Learning to Navigate Over Clutter in Indoor Environments using Vision

Anonymous Author(s)
Affiliation
Address
email

Abstract: We present Visual Navigation and Locomotion over obstacles (ViNL), which enables a quadrupedal robot to navigate unseen apartments while stepping over small obstacles that lie in its path (e.g., shoes, toys, cables). ViNL consists of: (1) a visual navigation policy that outputs linear and angular velocity commands that guides the robot to a goal in novel indoor environments; and (2) a visual locomotion policy that controls the robot's joints to avoid stepping on obstacles while following provided velocity commands. These two policies are trained independently, and can be seamlessly be coupled together upon deployment by feeding the velocity commands from the navigation policy to the locomotion policy. While several related prior works have demonstrated learning visual navigation policies or learning robust locomotion control, to the best of our knowledge, this is the first fully learned approach that leverages vision to accomplish both (1) intelligent navigation in new environments, and (2) intelligent visual locomotion that aims to traverse cluttered environments *without* disrupting obstacles. We find that ViNL outperforms prior work that was trained to robustly walk over challenging terrain using privileged terrain maps (+32.8% success and -4.4 collisions per meter).

Keywords: Legged Locomotion, Reinforcement Learning, Visual Navigation

1 Introduction

For mobile robotic assistants to be useful in the real-world, they must skillfully navigate environments they have never seen. This is critical since indoor environments are subject to constant change (e.g., change in furniture layouts, temporary clutter, etc.). In recent years, robust indoor navigation has seen significant progress using learned agents due to advances in deep reinforcement learning, extensive datasets of real-world indoor scans [1], and scalable photo-realistic simulation [2–6]. Works such as [7–12] show that agents trained entirely in simulation can be deployed in the real world in previously unseen environments without using a pre-computed map.

However, current progress in visual indoor navigation within previously unseen environments has been largely limited to using wheeled-base robots in homes with immaculate, flat terrain. Typical wheeled robots have limited maneuverability, which can pose a problem even in indoor home environments; they have difficulty going over clutter, doorways with thresholds, stairs, and even thicker carpets. In contrast, legged robots are well-suited for navigating under such conditions. In particular, they can step over obstacles without disrupting or breaking them. While there are several works in learning legged locomotion, these works often use blind and reactive policies meant for outdoor environments, which emphasize stability and robust recovery on rough terrain [13, 14].

In our work we aim to bring legged robots to the unstructured and messy human world by enabling them to navigate *over* clutter typically found in indoor environments. We replicate a cat's ability to carefully walk over small household obstacles (such as shoes, toys, clothes, etc.), rather than navigating around these obstacles entirely. We propose a fully learned hierarchical approach that uses only egocentric vision, proprioception, and egomotion estimates. Our low-level visual locomotion policy is learned in three stages using Isaac Gym [15]. First, we learn to walk in a wide variety of challenging terrains using large-scale deep reinforcement learning. Next, we learn to walk over clutter by fine-tuning the previous policy in a novel terrain that contains small obstacles on the ground. Finally, we learn to walk over clutter using egocentric vision through supervised learning to reconstruct the privileged terrain map using vision alone. Separately, we train a high-level visual

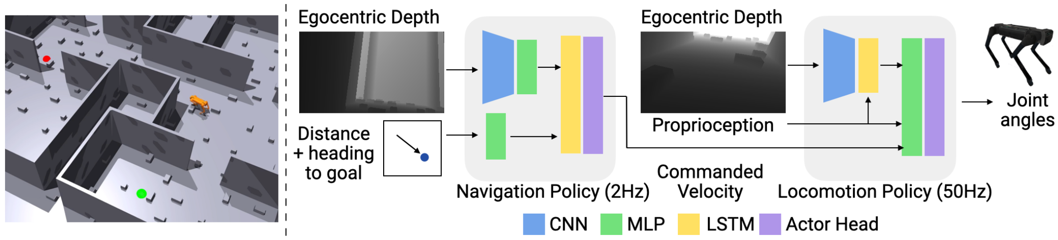

Figure 1: **Left:** ViNL navigates to goals in previously unseen environments while stepping over small obstacles on the ground. **Right:** ViNL is a fully learned hierarchical approach for navigation and locomotion.

navigation policy in photo-realistic 3D scans of real-world indoor environments using Habitat [2, 3]. Although the locomotion and navigation policies are trained in two different simulators (Isaac Gym and Habitat), the two can be combined zero-shot for the task of visual navigation over obstacles.

