# Learning Robotic Policy with Imagined Transition: Mitigating the Trade-off between Robustness and Optimality

Wei Xiao[1], Shangke Lyu[2†], Zhefei Gong[1], Renjie Wang[1], Donglin Wang[1]

***Abstract*— Existing quadrupedal locomotion learning paradigms usually rely on extensive domain randomization to alleviate the sim2real gap and enhance robustness. It trains policies by introducing various disturbances into simulated environments to improve their adaptability under uncertainty. However, since optimal performance without disturbances often conflicts with the need to handle worst-case scenarios, this paradigm has a trade-off between optimality and robustness. This trade-off forces the learned policy to prioritize stability in diverse and challenging environments over efficiency and accuracy in no-disturbed ones, leading to overly conservative behaviors that sacrifice peak performance. Inspired by disturbance rejection control, we propose a two-stage framework that integrates policy learning with motion reference to mitigate this phenomenon. This framework enhances the conventional reinforcement learning (RL) approach by incorporating imagined transitions as reference inputs. The imagined transitions are derived from an optimal policy and a dynamics model operating within an idealized setting without disturbances. By providing this reference, we found that the policy could generate actions that are better aligned with the desired motion, which led to accelerated training and reduced tracking errors under external disturbances.**

## I. INTRODUCTION

Learning-based locomotion control methods for quadruped robots have achieved remarkable results in recent years [1], [2], [3], [4], [5], [6]. To adapt to various wild environments, these methods collected large amounts of data in massive parallel simulators with extensive dynamics variations via domain randomization [7]. This data-driven paradigm allows RL-based robots to adapt to realistic terrain and unseen disturbances.

While domain randomization is a widely used technique to improve the robustness of robot policies, previous studies [8], [9] have shown that it often sacrifices optimality, resulting in conservative behaviors. For instance, when trained with large payload variations, the learned policy tends to maintain a lower base height rather than achieve the target height specified in the reward function. This example reveals that the robustness achieved through domain randomization essentially comes at the cost of compromising certain task objectives under unknown changes. Consequently, it becomes difficult to consistently meet high task standards, which limits the applicability of domain randomization in complex tasks, particularly those requiring high accuracy.

To mitigate this problem, we require a method that can maintain task standards without compromising adaptability under disturbances. A key element for enabling robots to recover from unknown perturbations is the explicit identification of the goal that the policy should adapt to. This aligns with the control-theoretic perspective, particularly classical disturbance rejection control, where reference signals are used to compute counteractions that cancel perturbation effects. These reference signals define the desired system behavior and represent the performance the controller seeks to preserve despite disturbances. In RL policy training, however, such references are only implicitly encoded in the reward functions, and no explicit goal-oriented guidance is provided. Inspired by disturbance rejection control, we therefore propose to supply explicit motion references in RL policy learning. These references not only provide a clearer objective for training, enabling the policy to learn to resist disturbances and track desired motions, but also deliver richer guidance for policy inference.

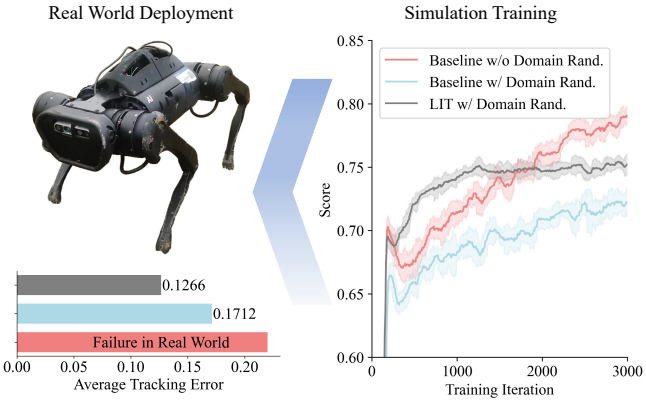


Fig. 1. **Proposed LIT has lower tracking error than baseline.** For current RL-based locomotion paradigms (Baseline), domain randomization trades optimality for robustness.

Normally, it is straightforward to train a policy in simulation under a fixed set of environmental parameters without any randomization. Although such a policy cannot be directly deployed in reality due to limited robustness, it can still serve as a reference motion to guide both policy training and inference. Alternatively speaking, a well-trained policy in an ideal simulation can be leveraged to generate goal information that assists the deployed policy in making better decisions. To this end, we propose **LIT** (**L**earning with **I**magined **T**ransition), a two-stage framework training paradigm that enables the learning of robust locomotion policies from ideal motion. By introducing "ideal" actions and transitions, the policy could generate actions that are better aligned with nominal dynamics under disturbances (as illustrated by the gray line

†Corresponding author.
[1]Westlake University, [2]Nanjing University.

in Fig. 1), thereby enabling more robust and accurate task execution.

