

Learning Visual Locomotion with Cross-Modal Supervision

Antonio Loquercio*, Ashish Kumar*, Jitendra Malik

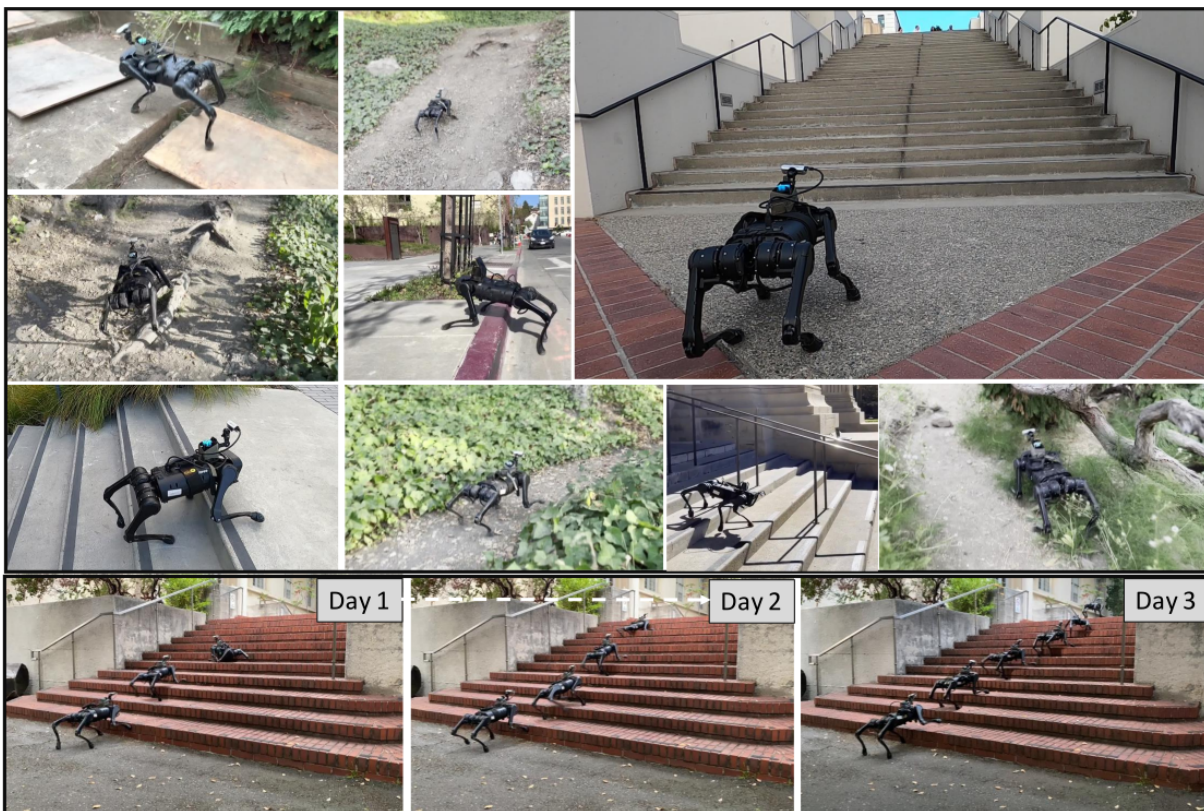


Fig. 1: The deployed walking policy shown above only uses a monocular egocentric RGB stream and proprioception. The terrains include stairs (up to 19cm high), curbs (up to 20cm high), slopes (35°), dirt roads, and unstructured construction sites. Several of these require precise foothold placement which is achieved by predicting the upcoming terrain using a visual lookahead module. This module is trained entirely in the real world. To do so, our proposed algorithm Cross-Modal Supervision(CMS) uses onboard proprioception to supervise a vision module. This naturally allows the policy to continually learn in the real world with its own experience. We show one such progression of continual learning at the bottom, where the policy goes from an initial success rate of 40% to 100% with less than 30 minutes of real-world data. Video results and code at https://antonilo.github.io/vision_locomotion/.

Abstract—In this work, we show how to learn a visual walking policy that only uses a monocular RGB camera and proprioception. Since simulating RGB is hard, we necessarily have to learn vision in the real world. We start with a blind walking policy trained in simulation. This policy can traverse some terrains in the real world but often struggles since it lacks knowledge of the upcoming geometry. This can be resolved with the use of vision. We train a visual module in the real world to predict the upcoming terrain with our proposed algorithm Cross-Modal Supervision (CMS). CMS uses time-shifted proprioception to supervise vision and allows the policy to continually improve with more real-world experience. We evaluate our vision-based walking policy over a diverse set of terrains including stairs (up to 19cm high), slippery slopes (inclination of 35°), curbs and tall steps (up to 20cm), and complex discrete terrains. We achieve this performance with less than 30 minutes of real-world data. Finally, we show that our policy can adapt to shifts in the visual

field with a limited amount of real-world experience.

I. INTRODUCTION

Gibson famously said that “We see in order to move and we move in order to see”. For land animals, if we interpret moving as walking, we might say “we see in order to walk and we walk in order to see”. Taken literally, this statement is surely false because obviously blind humans can walk. Blind robots can walk too [1], [2]. However Gibson wasn’t completely wrong either – while blind people can walk, they find it challenging to walk, and cannot walk as fast on complex terrains such as stairs. They have to probe gently with a walking stick first before they can start walking safely. Research on human vision [3], [4], [5], [6] shows that vision gives us “look-ahead” - when we can see the ground ahead of us, our foot placement is smoother and the walking more