The core contributions of our work are: 1) We propose ViNL, the first approach to the best of our knowledge to accomplish both intelligent navigation in new environments, and intelligent visual locomotion that aims to traverse cluttered environments *without* disrupting obstacles; 2) We show that our visual navigation policy can be seamlessly deployed to a simulator using full rigid-body dynamics and a low-level locomotion policy, despite being trained kinematically in a different simulator and different low-level control; 3) ViNL can successfully guide the robot to the goal in cluttered indoor environments with a success rate of 73.6%, a 32.8% increase from prior works [16].

2 Method

We propose ViNL, a hierarchical architecture (Figure 1 right) for navigating over clutter in indoor environments, which consists of (1) a visual locomotion policy (Figure 2) that avoids stepping on small obstacles, and (2) a visual navigation policy that commands linear and angular velocities. The locomotion and navigation policies are trained in parallel, independently of each other.

2.1 Visual Locomotion

We utilize a three-stage approach for learning a low-level controller that walks over small obstacles.

Stage 1: Learning to Walk. We first aim to learn a teacher locomotion policy that can robustly walk over challenging terrain (*e.g.*, stairs, rough terrain, etc.). Using the rough terrain environment in the Isaac Gym benchmark [16], we teach AlienGo using deep reinforcement learning to walk over challenging terrain while following commanded linear and angular velocities. In contrast to [16], we encode the local map of the robot's terrain using a 3-layer MLP, and pass the map encoding to the policy with the rest of the observations (Figure 2).

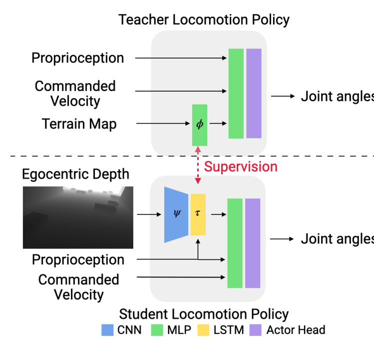

Figure 2: Teacher's encoder (terrain maps) distilled to student's (vision).

The teacher locomotion policy takes as input the robot's proprioceptive state and a privileged height map $\boldsymbol{H}$ containing information about the surrounding terrain. Specifically, at timestep t, the observation space consists of: joint positions $\boldsymbol{q}_t \in \mathbb{R}^{12}$, joint velocities $\dot{\boldsymbol{q}}_t \in \mathbb{R}^{12}$, previous joint commands $\boldsymbol{q}_{t-1} \in \mathbb{R}^{12}$, base linear velocity $\mathbf{v} \in \mathbb{R}^3$, base angular velocity $\boldsymbol{\omega} \in \mathbb{R}^3$, projected gravity vector $\boldsymbol{g} \in \mathbb{R}^3$, commanded velocities $(\boldsymbol{v_x}, \boldsymbol{v_y}, \boldsymbol{\omega_z})^* \in \mathbb{R}^3$, and a terrain map $\boldsymbol{H} \in \mathbb{R}^{187}$. The terrain map contains terrain height samples from a 1.6m $\times$ 1.0m grid around the robot. The output of the locomotion policy consists of joint angles that are used as target positions for PD motor controllers. We use the same rewards from [16], which encourages the robot to maintaining a smooth, natural gait. The episode is terminated when the robot falls over.

Stage 2: Learning to Walk Over Clutter.

We construct an obstacle terrain consisting of small blocks on the ground (Figure 3). We utilize a curriculum of increasing obstacle density, similar to [16]. The terrain consists of 25, 8m $\times$ 8m tiles, each with a different density of obstacles. Robots which are able to walk more than half the total tile distance in an episode are promoted to a more challenging tile in the terrain. Using this terrain and density

Figure 3: Our obstacle terrain.

curriculum, we fine-tune the locomotion policy from Stage 1 with an added penalty discouraging contact between the obstacles on the ground and the robot's feet. Specifically, the robot receives a penalty of -1 if any leg is in contact with an obstacle.

Stage 3: Learning to Walk Over Clutter with Vision.

Finally, we aim to lift the assumption of access to a privileged terrain map for navigating over clutter. Instead of a terrain map H, we train the robot to avoid clutter using egocentric vision d from a front-facing depth camera pitched downwards 30°. Figure 4 shows an example of our locomotion policy using vision to lift up its feet to avoid obstacles in its path.

