

# World Model-based Perception for Visual Legged Locomotion

Hang Lai<sup>1†</sup>, Jiahang Cao<sup>1</sup>, Jiafeng Xu<sup>2</sup>, Hongtao Wu<sup>2</sup>, Yunfeng Lin<sup>1†</sup>, Tao Kong<sup>2</sup>, Yong Yu<sup>1</sup>, Weinan Zhang<sup>1</sup>

**Abstract**—Legged locomotion over various terrains is challenging and requires precise perception of the robot and its surroundings from both proprioception and vision. However, learning directly from high-dimensional visual input is often data-inefficient and intricate. To address this issue, traditional methods attempt to learn a teacher policy with access to privileged information first and then learn a student policy to imitate the teacher’s behavior with visual input. Despite some progress, this imitation framework prevents the student policy from achieving optimal performance due to the information gap between inputs. Furthermore, the learning process is unnatural since animals intuitively learn to traverse different terrains based on their understanding of the world without privileged knowledge. Inspired by this natural ability, we propose a simple yet effective method, World Model-based Perception (WMP), which builds a world model of the environment and learns a policy based on the world model. We illustrate that though completely trained in simulation, the world model can make accurate predictions of real-world trajectories, thus providing informative signals for the policy controller. Extensive simulated and real-world experiments demonstrate that WMP outperforms state-of-the-art baselines in traversability and robustness. We will open-source our code to facilitate reproduction.

## I. INTRODUCTION

Reinforcement Learning (RL) has recently achieved remarkable success in legged locomotion across diverse terrains by training a policy in physical simulation and then transferring it to the real world (i.e., sim-to-real transfer) [1], [2]. Typically, such RL policy takes the proprioception (e.g., positions and velocities of joints) or visual image as input and outputs the desired position or effort control for each actuated joint [3], [4]. Previous literature has shown that a blind policy with only proprioceptive input can traverse terrains like slopes and stairs [5]–[9], but fails in more challenging ones like gaps or pits [10]–[12], where a robot must perceive such terrain in advance; therefore, visual image perception is indispensable [13], [14].

However, directly learning a policy with dense pixel input using reward signals is extremely data-inefficient [15]. Moreover, with a forward-facing camera, a policy needs to remember past perceptions to anticipate the terrain under the robot’s feet [10], which poses an additional challenge for policy learning. To facilitate policy training, the *privileged learning* framework [16] is proposed, which decomposes the training process into two phases. First, a teacher policy is trained with access to low-dimensional privileged information like the scandots around the robot, which is usually

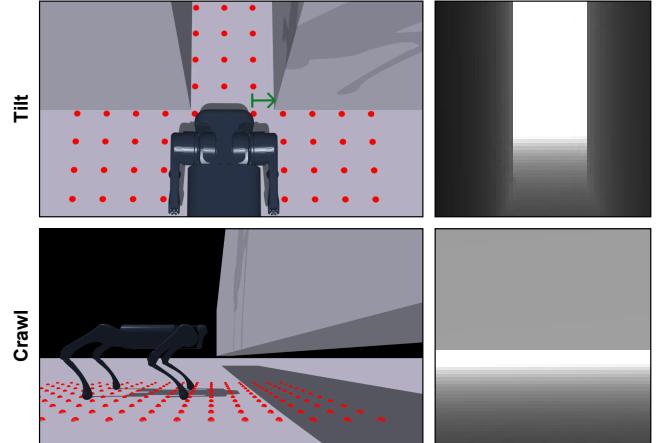


Fig. 1: Scandots (Left) and the corresponding depth images (Right). Top: Sparse scandots<sup>1</sup> can not distinguish the precise distance to the boundaries (indicated in green), leading to a collision with the left barrier. Bottom: Scandots can not represent off-ground obstacles. In contrast, depth images can represent these terrains well.

inaccessible in the real world. Afterward, a student policy is trained to mimic the teacher’s actions based on the seen images via ConvNet-RNN architecture [10]–[12].

Privileged learning has been widely exploited and exhibits traversability over multiple terrains in quadrupedal locomotion [5], [10], [11], [13], but it still has some limitations. Firstly, since the student cannot recover the teacher’s behavior perfectly due to generalization error [17], the performance of student policy tends to lag behind the teacher’s. The performance discrepancy is further magnified when an information gap exists between privileged information and images [18]. Secondly, the teacher policy needs to access different types of extra information, which is labor-intensive to design. For example, Cheng et al. [11] choose the terrain scandots as the privileged information, which can not be generalized to terrains that require precise boundary distinction, like Tilt, or ones with off-ground obstacles, like Crawl, as illustrated in Figure 1. Besides, Zhuang et al. [12] use the geometry of obstacles as privileged information, which is terrain-relevant. Therefore, they train different teachers separately for each terrain. These limitations may hinder its broader applications in more complex scenarios.

In contrast, animals naturally learn to traverse unstructured fields and can make good decisions in unfamiliar situations with limited perception. One hypothesis is that animals, especially humans, build a mental model that holds their

<sup>†</sup>Work done during internship at ByteDance Research.

<sup>1</sup>Dept. of Computer Sci. and Eng., Shanghai Jiao Tong University, China.

<sup>2</sup>ByteDance Research, China.

Code is available at: <https://anonymous.4open.science/r/WMP-ICRA25>

<sup>1</sup>The scandots are no denser than the terrain mesh vertices, which have a horizontal resolution of 10cm in our simulation setting.

understanding of the real world [19], [20]. When performing actions, it helps perceive past information and predict future sensory data [21]–[24]. Inspired by these findings, model-based RL (MBRL) strives to learn a world model from collected data and derives a policy from it [25]. With the help of such models, MBRL has made immense progress in a variety of tasks with limited data, ranging from simulated robot control [26], [27] to playing video games [20], [28], [29]. However, the application of world models in vision-based legged locomotion is still lacking.