* denotes equal contribution. All authors are affiliated to UC Berkeley

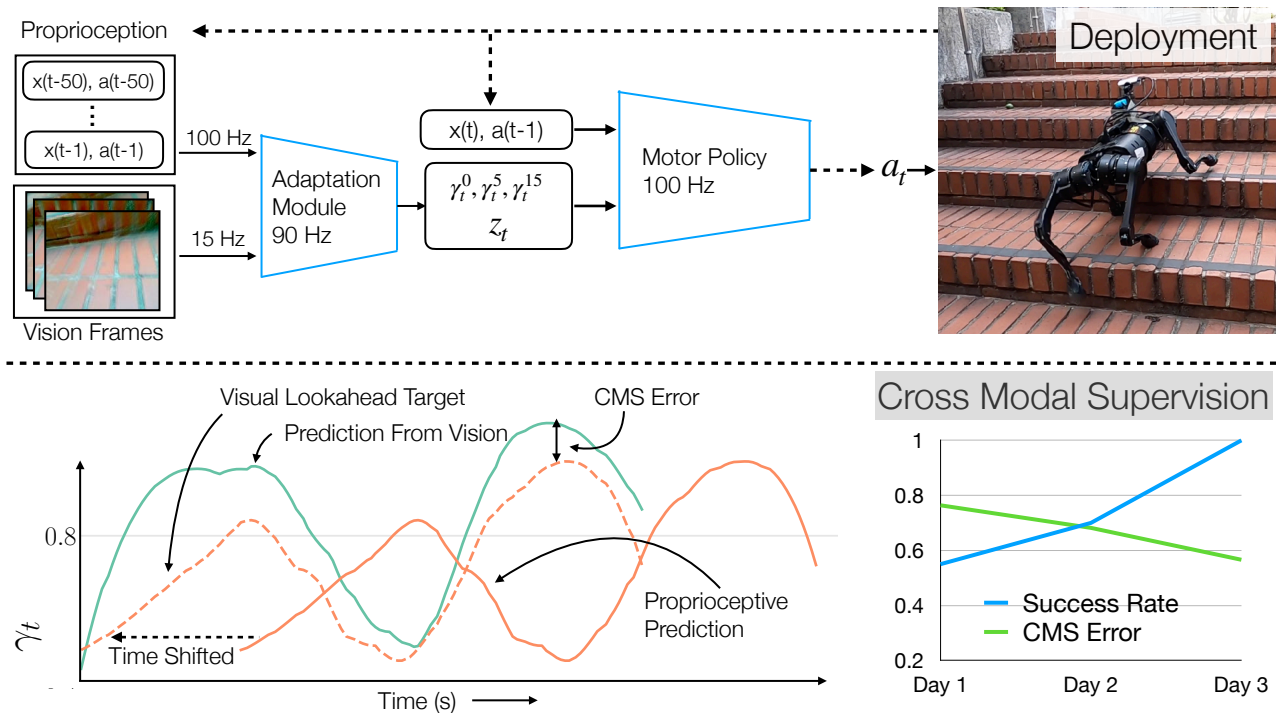


Fig. 2: **Top:** During deployment, we use an adaptation module to predict the extrinsic parameters z_t and an estimate of the terrain geometry below, 5cm, and 15cm in front of the robot ($\gamma^0, \gamma^5, \gamma^{15}$) from egocentric RGB and proprioception. **Bottom:** We train the visual module in the real world to predict a time-shifted proprioceptive estimate of the terrain’s slope by minimizing the CMS Error. We observe that CMS enables the adaptation module to improve continually with more real-world experience. On the bottom right we see that as the CMS error decreases the walking policy improves.

efficient. In addition, such look-ahead is tightly coupled to the walking strategy [7]. We study this question in a robotics context. This paper is about our development of walking policies for quadruped robots, using the visual system for look-ahead— predicting the height of the terrain in front of the robot before it gets there. Unsurprisingly, we found that policies with look-ahead indeed do better than blind ones (see Figure 1), thus validating the spirit, if not the letter of Gibson’s statement.

A vision sensor in a robotics context could be a single RGB camera, or one of the many different devices directly capturing depth using stereopsis or LIDAR for instance. Our decision was to go for RGB cameras since (1) there is evidence from human vision that stereo input contributes relatively little to making walking faster/more efficient [6] (2) for objects in terrestrial perspective, priors enable one to make reasonable depth estimates even from a single image [8] (3) the multiple views captured by a moving robot could in theory be exploited to recover depth directly as well [9]. We don’t wish to argue our point too strongly – there has been passionate but not necessarily productive debate on this in the context of self-driving cars already – but just say that the RGB version is scientifically interesting and there might also be engineering contexts where it is favored (cost, power, passive sensing). From a research novelty perspective, there are several systems which exploit depth input for guiding walking [10], [11], whereas the RGB only design choice has hardly been studied.

The RGB only choice leads to a technical challenge. The

most popular techniques for training walking policies with RL typically train in simulation and then deploy zero shot in the real world [1], [2]. This works out fine with depth inputs because the sim-to-real gap for depth is small, and multiple systems have been successfully demonstrated for walking and flying [12], [10], [11]. However rendered RGB images in a simulator look very different from those in the real world. The higher the fidelity desired in the rendering, the greater the computational cost, which makes it practically infeasible to use in an inner loop for training RL policies because of their high sample complexity.

Our core insight is that we can train the vision part of our policy in the real world from onboard sensors, while still managing to train the action policy in a simulator. In a simulator we can train policies which don’t use vision but "cheat" - they have access to look-ahead information of terrain height at a few "look-ahead" points ahead of the robot - and then it is a separate problem to train a vision system from real world experience which can predict the terrain height at those look-ahead points. This vision system is trained using cross-modal supervision (CMS) from proprioception with a time lag - I see a point A ahead of me whose height is currently unknown, but when my feet get to A, then its height can be inferred from the joint angles of my body. There is a "chicken-and-egg" aspect to this - to develop a vision system which can provide look-ahead for walking, you need to be able to walk first to get the data to provide the supervision for the vision system! Fortunately, there is no infinite regress

problem here, as the robot can walk (clumsily of course) when blind, so the bootstrapping can start from there.

Experiments show that our policy can walk over a diverse set of terrains including stairs (up to 19cm high), slippery slopes (inclination of 35°), curbs and tall steps (up to 20cm), and complex discrete terrains from a single onboard RGB camera and proprioception. To systematically understand the improvements over the initially deployed policy, we take 4 real-world stair cases on which the initial blind policy starts off at around 50%, and compare the performance of the visual walking policy as it improves with experience. We find that in all these cases, the visual walking policy reaches 100% performance with less than 30 minutes of data, collected across 4 days.(Figure 4 and Section IV).

Since the proposed algorithm, CMS (Cross-Modal Supervision), enables learning in the real world from onboard sensors our system can continue to learn lifelong. The experience collected online is used continually to improve the visual system, which in turn, improves the performance of the overall policy. We systematically evaluate this aspect of CMS in 4 challenging setups in the real world and show significant and consistent improvements in performance with less than 7 minutes of data each day. We also analyze the qualitative change in behavior of the visual terrain predictor compared to the proprioceptive estimator (Section IV) and see precise foot placement with vision. Finally, we perform the widely-known prism experiment [13] to demonstrate the visual plasticity of the learned model: our experiments indicate that with very limited data the model can adapt to large shifts in the visual field. Video results can be seen at https://antonilo.github.io/vision_locomotion/.

