-inefficient or even failed.

C. Ablation Studies

We evaluate different settings for ablation studies in simulation. The results are presented in Fig. 8, with error bars describing the std values. We draw the following conclusions based on the ablation results:

1) The proprioception as part of the observations is generally beneficial to the robustness against perception failures, as shown by the **NoPro** results.
2) The locomotion-level exteroception can help improve the performance on pits, but demonstrates fewer effects for

TABLE I

COMPARISON RESULTS IN SIMULATION

Environment	Success Rate (%)				Avg. Base Collisions				Avg. Time Cost (s)			
	Oracle	Ours	P Value	Planner	Oracle	Ours	P Value	Planner	Oracle	Ours	P Value	Planner
Obstacles-100 %	93.3±1.5	93.7±5.5	0.538	**100**	1.4±2.1	0±0	0.156	0.2	9.5±0.4	9.9±0.8	0.750	**6.9**
Obstacles-50 %	54.0±6.9	**92.0±1.7**	<0.001	61	32.1±4.3	**1.9±0.9**	<0.001	23.8	18.1±1.3	**10.5±0.8**	<0.001	15.7
Obstacles-0 %	33.3±6.4	**84.0±1.7**	<0.001	53	48.3 ± 13.6	**4.6±1.2**	0.003	24.9	22.5±1.5	**12.3±0.7**	<0.001	18
Pits-100 %	**99.7±0.6**	98.0±1.0	0.967	**100**					7.4±0.4	8.7±0.3	0.995	**6.1**
Pits-50 %	74.0±4.4	**92.7±2.3**	0.001	77	0	0	None	0	13.0±1.3	**9.8±0.9**	0.013	11.5
Pits-0 %	57.3±3.1	**93.0±5.3**	<0.001	57					18.7±0.6	**9.8±1.5**	<0.001	16.4

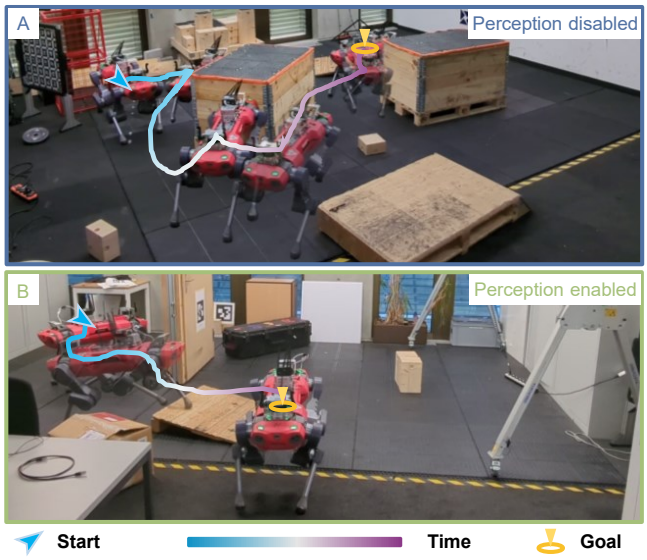

Fig. 9. (A) Our proposed policy can reach the target when the robot is blind. It can react to collisions on the thighs, the base front, the legs, and the base side. (B) Our proposed policy can traverse rough terrains when perception is enabled.

obstacles, as shown by **NoExt** results. We hypothesize that other observations are sufficient to detect the collisions, but insufficient to detect a missed step in time, while the exteroception may show some discrepancy with other observations.

3) The memory only slightly contributes to robustness, as shown by **NoMem** results. This does not comply with [33] where memory is the key to blind navigation. This can be explained by the detectability of collisions from other observations without memory, *e.g.*, in the bug algorithm [52].
4) The regularization for implicit state inference is *critical* to the performance in every scene, as shown by the **NoReg** results. Without such regularization to guide the learning, the policy may converge to a suboptimal solution.

Briefly speaking, both the locomotion-level observations and all techniques we propose contribute to the navigational robustness against perception failures.