This paper investigates whether visual legged locomotion can benefit from world model learning. To this end, we present **World Model-based Perception** (WMP), a novel end-to-end framework combining advanced MBRL with sim-to-real transfer. Specifically, WMP trains a world model in simulations to predict future perceptions using past ones and learns a policy given the abstract description extracted from the world model. Though trained entirely using simulated data, the world model can still predict real-world perception well. By leveraging the learned model, WMP circumvents the limitations of privileged learning and naturally compresses a series of high-dimensional perceptions into a meaningful representation, contributing to decision-making.

We compare WMP to state-of-the-art baselines over multiple terrains, including terrains like Slope and Stair and more difficult ones like Gap and Crawl. In simulation comparison, WMP obtains near-optimal rewards compared with the teacher policy, surpassing the student policy by a pronounced margin. Subsequently, we evaluate WMP and baselines on a real Unitree A1 robot, where WMP successfully traverses the tested terrains with increased difficulty, verifying the advantage of world model learning in robot control. For example, WMP can traverse Gap with 85cm (about 2.1x robot length), Climb with 55cm (about 2.2x robot height), and Crawl with 22cm (about 0.8x robot height), achieving the best traversal performance on the A1 robot. To the best of our knowledge, this is the first work that deals with challenging vision-based legged locomotion via world modeling, which could become a new paradigm for robot control tasks.

## II. RELATED WORK

**RL for Legged Locomotion.** Reinforcement learning (RL) has emerged as a promising method for legged locomotion [1]–[4], [30]. Previous literature has shown that policies with only proprioception as input can go through multiple terrains in the real world by leveraging biologically inspired rewards design [4], [5], [16], domain randomization [3], [7], [31], and curriculum learning [8], [32]. However, without visual perception, it can be extremely difficult for these “blind” robots to tackle more complex terrains [12], [13]. To incorporate visual information, the privileged learning framework is developed [10]–[12], where a teacher policy trained with access to privileged information is used to guide a vision-based student policy. However, the design of privileged information and the performance gap between teacher and student remain critical limitations of these methods [6], [33]. Besides, one concurrent work proposes to estimate the

scandots using past observations [34], which also suffers from the limitation of scandots inevitably. In contrast, Our method presents a more general and effective framework to incorporate visual images without these limitations.

**Model-based RL.** Model-based reinforcement learning (MBRL) approaches have shown promise for learning complex robot control policies by learning a dynamics model of the environment to help decision-making [25], [27], [35], [36]. In MBRL, much effort has been devoted to learning an accurate model in partially observable and pixel-input environments [20], [26], [28], [37]. As a prominent example, Dreamer-V3 [29] achieves impressive performance across diverse domains by learning a world model in a compact latent space, namely the Recurrent State-Space Model (RSSM). Inspired by the success of Dreamer, RSSM has also been widely exploited in robot control tasks, ranging from robotic manipulation [38]–[40] to blind quadrupedal locomotion [41]. Our work also builds upon the RSSM world model architecture, which can seamlessly take advantage of innovations in MBRL literature and push the boundaries of its application in legged locomotion with visual input.

## III. PRELIMINARIES

We formulate legged locomotion as a Partially Observable Markov Decision Process (POMDP) defined by the tuple  $(\mathcal{S}, \mathcal{O}, \mathcal{A}, T, r, \gamma)$ , where  $\mathcal{S}$ ,  $\mathcal{O}$ , and  $\mathcal{A}$  are the state, observation, and action spaces, respectively.  $T(s_{t+1} | s_t, a_t)$  is the transition density of state  $s_t$  given action  $a_t$ , and the reward function is denoted as  $r(s_t, a_t)$ .  $\gamma \in (0, 1)$  is a discount factor. At each timestep  $t$ , only the partial observation  $o_t \in \mathcal{O}$  can be observed instead of  $s_t$  due to the limitation of sensors. The goal of reinforcement learning (RL) is to find the optimal policy  $\pi^*: \mathcal{O} \rightarrow \mathcal{A}$  that maximizes the expected return (sum of discounted rewards):

$$\pi^* := \arg \max_{\pi} \mathbb{E}_{s_{t+1} \sim T(\cdot | s_t, a_t), a_t \sim \pi(\cdot | o_t)} \left[ \sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \right]. \quad (1)$$

Specifically, in the vision-based legged locomotion,  $o_t$  consists of the proprioception  $o_t^p$  and the depth image  $d_t$ . In addition to  $o_t$ , the underlying state  $s_t$  also contains the privileged information  $s_t^{\text{pri}}$ , which can only be accessed in the simulator:

$$o_t := (o_t^p, d_t), \quad (2)$$

$$s_t := (o_t, s_t^{\text{pri}}). \quad (3)$$

## IV. METHOD

This section introduces World Model-based Perception (WMP), an end-to-end framework that utilizes world models to extract information from high-dimensional sensor input. Unlike the two-stage training process used in privileged learning, WMP only adopts one stage to learn the world model and policy simultaneously.