II. OVERVIEW

We want a visual walking policy that can walk on complex terrains. We train adaptive walking policies similarly to RMA [1], but with the following significant change – we have an additional module that estimates the upcoming terrain geometry from the last three egocentric frames. More concretely, the policy π takes a) an estimate of terrain in front and below the robot γ for visual adaptation to complex terrain, b) an estimate of the environment parameters z to enable adaptation to variations in the environment (similar to RMA [1]) the current proprioceptive state \mathbf{x} , and uses them to predict the target actions \mathbf{a} . This system is shown in the top part of Figure 2. Once we have such a system deployed, it collects egocentric visual data, as well as proprioception. This multi-modal data can be used for Cross-Modal Supervision (CMS) to continually improve the visual estimates of the terrain parameters γ . This is enabled by the key insight that proprioception can get an accurate estimate of the same terrain that the egocentric camera records ahead of time. This allows using proprioceptive estimates of terrain to supervise the visual estimate, as shown in the bottom half of Figure 2. Specifically, we train a vision module to predict the future proprioceptive estimate of terrain from the egocentric visual input (that looks in front of the robot) by minimizing the CMS error. Since the supervision comes from onboard sensors, we

continually do so to get an increasingly better policy with more experience.

We initialize the CMS improvement procedure from a blind policy π_{blind} , which can be trained in simulation and transferred to the real world. This policy has no visual lookahead and uses proprioception to estimate the terrain under the feet of the robot as the robot walks. Once we have this policy, we can deploy it in the real world, and train a module for visual terrain lookahead using CMS. Once we have this visual predictor of the upcoming terrain, we can then use the lookahead policy π (also trained in simulation) to get a visual walking policy in the real world.

III. METHOD

A. Simulation Training

We train two walking policies (π and π_{blind}) in simulation to walk on complex terrains. Both the policies take the state \mathbf{x}_t , the extrinsics vector \mathbf{z}_t , and the terrain information γ_t to predict the target joint angles \mathbf{a}_t at 100Hz which is converted to torques using a PD controller. The lookahead policy π takes an additional input $\gamma_{t+\Delta t}$, which is the terrain estimate at 15cm lookahead in front of the robot. Concretely,

State Input. $\mathbf{x}_t = [\mathbf{q}_t, \mathbf{q}'_t, \mathbf{a}_{t-1}, \mathbf{z}_t, \gamma_t]^\top$, where $\mathbf{q}_t \in \mathbb{R}^{12}$ is the current joint position, $\mathbf{q}'_t \in \mathbb{R}^{12}$ the joint velocity, $\mathbf{a}_{t-1} \in \mathbb{R}^{12}$ the previous action.

Extrinsics Vector. $\mathbf{z}_t \in \mathbb{R}^8$ is a latent representation of the environment parameters which includes payload, motor strength, binarized foot contacts, linear velocity and friction.

Terrain Geometry Vector. γ_t is a latent representation of the ground truth terrain geometry \mathbf{h}_t under the feet of the robot, and $\gamma_{t+\Delta t}$ is the terrain estimate 15cm in front of the robot. This gives us:

$$\mathbf{z}_t = \mu(\mathbf{e}_t) \quad (1)$$

$$\gamma_t = \delta(\mathbf{h}_t) \quad (2)$$

$$\gamma_{t+\Delta t} = \delta(\mathbf{h}_{t+\Delta t}) \quad (3)$$

$$\mathbf{a}_t = \pi(\mathbf{x}_t, \mathbf{z}_t, \gamma_t, \gamma_{t+\Delta t}) \quad (4)$$

$$\mathbf{a}_t = \pi_{blind}(\mathbf{x}_t, \mathbf{z}_t, \gamma_t), \quad (5)$$

where μ , π and π_{blind} are MLPs with two hidden layers and [256, 128] and [128, 128] neurons, respectively. Note that we use the same network δ (hidden dimensions of [64, 16]) to process γ_t and $\gamma_{t+\Delta t}$. We train the π_{blind} and the dependent encoders jointly in an end-to-end manner using model-free reinforcement learning. Once trained, we freeze the network δ and μ , and train policy π using model-free RL. We now explain the environment design and the reward structure we use for RL which is shared between both the policies.

Environment Design. We recreate in simulation a set of natural conditions to elicit robust locomotion. Specifically, we train on fractal terrains (similar to [1], [14]) and parameterized inclines and stairs, which typically represent a majority of the commonly encountered terrains in human-made environments. To design the distribution of staircases, we take inspiration from a classic reference in the field of architecture [15]. According to the latter, to be comfortable for