D. Real-World Results

In the real world, we compare **Ours** against **Planner** in our lab. **Our** policy can perform well around obstacles not only when there is near-perfect perception, but also when being fully blind, as shown in Fig. 9. Under perception failures, the robot first collides with the obstacle and then tries to sidestep around it. Such reactions can happen with collisions

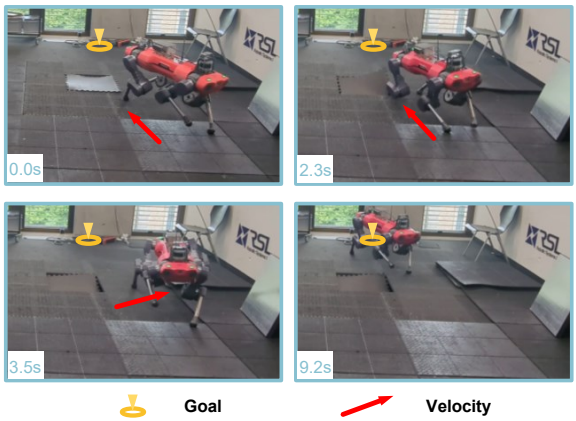

Fig. 10. Snapshots of our policy recovering from an invisible pit. After one missed step in the pit, the policy reacts with velocity commands to the right, and guides the robot slowly towards the target after recovery. Red arrows indicate the moving direction of the robot.

on different parts of the robot and in different directions, including the base, the thighs, and the legs (including feet). In comparison, the **Planner** always gets stuck in front of invisible obstacles (motions in our video attachment).

Our policy can also recover from an invisible pit, as shown in Fig. 10. It can react at the moment when the robot's foot goes deep under the perceived terrain and tries to drag the robot out of the pit with lateral velocity commands. After the recovery, the policy can keep guiding the robot toward the target. In comparison, the **Planner** cannot react to such cases and the locomotion policy cannot recover by itself.

VI. LIMITATIONS AND FUTURE WORKS

Despite our policy's generalization to different collision geometries, we find it cannot handle out-of-distribution mapping noises. Real world mapping corruption is hard to model. They are dependent on the SLAM module as well as on the various sensor modalities used to measure the environment.

Besides, our policy always avoids untraversable regions on the map. However, the perception module can generate artifacts, *i.e.*, non-existent obstacles or pits which can stop the robot from approaching the target. Hence, it is of great interest if we can train a policy to actively explore these areas and explicitly revise the map, allowing it to take into account false positive and false negative perception failures.

Besides, further research is needed to identify the optimal map representation provided to the local planner, which is currently limited to a 2.5D map involving only geometric information, as well as improving the performance under perfect perception. Further, we will investigate scaling the proposed approach to fully rough terrains in the wild.

REFERENCES

[1] J. Frey, M. Mattamala, N. Chebrolu, C. Cadena, M. Fallon, and M. Hutter, "Fast Traversability Estimation for Wild Visual Navigation," in *Robotics: Science and Systems*, Daegu, South Korea, July 2023.

[2] L. Wellhausen and M. Hutter, "Artplanner: Robust legged robot navigation in the field," *Field Robotics*, vol. 3, no. 1, pp. 413 – 434, 2023-03.

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

[4] D. Wooden, M. Malchano, K. Blankespoor, A. Howardy, A. A. Rizzi, and M. Raibert, "Autonomous navigation for bigdog," in *2010 IEEE international conference on robotics and automation*. Ieee, 2010, pp. 4736–4741.

[5] B. Yang, L. Wellhausen, T. Miki, M. Liu, and M. Hutter, "Real-time optimal navigation planning using learned motion costs," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 9283–9289.

[6] T. Miki, J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter, "Learning robust perceptive locomotion for quadrupedal robots in the wild," *Science Robotics*, vol. 7, no. 62, p. eabk2822, 2022.

[7] M. Tranzatto, T. Miki, M. Dharmadhikari, L. Bernreiter, M. Kulkarni, F. Mascarich, O. Andersson, S. Khattak, M. Hutter, R. Siegwart, *et al.*, "Cerberus in the darpa subterranean challenge," *Science Robotics*, vol. 7, no. 66, p. eabp9742, 2022.