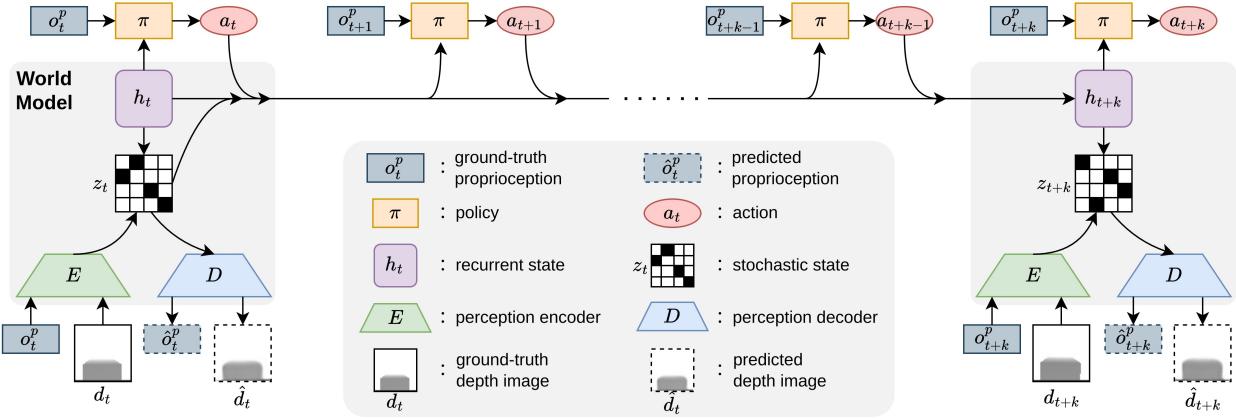


Fig. 2: Illustration of the WMP framework. The world model runs at a lower frequency than policy, with an update interval of  $k$  timesteps.

### A. World Model Learning

Following previous works [29], [37], we adopt a Recurrent State-Space Model (RSSM) variant as our world model architecture. Considering the computational cost of acquiring depth images in the simulator and the time cost of running RSSM on board, we modify the original RSSM by running the world model with a lower frequency than the policy. As Figure 2 shows, the world model updates the recurrent state  $h_t$  every  $k$  timesteps. Formally, the RSSM in our method consists of four components parameterized by  $\phi$ :

$$\begin{aligned} \text{Recurrent model: } & h_t = f_\phi(h_{t-k}, z_{t-k}, a_{t-k:t-1}) \\ \text{Encoder: } & z_t \sim q_\phi(\cdot | h_t, o_t) \\ \text{Dynamic predictor: } & \hat{z}_t \sim p_\phi(\cdot | h_t) \\ \text{Decoder: } & \hat{o}_t \sim p_\phi(\cdot | h_t, z_t). \end{aligned} \quad (4)$$

To be more specific, the recurrent model  $f_\phi$  operates in the low-dimensional latent space  $h$  and predicts the deterministic recurrent state  $h_t$  based on the previous  $h_{t-k}$ , sequence of action  $a_{t-k:t-1}$ , and the previous stochastic state  $z_{t-k}$ . From the recurrent state  $h_t$ , the RSSM computes two distributions over stochastic states  $z_t$ . The posterior state  $z_t$  incorporates information from observation  $o_t$  through the encoder  $q_\phi$ . The prior state  $\hat{z}_t$  aims to predict the posterior without access to  $o_t$ , enabling the model to anticipate future dynamics without ground-truth observation. The decoder generates the estimated observation  $\hat{o}_t$ , making it possible to reconstruct the high-dimensional observation. The recurrent model is implemented using the GRU (Gated Recurrent Unit) [42] network, and the encoder and decoder utilize the convolutional neural network (CNN) structure for depth image  $d_t$  and multi-layer perceptions (MLP) for proprioception observations  $o_t^p$ .

Similar to Hafner et al. [29], we optimize these four ingredients jointly by minimizing the loss over trajectories of length  $L$ :

$$\begin{aligned} \mathcal{L}(\phi) \doteq \mathbb{E}_{q_\phi} \left[ \sum_{t=k}^L -\ln p_\phi(o_t | z_t, h_t) \right. \\ \left. + \beta \text{KL}[q_\phi(\cdot | h_t, o_t) || p_\phi(\cdot | h_t)] \right], \end{aligned} \quad (5)$$

where  $n$  is a non-negative integer, and  $\beta$  is a hyperparameter. The first term in Eq. (5) is the reconstruction loss, which encourages the posterior  $z_t$  to contain sufficient information about  $o_t$ , while the second KL term regularizes the prior and posterior to approximate each other, allowing open-loop prediction of future observations based on current  $h_t$  and future actions. Please refer to the original Dreamer literature [26], [28], [29] for more details about RSSM training.

### B. Policy Learning

Policy learning for vision-based locomotion is non-trivial due to the partial observability, as described in Section III. However, the recurrent state  $h_t$  in a well-trained world model encapsulates sufficient information for future prediction, akin to the underlying Markovian state  $s_t$ . Building on this insight, we train a policy that incorporates  $h_t$  as input:

$$a_{t+i} \sim \pi_\theta(\cdot | o_{t+i}^p, \text{sg}(h_t)), \forall i \in [0, k-1], \quad (6)$$

where  $\text{sg}(\cdot)$  represents the stop-gradient operator. We employ the asymmetric actor-critic framework [33], where the critic can access the privileged information  $s_t^{\text{pri}}$ :

$$v(s_{t+i}) \sim V_\theta(\cdot | o_{t+i}^p, \text{sg}(h_t), s_{t+i}^{\text{pri}}), \forall i \in [0, k-1]. \quad (7)$$

We find that the recurrent state  $h_t$  plays an essential role in critic learning since the scandots in  $s_t^{\text{pri}}$  can not represent some types of terrains like Tilt and Crawl, as discussed in Section I.