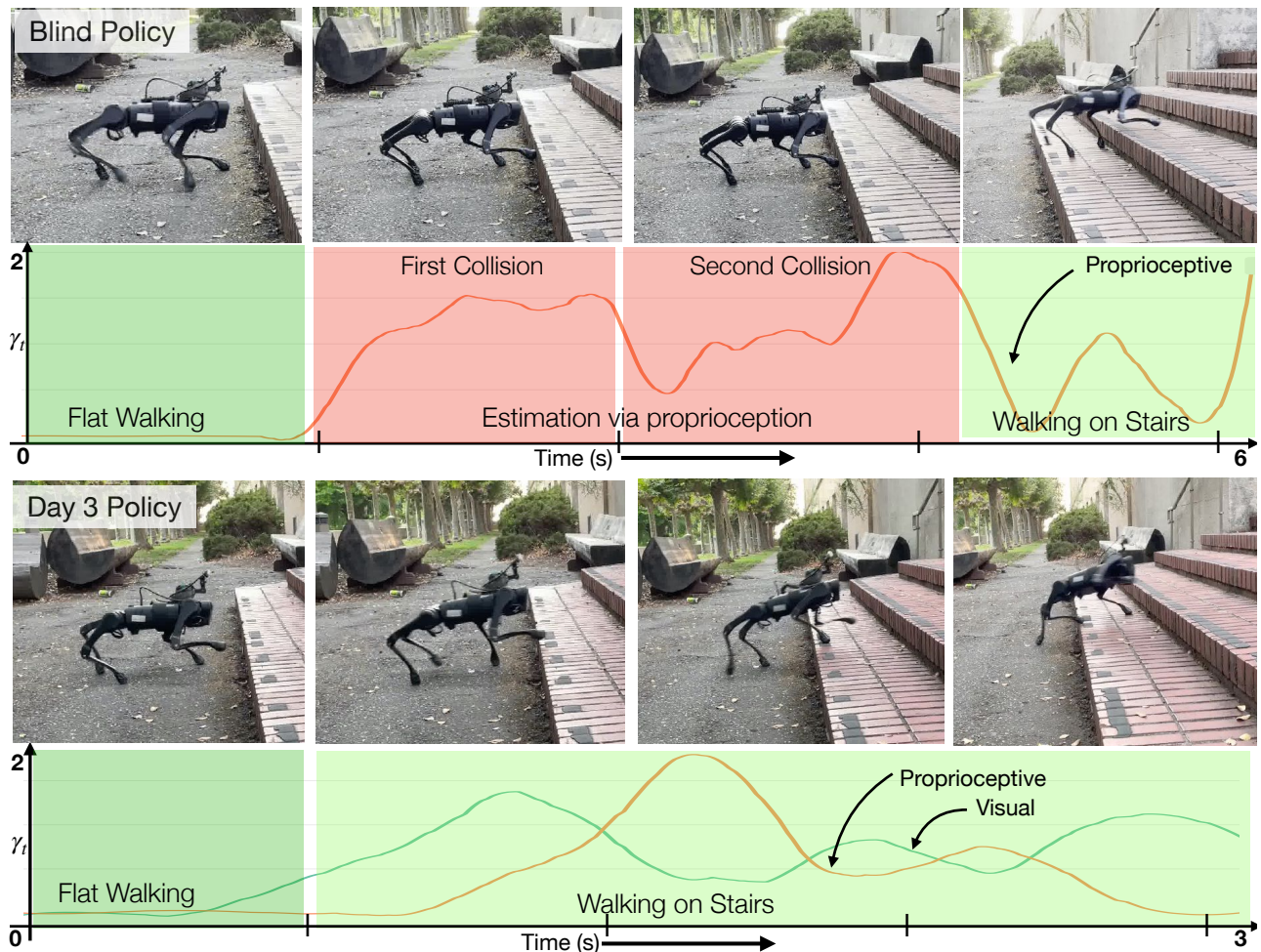


Fig. 3: **Behavior Analysis:** The top row shows a blind policy, which needs to touch the stairs multiple times before correctly estimating its height and climbing over it. In contrast, a vision-based policy trained to predict the future proprioceptive estimate of terrain can anticipate changes in the terrain geometry and adapt its gait accordingly. This speeds up the initial climb by approximately two times and gives an overall higher success rate.

humans, staircases should have height in the range [10, 19]cm and lengths not smaller than 30cm. In addition, two heights and one length should make for the length of a step, which is approximately 1mt. We use these two recommendations to build our parameterized staircases: we uniformly sample step heights in the range [10, 21]cm and lengths from the set [30, 40, 50, 60]cm.

RL Rewards. We design a reward function to promote the agent to move with a user-defined forward and angular speed in the ranges $v_x^d \in [0, 0.5]$ m/s and $w_z^d \in [-0.4, 0.4]$ rad/s, while penalizing lateral speed and jerky motions. We define v as linear velocity, ω as the angular velocity, α and $\dot{\alpha}$ as joint angles and velocities, τ as joint torques, v^f the feet velocity and g the binary foot contact indicator. Accordingly, we define the reward at time t as the sum of the following terms:

- Forward: $\min(v_x^d, v_x)$
- Lateral: $\|v_y\|$
- Angular: $-\|w_z^d - w_z\| + w_x^d$
- Work: $-\|\tau^T \cdot (\alpha_t - \alpha_{t-1})\|$
- Foot Slip: $-\|\text{diag}(g)_t \cdot v_t^f\|$

The scaling of each of the previous factors is 65, 1, 40, 0.05, 0.2. We additionally add a survival bonus of 9, which is doubled whenever the robot is tasked with walking on a staircase.

Curriculum. We start by training exclusively on flat terrain for 500M steps. Afterwards, we train on progressively difficult staircases: starting from heights of 10cm and gradually increasing the step height every 100M steps until it reaches the maximum. During training on staircases, we sample flat terrains with 10% probability to avoid catastrophic forgetting. To guide the optimization, we additionally add a regression loss to match the actions from a policy exclusively trained on flat ([14]) We randomly sample the environment parameters such as payload, friction, and motor strength from the ranges reported in [1].

Sim to Real Transfer. To transfer policy π_{blind} to the real world, we train a network g_0 to predict z_t and γ_t from proprioception and action history. We use the same structure and training procedure for g_0 as proposed by RMA [1].