Specifically, we reconstruct a learnable dynamics model to approximate transition propagation, obtained from the unperturbed simulation environment together with an ideal policy. These provide reference actions and next-step ideal observations for policy training and inference, guiding the policy toward robust locomotion even under external disturbances. To achieve this, we incorporate ideal transitions into the policy input while continuing to optimize the policy in an RL manner. This approach not only improves training efficiency but also enhances the generalization capability of the locomotion policy. However, the predicted observation from the dynamics model and ideal policy may become ineffective in Out-of-Distribution (OOD) scenarios. If we include it in the optimization objective, it may lead to a training crash or non-convergence. To address this, we designed an adjustment mechanism of the dynamics model to mitigate the negative impact of OOD observation during policy learning.

In summary, the contributions of our work are as follows: **(i)** We propose **LIT**, a two-stage RL-based locomotion framework that learns robust policies with imagined transitions generated from an ideal policy and a dynamics model trained in unperturbed simulation; **(ii)** We design an uncertainty-based adjustment mechanism to mitigate the adverse effect of dynamics model errors under out-of-distribution observations; **(iii)** We conduct simulation, ablation, and real-world experiments showing that LIT improves training efficiency, velocity tracking, and robustness under unknown disturbances.

## II. Related Work

### A. RL-based Legged Locomotion

Reinforcement learning has substantially advanced quadrupedal locomotion, supported by high-fidelity and GPU-accelerated simulators such as MuJoCo [10] and Isaac Gym [11], [12]. To address partial observability and environmental uncertainty, teacher-student frameworks [3], [13] and their extensions [14], [15], [16] have improved adaptation to unknown terrains and dynamics. Other works enhance generalization through diverse training environments [5], learned environmental representations [17], [4], [6], or the integration of model-based and adaptive control ideas [18], [19], [20], [21]. Domain randomization remains a common strategy for sim-to-real transfer [1], [2], [3], [4], [6], where policies are trained over broad parameter and noise ranges. However, this robustness often comes at the cost of optimality, leading to conservative behaviors under nominal or less perturbed conditions.

### B. Model-based RL

Model-based reinforcement learning learns dynamics models to predict future states and improve planning or data efficiency. Representative methods include the Dreamer series [22], [23], [24], which performs planning in latent space, and TDMPC [25], [26], which combines learned dynamics with model predictive control. In legged locomotion, DayDreamer [27] and PIP-loco [28] further demonstrate the utility of learned models for planning, control, or representation learning. In contrast, our method does not use the dynamics model for planning or latent representation. Instead, it uses the model to generate reference next observations, together with an ideal policy, to directly guide robust policy learning.

## III. Preliminary

*a) POMDP:* We model locomotion as an infinite-horizon POMDP $\mathscr{M} = (\mathscr{S}, \mathscr{O}, \mathscr{A}, d_0, p, r, \gamma)$, where the robot starts from $d_0$, receives partial observations $o \in \mathscr{O}$, takes continuous actions $a \in \mathscr{A}$, evolves under transition $p$, and optimizes discounted reward $r$ with factor $\gamma$. To handle partial observability, the policy uses an observation history $\mathbf{o}_{t-H:t} = [\mathbf{o}_{t-H}, \ldots, \mathbf{o}_{t-1}, \mathbf{o}_t]^T$.

*b) State Space:* The actor observation $\mathbf{o}_t$ includes command velocity $\mathbf{c}_t = [v_x^c, v_y^c, \omega_{\text{yaw}}^c]$, IMU measurements including base angular velocity and gravity direction, joint positions and velocities, and the previous action. During training, the critic additionally receives privileged information, including base velocity, external force, and terrain height.

*c) Action Space:* The action specifies scaled offsets from nominal joint positions: $\theta_{\text{target}} = \theta_0 + k\mathbf{a}_t$, where $k \leq 1$. For Unitree A1, the action dimension is 12.

## IV. Method

### A. Framework Overview

Our framework is shown in Fig. 2. It is a general approach that can be incorporated into most of the previous RL-based Locomotion methods such as RMA [3], Dreamwaq [4], HIMLoco [6], etc., and here we use HIMLoco as the backbone algorithm. It consists of two parts: Motion Reference Learning and Policy Learning with Imagined Transition. Section IV-B discusses how to perform Motion Reference Learning. It aims to obtain the optimal gait from a fixed simulation environment and compress the optimal gait into an ideal policy and a dynamics model. Section IV-C presents how to use motion reference to guide policy learning.