The actor and critic are trained using the data collected in the simulator via the PPO (Proximal Policy Optimization) algorithm [43]. Note that we do not utilize the world model to generate rollout data for policy training as in previous MBRL methods [27], [41]. Because the world model is trained using simulated data, which means it can not generate more accurate data than the simulator, and sampling data in the simulator is efficient enough. Training world models with real-world data is a possible way to make models more accurate, which we leave as future work.

### C. Training Details

**Environment.** We implement our method and baselines based upon the legged\_gym codebase [32], which leverages the Isaac Gym simulator [44] to support simulation

TABLE I: Comparison results on different terrains in simulation. Results are averaged over 100 trajectories with different difficulties. Bold numbers indicate the best scores among algorithms excluding Teacher. N/A means that the method is not applicable to the terrain.

		Slope (0-36°)	Stair (5-21cm)	Gap (0-90cm)	Climb (0-54cm)	Tilt (32-28cm)	Crawl (35-21cm)
Return $\uparrow$	<b>WMP (ours)</b>	<b>36.55 ± 0.82</b>	<b>35.06 ± 3.54</b>	<b>32.37 ± 8.24</b>	34.64 ± 2.99	<b>34.73 ± 3.13</b>	<b>36.60 ± 0.33</b>
	<b>Teacher</b>	36.70 ± 1.79	35.07 ± 0.68	32.80 ± 7.06	35.32 ± 3.33	N/A	N/A
	<b>Student</b>	36.31 ± 1.35	34.93 ± 1.32	27.07 ± 12.38	33.42 ± 5.66	N/A	N/A
	<b>Blind</b>	33.96 ± 2.58	23.56 ± 12.98	9.80 ± 9.51	14.11 ± 13.27	3.94 ± 2.28	8.92 ± 7.65
Tracking Error $\downarrow$	<b>WMP w/o Prop</b>	36.07 ± 0.27	34.62 ± 3.51	30.73 ± 9.93	<b>34.67 ± 1.95</b>	30.78 ± 7.22	36.41 ± 1.76
	<b>WMP (ours)</b>	<b>0.008 ± 0.006</b>	<b>0.013 ± 0.010</b>	<b>0.26 ± 0.13</b>	<b>0.13 ± 0.08</b>	<b>0.02 ± 0.01</b>	<b>0.006 ± 0.002</b>
	<b>Teacher</b>	0.007 ± 0.010	0.012 ± 0.005	0.23 ± 0.21	0.09 ± 0.07	N/A	N/A
	<b>Student</b>	<b>0.008 ± 0.004</b>	0.015 ± 0.006	0.35 ± 0.28	0.16 ± 0.07	N/A	N/A
	<b>Blind</b>	0.019 ± 0.011	0.033 ± 0.021	0.33 ± 0.19	0.26 ± 0.07	0.08 ± 0.06	0.051 ± 0.022
	<b>WMP w/o Prop</b>	0.009 ± 0.003	0.017 ± 0.027	0.28 ± 0.18	0.14 ± 0.09	0.08 ± 0.10	0.013 ± 0.006

of thousands of robots in parallel. Specifically, We create 4096 Unitree A1 [45] instances on six types of terrains, including Slope, Stair, Gap, Climb, Crawl, and Tilt, each with varying difficulty levels, as listed in Table I. We adopt the same terrain curriculum as in Rudin et al. [32]. All robots are initialized to different terrains with the lowest difficulty in a certain proportion. The robot is moved to a higher level of difficulty once it passes the borders of its terrain or assigned to a lower level if it moves by less than half of the distance required by its target velocity. Robots take actions at a frequency of 50 Hz, i.e., 0.02s per timestep. Depth images are computed every  $k$  timesteps and sent to the policy with 100ms latency to facilitate sim-to-real transfer. We also randomize the physical parameters to further improve the policy’s robustness as in Cheng et al. [11].

**State and Action Space.** Precisely, the proprioception observation  $o_t^p \in \mathbb{R}^{45}$  consists of base angular velocities, gravity projection, commands, positions and velocities of joints, and last action  $a_{t-1}$ . The privileged information  $s_t^{\text{pri}}$  contains scandots, foot contact forces, and randomized physical parameters.  $d_t \in \mathbb{R}^{64 \times 64}$  is the egocentric depth image with  $58^\circ \times 58^\circ$  field of view. The action  $a_t \in \mathbb{R}^{12}$  specifies the joints’ target positions:  $q_d = q_{\text{stand}} + a_t$ , where  $q_{\text{stand}}$  is the default joint positions when standing. The torques  $\tau$  are calculated through a PD controller:

$$\tau = K_p (q_d - q) + K_d (\dot{q}_d - \dot{q}), \quad (8)$$

where  $q$  and  $\dot{q}$  are the joints’ current positions and velocities, respectively. The target joint velocities  $\dot{q}_d$  are set to 0 and  $(K_p, K_d)$  are the parameters of the PD controller.