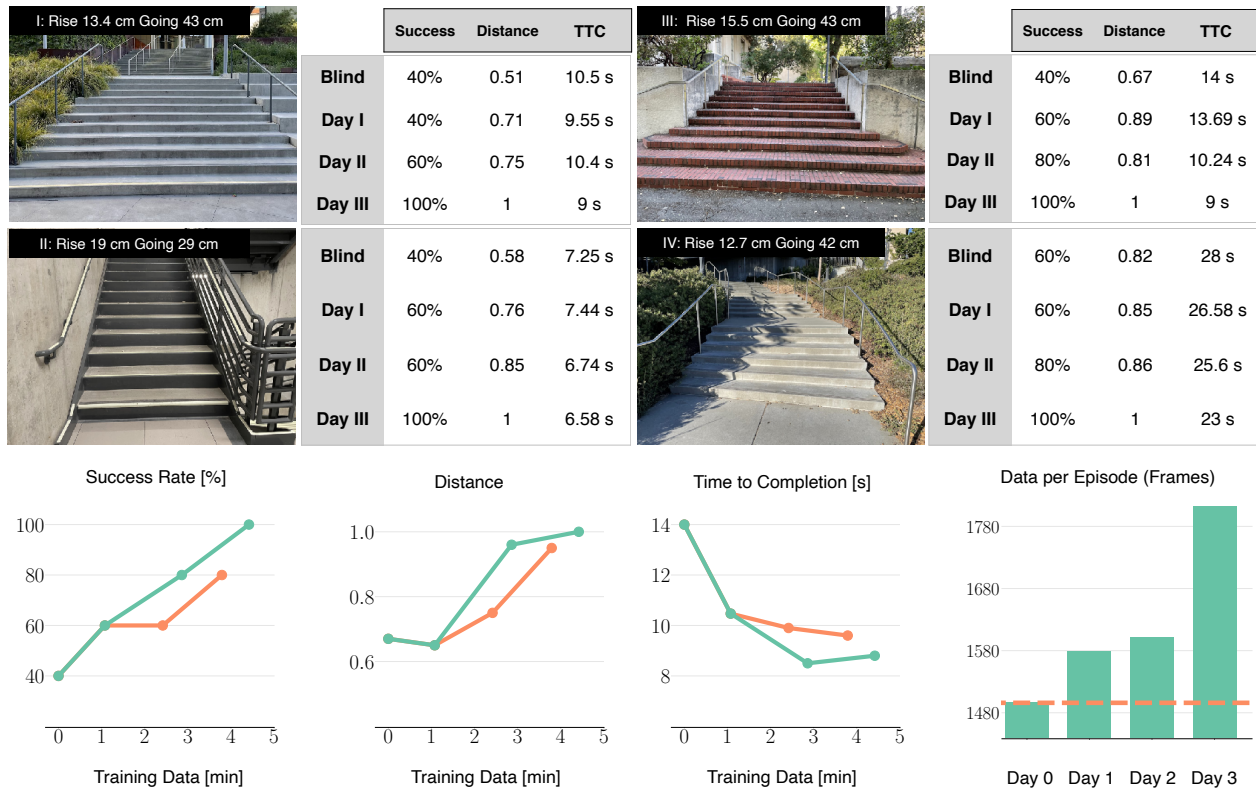


Fig. 4: **Real-World Improvement:** (Top Row) We find that using CMS continually improves the performance of the walking policy, from approximately 50% to 100% in all the settings considered above. This is enabled by the use of vision, which is trained in the real world with CMS. The final policy uses less than 30 minutes of data collected over 4 days in different conditions. (Bottom Row) We observe that using the latest available vision policy (green) to collect data, instead of the initial blind policy (orange), leads to faster convergence of the final visual policy. This shows that the data collected with the latest visual policy is of higher quality. Moreover, we observe that since the vision policy can walk longer without falling, it significantly improves the efficiency of data collection in the real world.

B. Vision-Based Locomotion via CMS

Once π_{blind} is deployed in the real world, we collect a dataset of trajectory rollouts in the real world $\mathcal{D} = ([I_t, \mathbf{x}_t], \gamma_t)$, where I_t is the egocentric visual input from the onboard camera, \mathbf{x}_t is the current proprioceptive state and γ_t is the current geometry estimate. We can now use this dataset to train a convolutional neural network g_i to estimate the current extrinsics \mathbf{z}_t and the future terrain geometry $\gamma_{t+\Delta t}$ using supervised learning as shown in Figure 2. Having a visual predictor of upcoming terrain we can now use the lookahead policy π which has better performance than the blind policy.

C. Lifelong Learning via CMS

CMS naturally allows for lifelong learning. Since we continuously collect the dataset \mathcal{D} with real-world experience, we can use the γ estimate from proprioception to continuously train the visual lookahead terrain predictor during execution and improve performance.

IV. EXPERIMENTAL SETUP

Simulation Setup: We use the RaiSim simulator [16] for rigid-body and contact dynamics simulation. We import the AI URDF file from Unitree. Each RL episode lasts for a

maximum of 1200 steps with early termination if the roll or pitch exceeds a threshold, or the base of the robot is very close to the ground. The control frequency of the policy is 100 Hz, and the simulation time step is 0.025s.

Vision-Based Estimator. We build a custom model to predict the future γ to allow inference on limited compute. The input of the model are the latest three frames (camera frequency is 15Hz), converted to grayscale and concatenated to make an input tensor. We compute features from such tensor with a shufflenet-V2 model [17]. We use as features the last layer before global-average pooling to maintain spatial information. The features are then projected to a 2dim channel space with 1D convolutions and processed by an MLP with hidden dimension of $[128, 64]$ to predict γ . We additionally provide a history of 50 IMU measurements (roll and pitch) and desired velocity commands to the predictor. This network outputs a 128 dimensional embedding which is later concatenated with the shufflenet features and passed through the final MLP.

V. RESULTS AND ANALYSIS

We design an evaluation procedure to answer three main questions: (i) What are the quantitative and qualitative benefits of vision-based locomotion over blind locomotion in the real