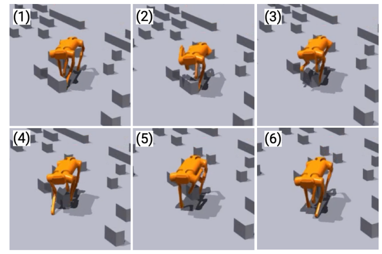

Figure 4: Time-lapse of locomotion policy walking over clutter.

Drawing inspiration from Kumar et al. [13], we use Learning by Cheating [17] to predict the encoding of a privileged terrain map using an encoder that takes in egocentric depth observations and proprioceptive states of the robot. A key difference between our method and Kumar et al. is that they use a history of 50 consecutive observations. In contrast, we rely on an LSTM provide the policy with temporal information. At each step, the LSTM takes in visual encodings of depth images from a CN, proprioceptive states of the robot and its previous hidden state to predict the encoding of the current terrain map. An LSTM enables the policy to leverage temporal dependencies using the hidden state, without the costly compute and memory that is typically required for processing a large buffer of images. This is important for navigating over clutter using a front-facing egocentric depth camera. As the robot walks over objects, and the objects leave the camera's view, it must make use of past temporal information in order to remember the location of these obstacles and avoid them with its legs. We train using supervised learning with on-policy data to minimize MSE loss, similar to [13, 18].

2.2 Visual Navigation

Task & Dataset. In PointGoal Navigation [19], a robot is tasked with navigating to a goal location in an indoor environment without being given a pre-built map of the environment. An episode is considered successful when the robot reaches within 0.325m of the goal. For training and evaluation, we use both the Habitat-Matterport (HM3D) [1] and Gibson [20] 3D datasets, which contains over 1000 scans of real-world indoor environments (homes, offices, etc.).

Kinematic Visual Navigation. The navigation policy takes as input an egocentric depth image and the robot's current distance and heading relative to the goal. The output of the navigation policy is the desired center-of-mass forward linear and angular velocities. The overall architecture of the navigation policy is shown in Figure 1 right. Specifically, we use a ResNet-18 [21] visual encoder to process the egocentric depth images, a linear layer to encode the goal vector, and a 2-layer LSTM for our policy. We use the reward function from [9], which encourages path efficiency and discourages collisions. The policy is trained using DD-PPO [22] using the Habitat simulator [3].

We use kinematic low-level control as an approximation for the robot's movement, which was shown in [9] to lead to better sim-to-real transfer. At each step, we directly teleport the robot (without running full rigid-body dynamics simulation) by integrating the commanded velocities at 2 Hz. If the resulting pose intersects with the environment, the robot is simply kept in its current pose. We additionally randomize the roll and pitch of the front-facing camera by up to $\pm 30°$ during training to improve the robustness of the policies against camera shake that occurs on moving quadrupeds.

3 Results

In this section, we aim to address the following questions: 1) How well can our navigation policy perform in a different simulator and low-level control? 2) How does ViNL compare to prior work? 3) How vital is the use of exteroception, pre-training, or an LSTM for the locomotion policy?

We evaluate using the task of PointGoal Navigation over clutter across 5 unique floor plans in Isaac Gym as shown on the right of Figure 1. In each floor plan, we randomly sample 10 start and goal locations for the robot, and report results across 5 seeds for a total of 250 experiments per method. For evaluation, we report success rate (SR), distance traveled, and foot collisions per meter. Since the low-level controller is being run at 50 Hz, a foot contact lasting 1 second will count as 50 foot collisions. During evaluation, the navigation policy uses a front-facing depth camera pitched 15° upwards, while the locomotion policy uses a front-facing depth camera pitched 30° downwards.

Table 1: We evaluate all policies in cluttered floor plans simulated in Isaac Sim.

Method	Sensors		Train Terrain		Leg Contact Penalty	Success Rate ↑	Distance traveled (m) ↑	Foot Collisions per Meter ↓
	Terrain Map	Vision	Rough	Obstacle				
Blind	✗	✗	✓	✓	✓	1.2 ± 1.0	$\mathbf{10.2 \pm 0.8}$	18.2 ± 5.3
Rough	✓	✗	✓	✗	✗	40.8 ± 3.5	7.6 ± 0.2	16.5 ± 2.5
ViNL (no-pretraining)	✗	✓	✗	✓	✓	54.4 ± 8.0	6.6 ± 1.0	14.1 ± 7.1
ViNL (MLP)	✗	✓	✓	✓	✓	66.8 ± 8.1	7.5 ± 0.7	12.2 ± 0.4
ViNL	✗	✓	✓	✓	✓	$\mathbf{73.6 \pm 5.9}$	8.2 ± 0.3	$\mathbf{12.1 \pm 0.7}$
ViNL Eval. w/ no obstacles	✗	✓	✓	✓	✓	86.8 ± 4.7	7.7 ± 0.4	2.9 ± 0.3