[8] D. D. Fan, K. Otsu, Y. Kubo, A. Dixit, J. Burdick, and A. Agha-mohammadi, "STEP: stochastic traversability evaluation and planning for risk-aware off-road navigation," in *Robotics: Science and Systems XVII, Virtual Event, July 12-16, 2021*, D. A. Shell, M. Toussaint, and M. A. Hsieh, Eds., 2021. [Online]. Available: https://doi.org/10.15607/RSS.2021.XVII.021

[9] C. Kamann and C. Rother, "Benchmarking the robustness of semantic segmentation models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 8828–8838.

[10] T. Miki, L. Wellhausen, R. Grandia, F. Jenelten, T. Homberger, and M. Hutter, "Elevation mapping for locomotion and navigation using gpu," 2022.

[11] Y. Kim, C. Kim, and J. Hwangbo, "Learning Forward Dynamics Model and Informed Trajectory Sampler for Safe Quadruped Navigation," in *Proceedings of Robotics: Science and Systems*, New York City, NY, USA, June 2022.

[12] 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.

[13] J. Truong, M. Rudolph, N. H. Yokoyama, S. Chernova, D. Batra, and A. Rai, "Rethinking sim2real: Lower fidelity simulation leads to higher sim2real transfer in navigation," in *Conference on Robot Learning*. PMLR, 2023, pp. 859–870.

[14] Z. Fu, A. Kumar, A. Agarwal, H. Qi, J. Malik, and D. Pathak, "Coupling vision and proprioception for navigation of legged robots," in *CVPR*, 2022.

[15] T. Ji, A. N. Sivakumar, G. Chowdhary, and K. Driggs-Campbell, "Proactive anomaly detection for robot navigation with multi-sensor fusion," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 4975–4982, 2022.

[16] M. Wermelinger, P. Fankhauser, R. Diethelm, P. Krüsi, R. Siegwart, and M. Hutter, "Navigation planning for legged robots in challenging terrain," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016, pp. 1184–1189.

[17] L. Wellhausen and M. Hutter, "Rough terrain navigation for legged robots using reachability planning and template learning," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 6914–6921.

[18] S. Bansal, V. Tolani, S. Gupta, J. Malik, and C. Tomlin, "Combining optimal control and learning for visual navigation in novel environments," 2019.

[19] 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.

[20] P. Fankhauser and M. Hutter, "A universal grid map library: Implementation and use case for rough terrain navigation," *Robot Operating System (ROS) The Complete Reference (Volume 1)*, pp. 99–120, 2016.

[21] J. Frey, D. Hoeller, S. Khattak, and M. Hutter, "Locomotion policy guided traversability learning using volumetric representations of complex environments," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 5722–5729.

[22] H. Oleynikova, A. Millane, Z. Taylor, E. Galceran, J. Nieto, and R. Siegwart, "Signed distance fields: A natural representation for both mapping and planning," in *RSS 2016 workshop: geometry and beyond-representations, physics, and scene understanding for robotics*. University of Michigan, 2016.

[23] J. Mainprice, N. Ratliff, M. Toussaint, and S. Schaal, "An interior point method solving motion planning problems with narrow passages," in *2020 29th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*. IEEE, 2020, pp. 547–552.

[24] H. Oleynikova, Z. Taylor, M. Fehr, R. Siegwart, and J. Nieto, "Voxblox: Incremental 3d euclidean signed distance fields for on-board mav planning," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 1366–1373.

[25] L. Nardi and C. Stachniss, "Actively improving robot navigation on different terrains using gaussian process mixture models," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 4104–4110.

[26] L. Wellhausen, A. Dosovitskiy, R. Ranftl, K. Walas, C. Cadena, and M. Hutter, "Where should i walk? predicting terrain properties from images via self-supervised learning," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1509–1516, 2019.

[27] A. Faust, K. Oslund, O. Ramirez, A. Francis, L. Tapia, M. Fiser, and J. Davidson, "Prm-rl: Long-range robotic navigation tasks by combining reinforcement learning and sampling-based planning," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 5113–5120.

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

[29] D. Hoeller, N. Rudin, D. Sako, and M. Hutter, "Anymal parkour: Learning agile navigation for quadrupedal robots," *arXiv preprint arXiv:2306.14874*, 2023.

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

[31] L. Xie, S. Wang, A. Markham, and N. Trigoni, "Towards monocular vision based obstacle avoidance through deep reinforcement learning," in *Robotics: Science and Systems Workshop 2017: New Frontiers for Deep Learning in Robotics*, 2017.