### B. Motion Reference Learning in Fixed Dynamics

*1) Ideal Policy:* In the process of reinforcement learning to train robotic policies, we found that wider ranges of dynamics randomization lead to lower returns after convergence. To obtain an optimal reference, we train a policy that is optimal in the unperturbed simulation environment in a fixed dynamic (i.e., no dynamic randomization setting) environment, which is called the ideal policy $\pi_{ideal}(\mathbf{a}_t^r | \mathbf{o}_{t-H:t})$ in the paper.

*2) Dynamics Model:* To obtain reference observations, we train a dynamics model under the unperturbed simulation environment. The output of the dynamics model includes $\mu$ and $\sigma$ in eq. 1, which means the mean and standard deviation of the next observation estimation. We use the dynamics model to output $\mu$ and $\sigma$, which parameterize a

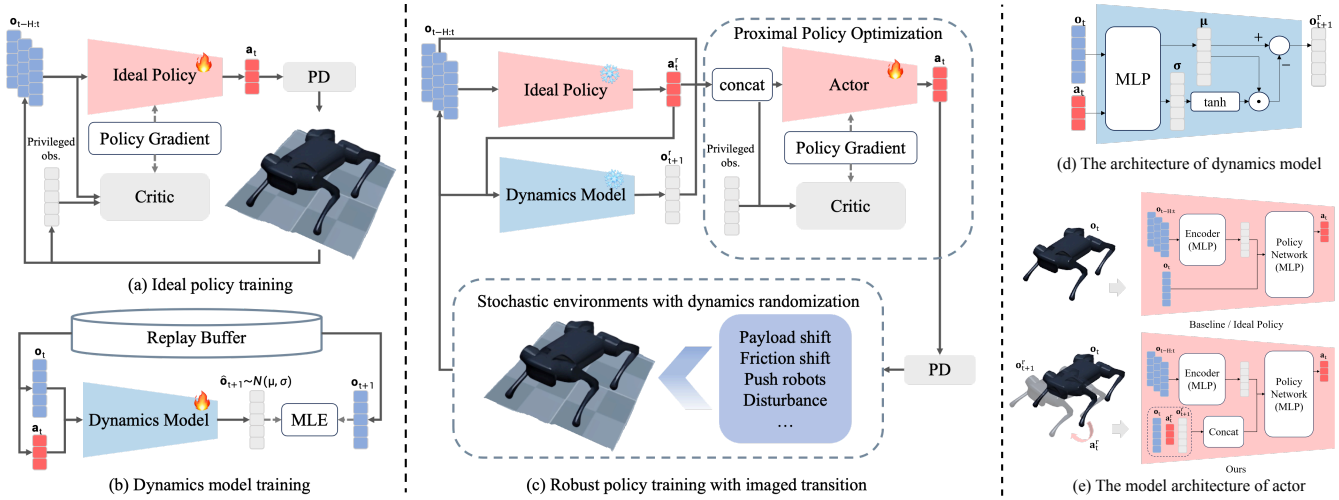


Fig. 2. **Framework Overview of LIT.** The left part is motion reference learning in fixed dynamics. The middle part is policy learning with imagined transition. $\mathbf{a}_t^r$ and $\mathbf{o}_{t+1}^r$ mean reference action and imagined next observation respectively. $[\mathbf{o}_t, \mathbf{a}_t^r, \mathbf{o}_{t+1}^r]$ constitutes the imagined transition. The right part shows the architectures of the actor and dynamics model.

normal distribution. The training objective is to maximize the likelihood of the true next observation $\mathbf{o}_{t+1}$ under this distribution.

$$(\mu_t, \sigma_t) = d_\theta(\mathbf{o}_t, \mathbf{a}_t), p_\theta(\mathbf{o}_{t+1} \mid \mathbf{o}_t, \mathbf{a}_t) = \mathcal{N}(\mu_t, \mathrm{diag}(\sigma_t^2)). \tag{1}$$

$$\mathscr{L}_{\mathrm{MLE}} = -\mathbb{E}_{(\mathbf{o}_t, \mathbf{a}_t, \mathbf{o}_{t+1}) \sim \mathscr{D}} \left[ \log p_\theta(\mathbf{o}_{t+1} \mid \mathbf{o}_t, \mathbf{a}_t) \right]. \tag{2}$$