To get an upper bound for the performance of ViNL, we first evaluate the performance of our visual navigation policy for PointGoal Navigation without clutter. We evaluate 'in-domain' (kinematic low-level control in Habitat), and compare its performance when evaluated out-of-domain (low-level control using ViNL in Isaac Gym). In-domain, the visual navigation policy is evaluated using 1000 episodes from scenes in the validation split of the HM3D dataset, and achieves a success rate (SR) of 89.70%. Out-of-domain, ViNL has a SR of 86.8% (Table 1, row 6), indicating a small sim-to-sim gap (-2.9% SR). This demonstrates that despite training the high-level navigation policy and low-level locomotion policy *separately in different simulators and different levels of abstraction*, the two can be nearly seamlessly coupled together with a minimal drop in navigation performance.

Next, we compare ViNL to other baselines and ablations. We couple the same high-level navigation policy with the following locomotion policies:

1. **Blind:** Locomotion policy trained in rough terrains, and fine-tuned in the obstacle terrain with the leg contact penalty. Neither a terrain map nor vision is used.
2. **Rough:** Locomotion policy from [16] trained from scratch in rough terrains, without the leg contact penalty. This policy has access to a privileged terrain map for locomotion.
3. **ViNL (no pre-training):** Same as ViNL, but all training is done in the obstacle terrain, with no pre-training phase in the rough terrain. Egocentric depth observations are used (no terrain map).
4. **ViNL (MLP):** Same as ViNL, but the encoder is an MLP instead of an LSTM. This locomotion policy uses egocentric depth observations.

All baselines are trained for the same amount of experience, including steps used for fine-tuning. Note that the navigation policy is being tested out-of-distribution using a different simulator, novel environments, and unseen low-level controllers with no adaptation.

From Table 1, we see that **Blind** completely struggles (1.2% SR, row 1), demonstrating the importance of using exteroceptive sensors in cluttered environments. Because the robots are spawned in obstacle-free patches of terrain during training and evaluation, we find that **Blind** learns to crawl in circles near the starting position and largely ignore commanded velocities in favor of not crashing due to obstacle collision. This results in a higher distance traveled, despite a near-zero success rate. In comparison to **Rough**, we see that our method results in an average increase of 32.8% SR, and average decrease of 4.4 foot collisions per meter traveled (rows 2 and 5), despite the fact that **Rough** has access to a privileged terrain map. We find that **Rough** often gets its hind feet stuck while climbing over an obstacle, causing the robot to fall over and thus failing the episode. Next, we compare against **ViNL (no-pretraining)**, which was trained solely in the obstacle terrain. **ViNL (no-pretraining)** achieves a slightly higher success rate than **Rough** (+13.6% SR, rows 2 and 3), and fewer foot collisions per meter (-2.41 collisions). However, this method still has a lower success rate than **ViNL** (-19.2% SR, rows 3 and 5), and more foot collisions per meter (+2.01 collisions). This emphasizes the importance of pre-training in the rough terrain in stage 1 of our approach to avoid falling over (which leads to episode failure). We see that **ViNL (MLP)** performs just as well as **ViNL** for foot collision per meter (12.6 vs. 12.1, rows 4 and 5). However, **ViNL** outperforms **ViNL (MLP)** by +6.8% SR, which highlights the benefit of leveraging temporal information with an LSTM. Because **ViNL (MLP)** does not have any information about the whereabouts of objects that have left the robot's field-of-view, it cannot avoid obstacles that it tries to step over as adequately. This causes the robot to trip and fall over, leading to more episode failures.

4 Conclusion

We present ViNL, which enables a robot to intelligently navigate cluttered indoor environments using egocentric vision. While prior works focus on recovering from instability, or path planning around obstacles, we present a fully learned hierarchical approach that avoids these obstacles using fine-grained control of its legs.

References

[1] S. K. Ramakrishnan, A. Gokaslan, E. Wijmans, O. Maksymets, A. Clegg, J. M. Turner, E. Undersander, W. Galuba, A. Westbury, A. X. Chang, M. Savva, Y. Zhao, and D. Batra. Habitat-matterport 3d dataset (HM3d): 1000 large-scale 3d environments for embodied AI. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021. URL https://openreview.net/forum?id=-v4OuqNs5P.