**Reward Function.** The robot is trained to track a 3-dim command:  $(v_x^{\text{cmd}}, v_y^{\text{cmd}}, \omega_z^{\text{cmd}})$ . To achieve this, we adopt a suite of reward functions similar to Cheng et al. [11]. The main difference is that they manually select waypoints along a preset trajectory and compute the velocity-tracking reward based on the direction to the next waypoint. On the contrary, we use a simpler form of velocity-tracking reward to reduce human efforts in waypoint selection:

$$r_{\text{tracking}} = \exp\left(\left(\min(v_{xy}, v_{xy}^{\text{cmd}} + 0.1) - v_{xy}^{\text{cmd}}\right)^2 / \sigma\right), \quad (9)$$

where the clipping operation encourages the robot to follow the command most of the time, but it can also reach higher

speeds when necessary, e.g., jumping over gaps. We add additional penalties to avoid getting stuck or turning around the obstacles. Besides, we employ an AMP (Adversarial Motion Priors) [46] style reward to make the robot converge to a more natural behavior [8]:

$$r_{\text{style}}(s, s') = \max[0, 1 - 0.25(D_\psi(s, s') - 1)^2], \quad (10)$$

where  $D_\psi$  is the discriminator trained to distinguish whether a state transition is from a reference dataset  $\mathcal{D}_{\text{ref}}$  or produced by the agent:

$$\begin{aligned} \arg \min_{\psi} & \mathbb{E}_{(s, s') \sim \mathcal{D}_{\text{ref}}} [(D_\psi(s, s') - 1)^2] \\ & + \mathbb{E}_{(s, s') \sim \pi_\theta(s, a)} [(D_\psi(s, s') + 1)^2] \\ & + \frac{w_{\text{gp}}}{2} \mathbb{E}_{(s, s') \sim \mathcal{D}_{\text{ref}}} [\|\nabla_\psi D_\psi(s, s')\|^2]. \end{aligned} \quad (11)$$

## V. EXPERIMENTAL RESULTS

Our experiments aim to answer the following questions:

- How does WMP perform compared with previous state-of-the-art methods in vision-based locomotion?
- Could a world model trained in a simulator predict real-world trajectory well?
- Could the achievement of WMP in simulation be well transferred to real robots?

To ensure a fair comparison, all the methods are trained using the same environment and reward functions described in Section IV-C.

### A. Simulation Comparison

To answer these questions, we first evaluate our method and baselines in terms of RL return and velocity tracking error over different terrains in simulation, where the velocity tracking error is defined as the mean square error between  $v_{xy}^{\text{cmd}}$  and  $v_{xy}$ . The baselines we compare to include:

- **Teacher.** The teacher policy is trained with access to privileged information like scandots, serving as an oracle baseline.
- **Student.** We reproduce the student policy according to Cheng et al. [11], which utilizes a ConvNet-RNN to mimic the teacher’s policy using depth images.
- **Blind.** We ablate the depth image in the world model, resulting in a blind policy that gives actions purely based on proprioception.

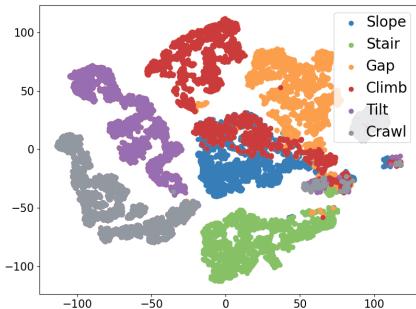


Fig. 3: T-SNE result of recurrent state  $h_t$  over six different terrains.

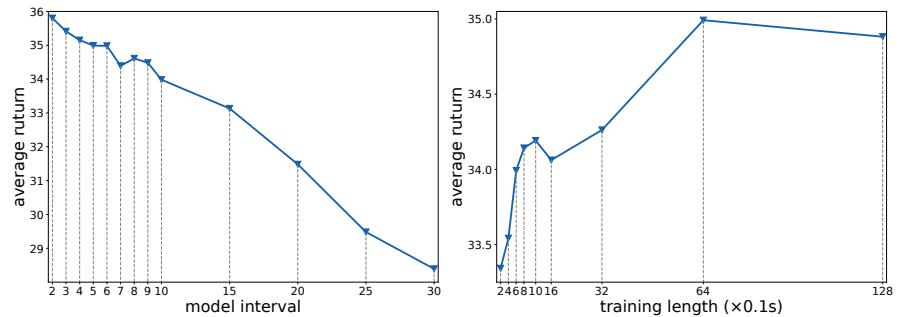


Fig. 4: Average return over different world model interval  $k$  (Left) and different lengths of training data (Right).

- **WMP w/o Prop.** Similar to Blind, we remove the proprioception in the world model.

The comparison results are shown in Table I. Note that the Teacher and Student baselines do not apply to Tilt and Crawl due to the limitation of scandots, as discussed in Section I. From the result, our method WMP achieves higher return and smaller velocity tracking error than baselines in most tasks. The performance gap between WMP and Teacher is much smaller than between Teacher and Student, revealing the superiority of WMP by leveraging the world model to extract proper information from past perceptions. Moreover, removing the depth image in the world model causes a severe performance dropping except for Slope, underscoring the importance of visual information for locomotion over challenging terrains [10], [12]. Besides, ablating the proprioception also decreases the performance slightly, which we attribute to the fact that predicting the proprioception can help capture the physical properties of the environment.

### B. Empirical Study

This section provides empirical studies to understand the benefits of our method from the world model.

**Recurrent State Visualization.** We first collect the recurrent state  $h_t$  over six terrains and visualize them in Figure 3 to investigate whether  $h_t$  contains enough information for versatile locomotion. As the t-SNE figure shows, the  $h_t$  of different terrains holds clear boundaries, with only a slight overlap between Slope and Climb, since these two terrains have similar depth images and Climb can be considered as a 90° Slope. From the visualization result,  $h_t$  represents the terrains well and can help the policy take action according to the specific task.