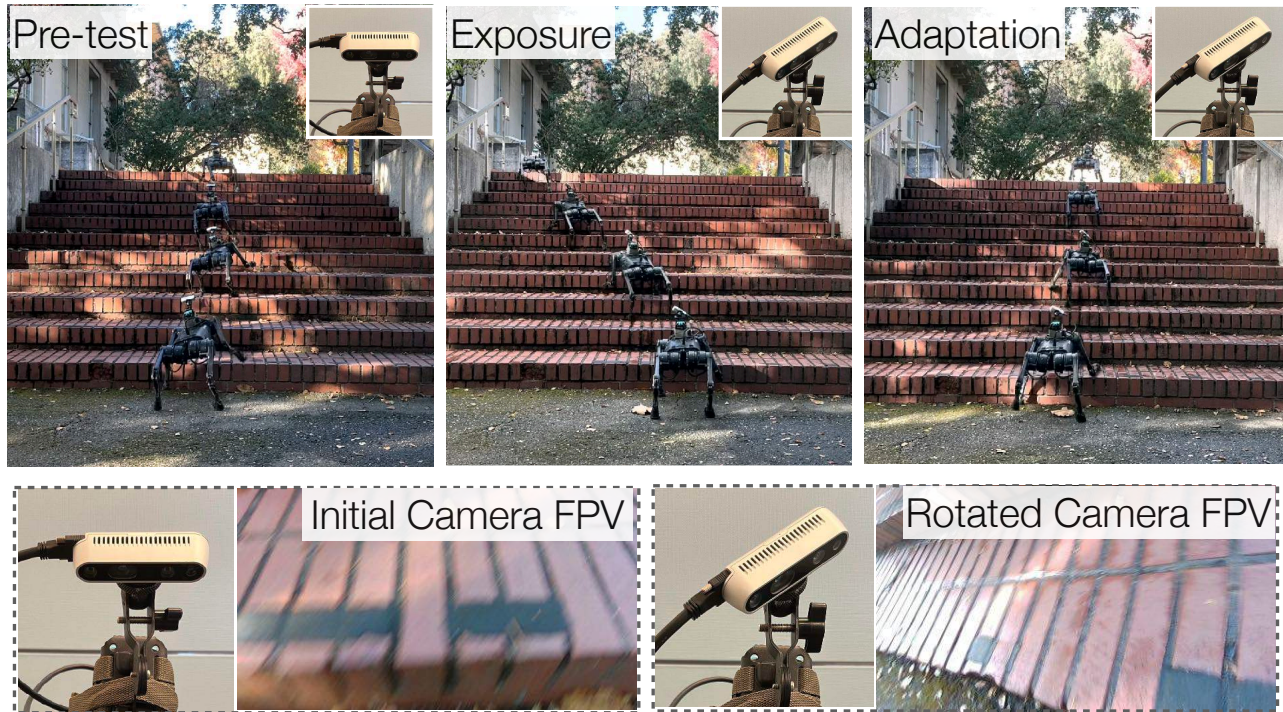


Fig. 5: **Prism-Adaptation Test:** We perform the widely-known prism test [18] on our robot. In this test, the visual field of the subject is altered by using a prism in front of the eyes. To simulate this, we rotate the camera from its nominal position. (Bottom Row) The left camera is the nominal position, and the right camera is the one after rotation. The view from the camera is shown next to the camera orientation. Despite the robot’s position being the same, the visual appearance changes drastically between the two camera configurations. (Top Row) The left image shows the visual walking policy under nominal conditions (*pre-test*), the center image is the performance of this visual walking policy immediately after the camera is rotated (*exposure*), and the right image shows the performance after training for 80 seconds of experience (*adaptation*). The behavior recovers with just 1 minute of finetuning on the new visual input.

world? (ii) During the continual training in the real world, what benefit does deploying the latest available policy have over collecting all the data with a blind policy and using it at once? (iii) What is the quantitative benefit of having a visual lookahead on terrain geometry?

A. Lifelong Vision Learning

A blind policy walks very differently than a vision-based policy. In Figure 3 we analyze the qualitative change in behavior as a consequence of improved terrain prediction by plotting the estimated γ as a function of time for the blind policy and the visual walking policy. The blind policy is clumsy but "exploratory", tapping the terrain twice before walking on it. Conversely, the vision policy can anticipate the geometry of the terrain and adapt its stride accordingly. This results in a smooth and agile gait, enabling the robot to start climbing the stairs without stumbling and speeding up the initial climb by approximately two times.

The quality of the vision predictor improves with experience. We study the dependence between experience and performance in Fig. 4. We select four stairs of different heights and lengths and evaluate the performance over multiple iterations of data collection. For each policy, we perform five trials on each stair. We define a "day" as the policy trained with all data collected by its predecessors. All the

metrics constantly improve as a function of time. A policy trained on three "days" of experience, or more specifically on 26 minutes of walking data, can climb all stairs without a single failure. Overall, the vision policy can walk faster (average speed is 26% higher) and more effectively (the success rate is 50 percentage points higher) than its blind counterpart.

Since the visual walking policy can go longer without falling it collects data more efficiently. As shown in Fig. 4, the amount of experience collected on average during a real-world experiment increases by 21%. In addition, the quality of the data collected using vision is of superior quality. For approximately the same amount of data, the policy trained on data collected exclusively from the blind controller performs worse than the one trained in an iterative fashion. Intuitively, this is due to the fact that the blind controller occasionally stumbles or taps the terrain before taking a step, which results in corrupted data. Conversely, a (possibly intermediate) vision controller is more precise in its gait, leading to higher quality data.

Generalization Experiments. We test the generalization of our final vision policy on several challenging terrains, as shown in Figure 1. The terrains include previously unseen staircases, steep inclines (up to 35 degree steep), discrete terrains, and curbs. Our approach was successful on 80%

of previously unseen staircases, 100% of curbs, and 60% percent of inclines, which is remarkable since the vision module was never trained for them. All generalization videos are available at https://antonilo.github.io/vision_locomotion/.

Visual Plasticity. The prism test is a widely-known experiment designed to test adaptation to an artificial shift of the visual field [18]. This test is generally performed on patients to improve the spatial deficits caused by brain damage, e.g., after a stroke [13]. A prism adaptation session consists of three phases: the *pre-test*, where a subject performs a task without any disturbance of the visual field; the *exposure*, where the subject performs the same task under an horizontal shift of the visual field (generally wearing prism wedges); and the *adaptation*, where the subject adapts to the new visual field and can perform the task at pre-test levels. Ideally, the subject should complete the task perfectly in the pre-test phase and quickly adapt to the misalignment of the visual and proprioceptive maps caused by the prism wedges.

We perform a similar test to measure the adaptation ability of our visual locomotion policy. However, instead of adding prism wedges, we shift the visual field by rotating the camera on its yaw axis of approximately 30 degrees (Fig. 5). Such shift makes the terrain in front of the robot almost unobservable. Immediately after altering the robot’s visual field, the policy stumbles on the steps and accumulates drift while climbing stairs. To account for such vision impairment, we fine-tune the vision policy by minimizing the CMS error. Specifically, after each trial in the modified setup, we finetune only the last 3 layers of gamma predictor for 10 epochs. After just three trials (roughly corresponding to 80 seconds of data), the vision policy learns to account for the systematic bias in the visual field and manages to correctly anticipate stairs and walk straight (Fig. 5).

Finally, we bring the robot’s camera back to its initial position and repeat the task. In this phase, generally referred to as *post-test* [13], the robot is required to quickly adjust back to its original visual field. We repeat the same procedure as above and observe that also in this case, after two trials, the policy adapts back to its original visual field. We invite the reader to visit our project web-page for a video of the experiment.

B. The value of visual lookahead

We compare in simulation the performance of policies with access to a larger ground-truth terrain lookahead. Table I shows the results of this evaluation. These results indicate a sharp increase in performance (from 58% to 75%) when look ahead increases from 0 to 15cm.

Look Ahead	Success (↑)	TTF (↑)	Distance (m) (↑)	Smoothness (↓)
0cm (RMA [1])	58%	0.83	4.11	34.19
5cm	64%	0.88	4.67	44.30
15cm	75%	0.95	4.73	30.77
25cm	74%	0.96	4.96	37.19
35cm	74%	0.94	4.66	31.43

TABLE I: Performance of motor policies with access to different ground-truth visual lookahead. The value of lookahead saturates after 15cm.

VI. RELATED WORK

A. Blind Locomotion

Traditionally, locomotion controllers were model-based [19], [20], [21], [22], [23], [24], [25]. However, these methods require accurate modeling of the robot and substantial task-specific engineering for tuning locomotion gaits and behaviors. While some of these problems could be mitigated with optimization-based controllers [26], [27], these methods tend to become brittle outside of lab-controlled conditions [28], [2]. While using learning for locomotion has a long history [29], [30], recent deep reinforcement learning (RL) controllers achieved impressive results in real-world rough terrains [2], [1]. Being proprioception relatively low dimensional, training blind controllers can be done within minutes in highly optimized simulators [31]. Independently of the training procedure, blind controllers can only react to varying terrain conditions, but not anticipate them. This results in clumsy locomotion behaviours, especially when the terrain has discontinuities.