[32] X. Zhao, H. Agrawal, D. Batra, and A. G. Schwing, "The surprising effectiveness of visual odometry techniques for embodied pointgoal navigation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2021, pp. 16 127–16 136.

[33] E. Wijmans, M. Savva, I. Essa, S. Lee, A. S. Morcos, and D. Batra, "Emergence of maps in the memories of blind navigation agents," in *The Eleventh International Conference on Learning Representations*, 2023.

[34] K. Nakhleh, M. Raza, M. Tang, M. Andrews, R. Boney, I. Hadzic, J. Lee, A. Mohajeri, and K. Palyutina, "Sacplanner: Real-world collision avoidance with a soft actor critic local planner and polar state representations," *arXiv preprint arXiv:2303.11801*, 2023.

[35] J. Truong, A. Zitkovich, S. Chernova, D. Batra, T. Zhang, J. Tan, and W. Yu, "Indoorsim-to-outdoorreal: Learning to navigate outdoors without any outdoor experience," *arXiv preprint arXiv:2305.01098*, 2023.

[36] B. Sun, J. Xing, H. Blum, R. Siegwart, and C. Cadena, "See yourself in others: Attending multiple tasks for own failure detection," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 8409–8416.

[37] P. Antonante, S. Veer, K. Leung, X. Weng, L. Carlone, and M. Pavone, "Task-aware risk estimation of perception failures for autonomous vehicles," *arXiv preprint arXiv:2305.01870*, 2023.

[38] K. Jin, F. Mu, X. Han, G. Wang, and Z. Liu, "Anomaly detection for robust autonomous navigation," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 10 026–10 032.

[39] M. A. Lee, M. Tan, Y. Zhu, and J. Bohg, "Detect, reject, correct: Crossmodal compensation of corrupted sensors," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 909–916.

[40] N. Khedekar, M. Kulkarni, and K. Alexis, "Mimosa: A multi-modal slam framework for resilient autonomy against sensor degradation," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 7153–7159.

[41] V. Vasilopoulos, G. Pavlakos, K. Schmeckpeper, K. Daniilidis, and D. E. Koditschek, "Reactive navigation in partially familiar planar environments using semantic perceptual feedback," *The International Journal of Robotics Research*, vol. 41, no. 1, pp. 85–126, 2022.

[42] S. Khattak, C. Papachristos, and K. Alexis, "Visual-thermal landmarks and inertial fusion for navigation in degraded visual environments," in *2019 IEEE Aerospace Conference*. IEEE, 2019, pp. 1–9.

[43] P. De Petris, H. Nguyen, M. Kulkarni, F. Mascarich, and K. Alexis, "Resilient collision-tolerant navigation in confined environments," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 2286–2292.

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

[45] M. Bloesch, M. Hutter, M. A. Hoepflinger, S. Leutenegger, C. Gehring, C. D. Remy, and R. Siegwart, "State estimation for legged robots-consistent fusion of leg kinematics and imu," *Robotics*, vol. 17, pp. 17–24, 2013.

[46] Y. Ma, F. Farshidian, and M. Hutter, "Learning arm-assisted fall damage reduction and recovery for legged mobile manipulators," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 12 149–12 155.

[47] J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter, "Learning quadrupedal locomotion over challenging terrain," *Science robotics*, vol. 5, no. 47, p. eabc5986, 2020.

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

[49] 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.

[50] C. Zhang and L. Yang, "Generating a terrain-robustness benchmark for legged locomotion: A prototype via terrain authoring and active learning," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023.

[51] V. Makoviychuk, L. Wawrzyniak, Y. Guo, M. Lu, K. Storey, M. Macklin, D. Hoeller, N. Rudin, A. Allshire, A. Handa, and G. State, "Isaac gym: High performance GPU based physics simulation for robot learning," in *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021. [Online]. Available: https://openreview.net/forum?id=fgFBtYgJQX_

[52] J. Ng and T. Bräunl, "Performance comparison of bug navigation algorithms," *Journal of Intelligent and Robotic Systems*, vol. 50, pp. 73–84, 2007.