**Model Interval.** The model interval parameter  $k$  affects both world model training and real-world deployment. To investigate its influence, we vary  $k$  from 2 to 30. The results are shown in Figure 4. In general, a world model with a smaller interval obtains higher rewards in simulation because it enables the robot to respond to changes in its surroundings more quickly. However, unlike the ideal situation in the simulator, real-world applications have non-negligible latency for depth image acquiring and world model computing, taking around 40ms in total on A1 hardware. Therefore, we choose  $k = 5$ , i.e., world model intervals of 0.1s, a trade-off between ideal performance and computational cost.

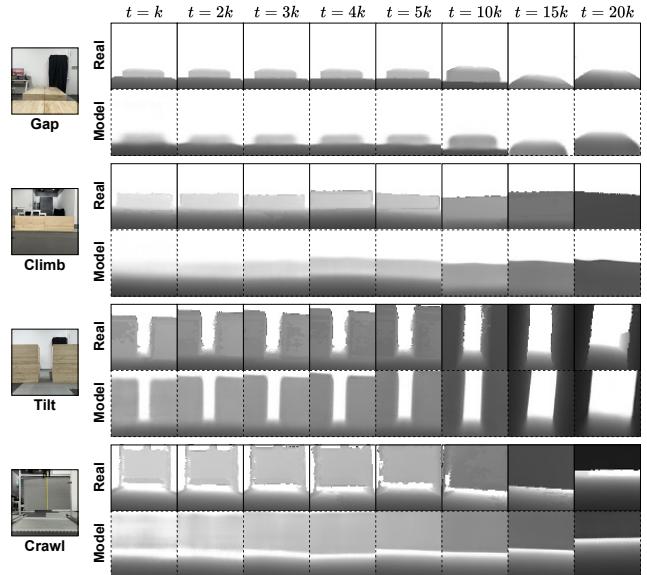


Fig. 5: Real-world depth images and long-term predictions of depth images using the world model.

**Training Length.** During world model training, we randomly sample trajectory segments with fixed length  $L$  and train the model to predict current perception based on previous ones in the segment. The length of the training data determines how long the historical information model can remember. In Figure 4, we conduct experiments with different training lengths. According to the result, training world models with 1-second segments is sufficient to achieve acceptable performance. This is consistent with our intuition: what the robot saw one second before is roughly under its feet. Further extending the horizon can help perceive the environment dynamics, but a segment that is too long may make it inefficient to back-propagate the gradient through RSSM. For this reason, we set the training length to 6.4 seconds in other experiments throughout the paper.

**Real World Prediction.** While the world model performs well in simulation, whether the excellent performance can transfer to the real world remains to be justified. To verify this, we collect trajectories in the real world and use the model to predict the future, given the initial observation and action sequence without access to intermediate depth images. Results are shown in Figure 5. From the result, a world model trained purely in the simulator can give accurate predictions

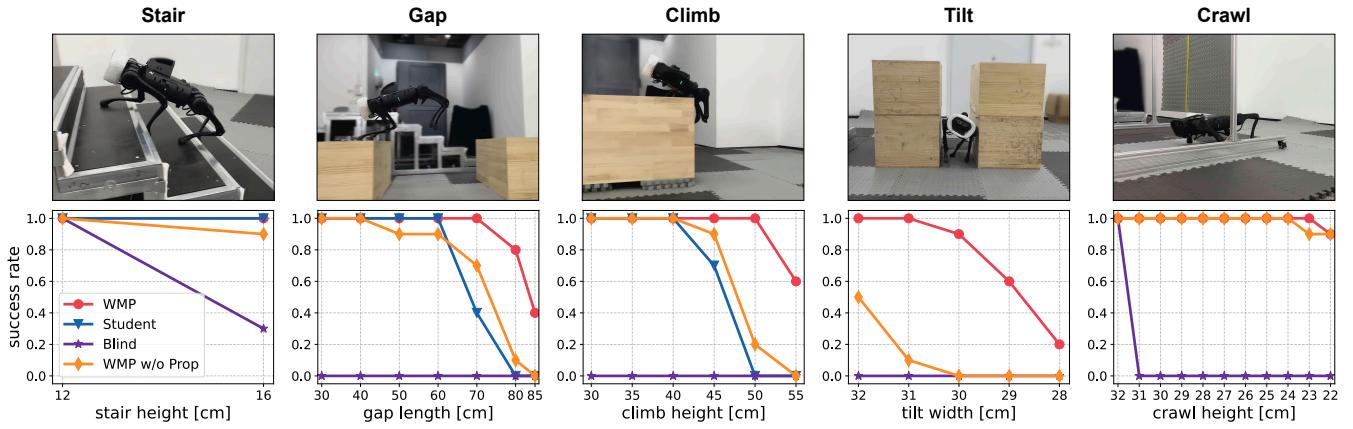


Fig. 6: Real-world evaluation over multiple terrains with different difficulties. Success rates are calculated over ten trials.

for real-world trajectories, especially in the critical place it will pass. For example, in the Crawl task, the shape of the obstacle in the predicted depth image is different from the real ones since robots have never seen this shape of obstacles in the simulator. Nevertheless, the position and angle of the narrow crevice it can traverse are highly consistent in real and model images, highlighting the strength and generalization of latent space world modeling. This finding may help explain why our method exhibits smooth sim-to-real transfer.

### C. Real-world Evaluation

Subsequently, we apply WMP and other baselines to a real-world A1 robot. All methods are directly run on the onboard Jetson NX hardware without external computing devices. Depth images are read from the front Intel D435i camera at 60 Hz with a resolution of  $424 \times 240$ . We preprocess the noisy depth images with spatial and temporal filters to narrow the visual sim-to-real gap [12]. The processed images are then cropped and down-sampled to  $64 \times 64$  and sent to the world model with 100ms latency. We set  $K_p = 40$ ,  $K_d = 1.0$  to make it consistent with the simulation setting. We select five terrains with different difficulties for a comprehensive evaluation, including Stair, Gap, Climb, Tilt, and Crawl, but exclude Slope, which is too easy to distinguish between methods.