B. Vision-Based Locomotion

Using vision unlocks locomotion on complex terrain. However, developing such a system is challenging because of the high dimensionality of images. These challenges pushed prior work to favour modular systems to end-to-end ones, essentially dividing the locomotion task in a vision and control sub-task. Vision observations are generally used to build a local elevation map [32], [33], [34], [11], [35], [36]. Such maps are used to predict motion primitives (in terms of goal velocities [35], foothold placement [32], [33], [34] or leg motion phase [11]), which are tracked by a separate controller module. A parallel line of work aims to directly predict navigation commands (either in terms of foothold placement of joint velocities) from on-board depth observations [37], [38], [39]. Overall, the aforementioned methods achieve impressive performance in simulation, but did not yet obtain a similar performance in real-world conditions. One exception is the recent work of Miki et al. [11], which obtains impressive results in real-world conditions. However, such method has very high sensing requirements (up to three lidars or eight stereo modules) and computational costs, making it unfeasible on smaller scale robots as the one considered in this work.

C. Self-supervised learning

Early works in self-supervised learning for locomotion mainly used haptic feedback to identify the terrain [40], [41], [42], [43], [44]. Such information can be used to bias motion over easily traversable environments [41], [45]. Another line of works estimate traversability information (e.g. probability of falling) from local elevation maps and used it for path planning. The training data for such estimators is generally collected through experience in simulation [46], [47], [48]. Similar ideas have also been investigated on other robotic platforms [49], [50], [47]. Overall, the aforementioned works generally operate on an high-level abstraction (traditionally, a trajectory), generally tracked by a separate low-level controller module. In contrast, our work aims to improve the

quality of the low-level controller via self-supervised learning, practically increasing the range of traversable terrain through experience.

VII. CONCLUSION

We propose CMS, a technique which uses onboard data collected from multiple sensory inputs to continually improve its performance. We demonstrate this idea for the task of visual walking, using time-shifted proprioception for supervision. Our approach show significant improvements in performance with less than 7 minutes of data per run. One limitation of the current system is that it does not improve the motor system in the real world. An interesting venue for future work is to explore the coupling of techniques that additionally improve the motor policy along with CMS for improving the visual system.

VIII. ACKNOWLEDGEMENT

This work was supported by the DARPA Machine Common Sense program and by the ONR MURI award N00014-21-1-2801. We would like to thank Sasha Sax for the helpful discussions, Noemi Aepli for support with media material, and Haozhi Qi for support with the website creation.

REFERENCES

- [1] A. Kumar, Z. Fu, D. Pathak, and J. Malik, "Rma: Rapid motor adaptation for legged robots," in *Robotics: Science and Systems*, 2021.
- [2] 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.
- [3] A. E. Patla, "Understanding the roles of vision in the control of human locomotion," *Gait & posture*, vol. 5, no. 1, pp. 54–69, 1997.
- [4] A. Patla, "Strategies for dynamic stability during adaptive human locomotion," *IEEE Engineering in Medicine and Biology Magazine*, vol. 22, no. 2, pp. 48–52, 2003.
- [5] J. M. Loomis, J. A. Da Silva, N. Fujita, and S. S. Fukushima, "Visual space perception and visually directed action," *Journal of experimental psychology: Human Perception and Performance*, vol. 18, no. 4, p. 906, 1992.
- [6] K. Bonnen, J. S. Matthis, A. Gibaldi, M. S. Banks, D. M. Levi, and M. Hayhoe, "Binocular vision and the control of foot placement during walking in natural terrain," *Scientific reports*, vol. 11, no. 1, pp. 1–12, 2021.
- [7] K. S. Kretch, J. M. Franchak, and K. E. Adolph, "Crawling and walking infants see the world differently," *Child Development*, vol. 85, no. 4, pp. 1503–1518, dec 2013.
- [8] R. Ranftl, K. Lasinger, D. Hafner, K. Schindler, and V. Koltun, "Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 3, 2022.
- [9] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*. Cambridge university press, 2003.
- [10] A. Agarwal, A. Kumar, J. Malik, and D. Pathak, "Legged locomotion in challenging terrains using egocentric vision," 2022. [Online]. Available: <https://openreview.net/pdf?id=Re3NjSfwf0WF>
- [11] 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.
- [12] A. Loquercio, E. Kaufmann, R. Ranftl, M. Müller, V. Koltun, and D. Scaramuzza, "Learning high-speed flight in the wild," *Science Robotics*, vol. 6, no. 59, p. eabg5810, 2021.
- [13] A. Barrett, K. M. Godert, and J. C. Basso, "Prism adaptation for spatial neglect after stroke: translational practice gaps," *Nature Reviews Neurology*, vol. 8, no. 10, p. 567, 2012.
- [14] Z. Fu, A. Kumar, J. Malik, and D. Pathak, "Minimizing energy consumption leads to the emergence of gaits in legged robots," 2021.
- [15] J. Templer, *The Staircase: Studies of Hazards, Falls, and Safer Design*. The MIT Press/National Endowment for the Humanities/Andrew W. Mellon Foundation Humanities Open Book Program., 03 1995. [Online]. Available: <https://doi.org/10.7551/mitpress/6434.001.0001>
- [16] J. Hwangbo, J. Lee, and M. Hutter, "Per-contact iteration method for solving contact dynamics," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 895–902, 2018. [Online]. Available: www.raisim.com
- [17] N. Ma, X. Zhang, H.-T. Zheng, and J. Sun, "Shufflenet v2: Practical guidelines for efficient cnn architecture design," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 116–131.
- [18] H. Von Helmholtz, *Helmholtz's treatise on physiological optics*. Optical Society of America, 1925, vol. 3.
- [19] J. Buchli, M. Kalakrishnan, M. Mistry, P. Pastor, and S. Schaal, "Compliant quadruped locomotion over rough terrain," in *2009 IEEE/RSJ international conference on Intelligent robots and systems*. IEEE, 2009, pp. 814–820.
- [20] A. D. Ames, K. Galloway, K. Sreenath, and J. W. Grizzle, "Rapidly exponentially stabilizing control lyapunov functions and hybrid zero dynamics," *IEEE Transactions on Automatic Control*, vol. 59, no. 4, pp. 876–891, apr 2014.
- [21] M. Khoramshahi, H. J. Bidgoly, S. Shafiee, A. Asaei, A. J. Ijspeert, and M. N. Ahmadabadi, "Piecewise linear spine for speed–energy efficiency trade-off in quadruped robots," *Robotics and Autonomous Systems*, vol. 61, no. 12, pp. 1350–1359, dec 2013.
- [22] D. J. Hyun, J. Lee, S. Park, and S. Kim, "Implementation of trot-to-gallop transition and subsequent gallop on the MIT cheetah i," *The International Journal of Robotics Research*, vol. 35, no. 13, pp. 1627–1650, jul 2016.
- [23] C. D. Bellicoso, F. Jenelten, P. Fankhauser, C. Gehring, J. Hwangbo, and M. Hutter, "Dynamic locomotion and whole-body control for quadrupedal robots," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 3359–3365.
- [24] M. Kalakrishnan, J. Buchli, P. Pastor, M. Mistry, and S. Schaal, "Learning, planning, and control for quadruped locomotion over challenging terrain," *The International Journal of Robotics Research*, vol. 30, no. 2, pp. 236–258, 2011.
- [25] K. Byl and R. Tedrake, "Dynamically diverse legged locomotion for rough terrain," in *2009 IEEE International Conference on Robotics and Automation*, 2009, pp. 1607–1608.
- [26] C. Gehring, S. Coros, M. Hutter, C. Dario Bellicoso, H. Heijnen, R. Diethelm, M. Bloesch, P. Fankhauser, J. Hwangbo, M. Hoepflinger, and R. Siegwart, "Practice makes perfect: An optimization-based approach to controlling agile motions for a quadruped robot," *IEEE Robotics Automation Magazine*, vol. 23, no. 1, pp. 34–43, 2016.
- [27] R. Calandra, A. Seyfarth, J. Peters, and M. P. Deisenroth, "Bayesian optimization for learning gaits under uncertainty," *Annals of Mathematics and Artificial Intelligence*, vol. 76, no. 1-2, pp. 5–23, jun 2015.
- [28] J. Hwangbo, J. Lee, A. Dosovitskiy, D. Bellicoso, V. Tsounis, V. Koltun, and M. Hutter, "Learning agile and dynamic motor skills for legged robots," *Science Robotics*, vol. 4, no. 26, p. eaau5872, 2019.
- [29] J. Z. Kolter and A. Y. Ng, "The stanford LittleDog: A learning and rapid replanning approach to quadruped locomotion," *The International Journal of Robotics Research*, vol. 30, no. 2, pp. 150–174, jan 2011.
- [30] M. Zucker, N. Ratliff, M. Stolle, J. Chestnutt, J. A. Bagnell, C. G. Atkeson, and J. Kuffner, "Optimization and learning for rough terrain legged locomotion," *The International Journal of Robotics Research*, vol. 30, no. 2, pp. 175–191, 2011.
- [31] 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.
- [32] V. Tsounis, M. Alge, J. Lee, F. Farshidian, and M. Hutter, "Deepgait: Planning and control of quadrupedal gaits using deep reinforcement learning," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3699–3706, 2020.
- [33] A. Agrawal, S. Chen, A. Rai, and K. Sreenath, "Vision-aided dynamic quadrupedal locomotion on discrete terrain using motion libraries," *arXiv preprint arXiv:2110.00891*, 2021.
- [34] 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.
- [35] Z. Fu, A. Kumar, A. Agarwal, H. Qi, J. Malik, and D. Pathak, "Coupling vision and proprioception for navigation of legged robots," *arXiv preprint arXiv:2112.02094*, 2021.