[2] M. Savva, A. Kadian, O. Maksymets, Y. Zhao, E. Wijmans, B. Jain, J. Straub, J. Liu, V. Koltun, J. Malik, D. Parikh, and D. Batra. Habitat: A Platform for Embodied AI Research. 2019.

[3] A. Szot, A. Clegg, E. Undersander, E. Wijmans, Y. Zhao, J. Turner, N. Maestre, M. Mukadam, D. Chaplot, O. Maksymets, A. Gokaslan, V. Vondrus, S. Dharur, F. Meier, W. Galuba, A. Chang, Z. Kira, V. Koltun, J. Malik, M. Savva, and D. Batra. Habitat 2.0: Training home assistants to rearrange their habitat. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[4] B. Shen, F. Xia, C. Li, R. Martín-Martín, L. Fan, G. Wang, C. Pérez-D'Arpino, S. Buch, S. Srivastava, L. P. Tchapmi, M. E. Tchapmi, K. Vainio, J. Wong, L. Fei-Fei, and S. Savarese. igibson 1.0: a simulation environment for interactive tasks in large realistic scenes. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, page accepted. IEEE, 2021.

[5] F. Xiang, Y. Qin, K. Mo, Y. Xia, H. Zhu, F. Liu, M. Liu, H. Jiang, Y. Yuan, H. Wang, L. Yi, A. X. Chang, L. J. Guibas, and H. Su. SAPIEN: A simulated part-based interactive environment. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[6] M. Deitke, W. Han, A. Herrasti, A. Kembhavi, E. Kolve, R. Mottaghi, J. Salvador, D. Schwenk, E. VanderBilt, M. Wallingford, et al. Robothor: An open simulation-to-real embodied AI platform. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3164–3174, 2020.

[7] N. Yokoyama, S. Ha, and D. Batra. Success weighted by completion time: A dynamics-aware evaluation criteria for embodied navigation. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021.

[8] A. Kadian, J. Truong, A. Gokaslan, A. Clegg, E. Wijmans, S. Lee, M. Savva, S. Chernova, and D. Batra. Sim2real predictivity: Does evaluation in simulation predict real-world performance? 2020.

[9] J. Truong, M. Rudolph, N. 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 (CoRL)*, 2022.

[10] D. S. Chaplot, D. Gandhi, S. Gupta, A. Gupta, and R. Salakhutdinov. Learning to explore using active neural slam. In *International Conference on Learning Representations (ICLR)*, 2020.

[11] R. Partsey, E. Wijmans, N. Yokoyama, O. Dobosevych, D. Batra, and O. Maksymets. Is mapping necessary for realistic pointgoal navigation? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17232–17241, June 2022.

[12] Robothor challenge @ CVPR 2020. https://ai2thor.allenai.org/robothor/challenge/.

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

[14] A. Iscen, K. Caluwaerts, J. Tan, T. Zhang, E. Coumans, V. Sindhwani, and V. Vanhoucke. Policies modulating trajectory generators. In *Conference on Robot Learning*, pages 916–926. PMLR, 2018.

[15] 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.

[16] 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*, pages 91–100. PMLR, 2022.

[17] D. Chen, B. Zhou, V. Koltun, and P. Krähenbühl. Learning by cheating. In *Conference on Robot Learning (CoRL)*, 2019.

[18] S. Ross, G. J. Gordon, and J. A. Bagnell. No-regret reductions for imitation learning and structured prediction. *CoRR*, abs/1011.0686, 2010. URL http://arxiv.org/abs/1011.0686.

[19] P. Anderson, A. Chang, D. S. Chaplot, A. Dosovitskiy, S. Gupta, V. Koltun, J. Kosecka, J. Malik, R. Mottaghi, M. Savva, et al. On Evaluation of Embodied Navigation Agents. *arXiv preprint arXiv:1807.06757*, 2018.

[20] F. Xia, A. R. Zamir, Z. He, A. Sax, J. Malik, and S. Savarese. Gibson env: Real-world perception for embodied agents. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[21] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[22] E. Wijmans, A. Kadian, A. Morcos, S. Lee, I. Essa, D. Parikh, M. Savva, and D. Batra. DD-PPO: Learning near-perfect pointgoal navigators from 2.5 billion frames. In *International Conference on Learning Representations (ICLR)*, 2020.