The success rates are listed in Figure 6. From the comparison, the Student policy can traverse through the first three terrains with low difficulty but fails in more difficult cases, reflecting the performance gap between the Student policy and the optimal Teacher policy. In contrast, our method exhibits a more stable control behavior and successfully traverses more challenging terrains, including Tilt and Crawl, which the Student policy cannot tackle. To name a few, WMP can traverse Gap of 85cm (about 2.1x robot length), Climb of 55cm (about 2.2x robot height), and Crawl of 22cm (about 0.8x robot height), close to the hardest level in the simulator. This means that our method achieves smaller sim-to-real gaps through world modeling. Besides, ablating proprioception or images in the world model degrades performance to different degrees, demonstrating the advantage of physical and visual world modeling for locomotion. Please refer to the supplemental video for detailed comparisons.



Fig. 7: Snapshots of outdoor experiments.

We also deploy our policy to outdoor environments in a park. Some snapshots are shown in Figure 7. Our policy shows consistent behavior in outdoor and indoor environments and successfully goes up and down stairs, climbs platforms up to 45cm, and traverses grass and gravel, which further validates the generalization of our method.

## VI. CONCLUSION

In this paper, we present World Model-based Perception (WMP), a simple yet effective framework that combines MBRL with vision-based legged locomotion, drawing inspiration from the role of the mental model in animal cognition and decision-making. By leveraging the advanced world model, WMP outperforms previous state-of-the-art baselines in both simulation and real-world evaluation, achieving the best traversal performance on Unitree A1 robots. Further empirical analyses reveal that the main superiority of WMP lies in utilizing the world model to extract useful information from historical high-dimensional perceptions. We hope our method could provide insight into the emergence of a better natural learning paradigm for robots. For future work, it is tempting to train the world model with a mixture of simulated and real-world data, which holds the promise to construct a more realistic world model. Besides, it is also appealing to incorporate other forms of perception, like the sense of touch, into the world model to expand its applications.