- [36] H.-W. Park, P. M. Wensing, S. Kim *et al.*, “Online planning for autonomous running jumps over obstacles in high-speed quadrupeds,” *Robotics: Science and Systems*, 2015.
- [37] G. B. Margolis, T. Chen, K. Paigwar, X. Fu, D. Kim, S. bae Kim, and P. Agrawal, “Learning to jump from pixels,” in *5th Annual Conference on Robot Learning*, 2021. [Online]. Available: <https://openreview.net/forum?id=R4E8wTUtxdl>
- [38] 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. [Online]. Available: <https://openreview.net/forum?id=NDYbXf-DvwZ>
- [39] R. Yang, M. Zhang, N. Hansen, H. Xu, and X. Wang, “Learning vision-guided quadrupedal locomotion end-to-end with cross-modal transformers,” in *International Conference on Learning Representations*, 2022. [Online]. Available: <https://openreview.net/forum?id=nhnJ3oo6AB>
- [40] M. A. Hoepflinger, C. D. Remy, M. Hutter, L. Spinello, and R. Siegwart, “Haptic terrain classification for legged robots,” in *2010 IEEE International Conference on Robotics and Automation*. IEEE, 2010, pp. 2828–2833.
- [41] K. Walas, “Terrain classification and negotiation with a walking robot,” *Journal of Intelligent & Robotic Systems*, vol. 78, no. 3, pp. 401–423, 2015.
- [42] A. Valada and W. Burgard, “Deep spatiotemporal models for robust proprioceptive terrain classification,” *The International Journal of Robotics Research*, vol. 36, no. 13-14, pp. 1521–1539, 2017.
- [43] M. Hoffmann, K. Štěpánová, and M. Reinstein, “The effect of motor action and different sensory modalities on terrain classification in a quadruped robot running with multiple gaits,” *Robotics and Autonomous Systems*, vol. 62, no. 12, pp. 1790–1798, 2014.
- [44] X. A. Wu, T. M. Huh, R. Mukherjee, and M. Cutkosky, “Integrated ground reaction force sensing and terrain classification for small legged robots,” *IEEE Robotics and Automation Letters*, vol. 1, no. 2, pp. 1125–1132, 2016.
- [45] 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, pp. 1509–1516, 2019.
- [46] 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.
- [47] J. Guzzi, R. O. Chavez-Garcia, M. Nava, L. M. Gambardella, and A. Giusti, “Path planning with local motion estimations,” *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 2586–2593, 2020.
- [48] R. O. Chavez-Garcia, J. Guzzi, L. M. Gambardella, and A. Giusti, “Learning ground traversability from simulations,” *IEEE Robotics and Automation letters*, vol. 3, no. 3, pp. 1695–1702, 2018.
- [49] G. Kahn, P. Abbeel, and S. Levine, “Badgr: An autonomous self-supervised learning-based navigation system,” *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 1312–1319, 2021.
- [50] M. Nava, J. Guzzi, R. O. Chavez-Garcia, L. M. Gambardella, and A. Giusti, “Learning long-range perception using self-supervision from short-range sensors and odometry,” *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1279–1286, 2019.