## REFERENCES

- [1] J. Tan, T. Zhang, E. Coumans, A. Iscen, Y. Bai, D. Hafner, S. Bohez, and V. Vanhoucke, "Sim-to-real: Learning agile locomotion for quadruped robots," *arXiv preprint arXiv:1804.10332*, 2018.
- [2] W. Yu, V. C. Kumar, G. Turk, and C. K. Liu, "Sim-to-real transfer for biped locomotion," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 3503–3510.
- [3] X. B. Peng, E. Coumans, T. Zhang, T.-W. Lee, J. Tan, and S. Levine, "Learning agile robotic locomotion skills by imitating animals," *arXiv preprint arXiv:2004.00784*, 2020.
- [4] 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.
- [5] A. Kumar, Z. Fu, D. Pathak, and J. Malik, "Rma: Rapid motor adaptation for legged robots," *arXiv preprint arXiv:2107.04034*, 2021.
- [6] H. Lai, W. Zhang, X. He, C. Yu, Z. Tian, Y. Yu, and J. Wang, "Sim-to-real transfer for quadrupedal locomotion via terrain transformer," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. London, United Kingdom: IEEE, May 2023, p. 5141–5147. [Online]. Available: <https://ieeexplore.ieee.org/document/10160497/>
- [7] I. M. A. Nahrendra, B. Yu, and H. Myung, "Dreamwaq: Learning robust quadrupedal locomotion with implicit terrain imagination via deep reinforcement learning," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 5078–5084.
- [8] J. Wu, G. Xin, C. Qi, and Y. Xue, "Learning robust and agile legged locomotion using adversarial motion priors," *IEEE Robotics and Automation Letters*, 2023.
- [9] J. Long, Z. Wang, Q. Li, L. Cao, J. Gao, and J. Pang, "Hybrid internal model: Learning agile legged locomotion with simulated robot response," in *The Twelfth International Conference on Learning Representations*, 2024.
- [10] A. Agarwal, A. Kumar, J. Malik, and D. Pathak, "Legged locomotion in challenging terrains using egocentric vision," in *Conference on robot learning*. PMLR, 2023, pp. 403–415.
- [11] X. Cheng, K. Shi, A. Agarwal, and D. Pathak, "Extreme parkour with legged robots," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 11443–11450.
- [12] Z. Zhuang, Z. Fu, J. Wang, C. Atkeson, S. Schwertfeger, C. Finn, and H. Zhao, "Robot parkour learning," *arXiv preprint arXiv:2309.05665*, 2023.
- [13] 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.
- [14] H. Duan, B. Pandit, M. S. Gadde, B. Van Marum, J. Dao, C. Kim, and A. Fern, "Learning vision-based bipedal locomotion for challenging terrain," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 56–62.
- [15] A. Anand, E. Racah, S. Ozair, Y. Bengio, M.-A. Côté, and R. D. Hjelm, "Unsupervised state representation learning in atari," *Advances in neural information processing systems*, vol. 32, 2019.
- [16] 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.
- [17] M. Mohri, "Foundations of machine learning," 2018.
- [18] H. He, C. Bai, H. Lai, L. Wang, and W. Zhang, "Privileged knowledge distillation for sim-to-real policy generalization," *arXiv preprint arXiv:2305.18464*, 2023.
- [19] J. W. Forrester, "Counterintuitive behavior of social systems," *Theory and decision*, vol. 2, no. 2, pp. 109–140, 1971.
- [20] D. Ha and J. Schmidhuber, "World models," *arXiv preprint arXiv:1803.10122*, 2018.
- [21] G. B. Keller, T. Bonhoeffer, and M. Hübener, "Sensorimotor mismatch signals in primary visual cortex of the behaving mouse," *Neuron*, vol. 74, no. 5, pp. 809–815, 2012.
- [22] G. W. Maus, J. Fischer, and D. Whitney, "Motion-dependent representation of space in area mt+," *Neuron*, vol. 78, no. 3, pp. 554–562, 2013.
- [23] N. Nortmann, S. Rekauzke, S. Onat, P. König, and D. Jancke, "Primary visual cortex represents the difference between past and present," *Cerebral Cortex*, vol. 25, no. 6, pp. 1427–1440, 2015.
- [24] M. Leinweber, D. R. Ward, J. M. Sobczak, A. Attinger, and G. B. Keller, "A sensorimotor circuit in mouse cortex for visual flow predictions," *Neuron*, vol. 95, no. 6, pp. 1420–1432, 2017.
- [25] F.-M. Luo, T. Xu, H. Lai, X.-H. Chen, W. Zhang, and Y. Yu, "A survey on model-based reinforcement learning," *Science China Information Sciences*, vol. 67, no. 2, p. 121101, 2024.
- [26] D. Hafner, T. Lillicrap, J. Ba, and M. Norouzi, "Dream to control: Learning behaviors by latent imagination," *arXiv preprint arXiv:1912.01603*, 2019.
- [27] M. Janner, J. Fu, M. Zhang, and S. Levine, "When to trust your model: Model-based policy optimization," *Advances in neural information processing systems*, vol. 32, 2019.
- [28] D. Hafner, T. Lillicrap, M. Norouzi, and J. Ba, "Mastering atari with discrete world models," *arXiv preprint arXiv:2010.02193*, 2020.
- [29] D. Hafner, J. Pasukonis, J. Ba, and T. Lillicrap, "Mastering diverse domains through world models," *arXiv preprint arXiv:2301.04104*, 2023.
- [30] Z. Fu, A. Kumar, J. Malik, and D. Pathak, "Minimizing energy consumption leads to the emergence of gaits in legged robots," *arXiv preprint arXiv:2111.01674*, 2021.
- [31] Z. Xie, X. Da, M. van de Panne, B. Babich, and A. Garg, "Dynamics randomization revisited: A case study for quadrupedal locomotion," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 4955–4961.
- [32] 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.
- [33] L. Pinto, M. Andrychowicz, P. Welinder, W. Zaremba, and P. Abbeel, "Asymmetric actor critic for image-based robot learning," *arXiv preprint arXiv:1710.06542*, 2017.
- [34] S. Luo, S. Li, R. Yu, Z. Wang, J. Wu, and Q. Zhu, "Pie: Parkour with implicit-explicit learning framework for legged robots," *arXiv preprint arXiv:2408.13740*, 2024.
- [35] K. Chua, R. Calandra, R. McAllister, and S. Levine, "Deep reinforcement learning in a handful of trials using probabilistic dynamics models," in *Advances in Neural Information Processing Systems*, 2018, pp. 4754–4765.
- [36] Y. Luo, H. Xu, Y. Li, Y. Tian, T. Darrell, and T. Ma, "Algorithmic framework for model-based deep reinforcement learning with theoretical guarantees," *arXiv preprint arXiv:1807.03858*, 2018.
- [37] D. Hafner, T. Lillicrap, I. Fischer, R. Villegas, D. Ha, H. Lee, and J. Davidson, "Learning latent dynamics for planning from pixels," in *International conference on machine learning*. PMLR, 2019, pp. 2555–2565.
- [38] J. Yamada, M. Rigter, J. Collins, and I. Posner, "Twist: Teacher-student world model distillation for efficient sim-to-real transfer," *arXiv preprint arXiv:2311.03622*, 2023.
- [39] S. Ferraro, P. Mazzaglia, T. Verbelen, and B. Dhoedt, "FOCUS: Object-Centric World Models for Robotics Manipulation," *arXiv preprint arXiv:2307.02427*, 2023.
- [40] R. Mendonca, S. Bahl, and D. Pathak, "Structured world models from human videos," *arXiv preprint arXiv:2308.10901*, no. arXiv:2308.10901, Aug. 2023, arXiv:2308.10901 [cs]. [Online]. Available: <http://arxiv.org/abs/2308.10901>
- [41] P. Wu, A. Escontrela, D. Hafner, P. Abbeel, and K. Goldberg, "Day-dreamer: World models for physical robot learning," in *Conference on robot learning*. PMLR, 2023, pp. 2226–2240.
- [42] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:1412.3555*, 2014.
- [43] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.
- [44] V. Makoviychuk, L. Wawrzyniak, Y. Guo, M. Lu, K. Storey, M. Macklin, D. Hoeller, N. Rudin, A. Allshire, A. Handa *et al.*, "Isaac gym: High performance gpu-based physics simulation for robot learning," *arXiv preprint arXiv:2108.10470*, 2021.
- [45] Unitree, "Unitree robotics," <https://www.unitree.com/>, 2022.
- [46] A. Escontrela, X. B. Peng, W. Yu, T. Zhang, A. Iscen, K. Goldberg, and P. Abbeel, "Adversarial motion priors make good substitutes for complex reward functions," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 25–32.