To prevent training and deployment collisions caused by model inaccuracies, we developed an adjustment mechanism to adjust the prediction of the next observation in the following robust policy learning stage (see Sec. IV-C) and deployment. According to the general law of supervised learning, the larger the standard deviation $\sigma$ the less reliable the prediction $\mu$ of the dynamics model. Thus, when the variance is large, we should minimize the effect of the observation reference on the subsequent network. Inspired by [4], [29], we multiply $\mu$ element-wise by the uncertainty weight $1 - \tanh(\sigma) \in [0, 1]$, as shown in Eq. 3. Experiments in Fig. 5 showed that this design significantly improved the robot's performance in the unseen region.

$$\mathbf{o}_{t+1}^r = (1 - \tanh(\sigma)) \odot \mu \tag{3}$$

### C. Policy Learning with Imagined Transition

In short, the only difference in our policy network is that we are adding the input of action and observation reference, which represents motion reference. The combination of $\mathbf{o}_t$, $\mathbf{a}_t^r$, and $\mathbf{o}_{t+1}^r$ forms a complete state transition $\{\mathbf{o}_t, \mathbf{a}_t^r, \mathbf{o}_{t+1}^r\}$, which is our imagined transition. We concatenate reference action $\mathbf{a}_t^r$ and imagined next observation $\mathbf{o}_{t+1}^r$ from the motion reference with proprioceptive observations $\mathbf{o}_t$, whereas the baseline, depicted on the top, follows the encoder-policy network architecture used in[4], [6].

$$\mathbf{a}_t \sim \pi(\cdot | \mathbf{o}_{t-H:t}) \tag{4}$$

$$\mathbf{a}_t \sim \pi(\cdot | \mathbf{o}_{t-H:t}, \mathbf{a}_t^r, \mathbf{o}_{t+1}^r) \tag{5}$$

We argue that while dynamics randomization greatly extends the solution space for robotic policies, it does not change the desired behavior that the robot should execute. For instance, when a high-level command specifies moving forward by 1.0 m, the policy is expected to execute exactly that distance across all randomized environments, rather than drifting to 0.9 m or 1.1 m. The imagined transition $\{\mathbf{o}_t, \mathbf{a}_t^r, \mathbf{o}_{t+1}^r\}$ represents the desired behavior established in Sec. IV-B. By leveraging it, policy optimization becomes more efficient than searching for the optimal solution directly in the huge solution space.

## V. Experiments

### A. Experimental Setup

**(i)** Compared Methods: We compare six variants: (A) Baseline, equivalent to [6]; (B) Ours w/o Adjust, using the raw mean output of the dynamics model as the observation reference; (C) Ours w Residual, adding the action reference to the actor output [30], [31]; (D) Ours w/o Action Reference; (E) Ours w/o State Reference; and (F) Ours. **(ii)** Simulation Setup: We train policies with PPO in Isaac Gym [11], using 4096 parallel environments, rollout length 100, and 2000 iterations on an NVIDIA A800 GPU. Longer training is used only for the learning-curve comparison in Sec. V-B.2.

### B. Evaluation on Simulator

*1) Command Tracking:* We evaluate linear and angular velocity tracking errors under different terrains, computed as $\|v_{x,y} - v_{x,y}^{\mathrm{target}}\|^2$ and $\|\omega_{\mathrm{yaw}} - \omega_{\mathrm{yaw}}^{\mathrm{target}}\|^2$. Table I shows that our method achieves lower tracking errors in most settings.

*2) Learning Curve:* We record normalized tracking scores and terrain levels during training, following the curriculum setting of [6]. The normalized angular velocity tracking score and the normalized linear velocity tracking score are

Fig. 3. **Deployment on Real World.** Five scenarios in order from left to right: (1) Plane - flat surface walking, (2) Payload - carrying 3 Kg additional weight, (3) Disturbance-1 - external force applied from the back, (4) Disturbance-2 - external disturbance applied to the right-back leg, and (5) Lawn - walking on uneven grassy terrain.

TABLE I

AVERAGE TRACKING ERROR IN SIMULATOR OVER 1000 TRIALS.

| Terrain Types | Velocity Types | Baseline | Ours |
|---|---|---|---|
| Smooth Slopes | Linear | 0.170 | **0.120** |
| | Angular | 0.101 | **0.099** |
| Rough Slopes | Linear | 0.387 | **0.162** |
| | Angular | 0.164 | **0.130** |
| Stairs | Linear | 2.371 | **0.992** |
| | Angular | **0.515** | 0.552 |
| Discrete | Linear | 1.955 | **0.120** |
| | Angular | 0.220 | **0.099** |

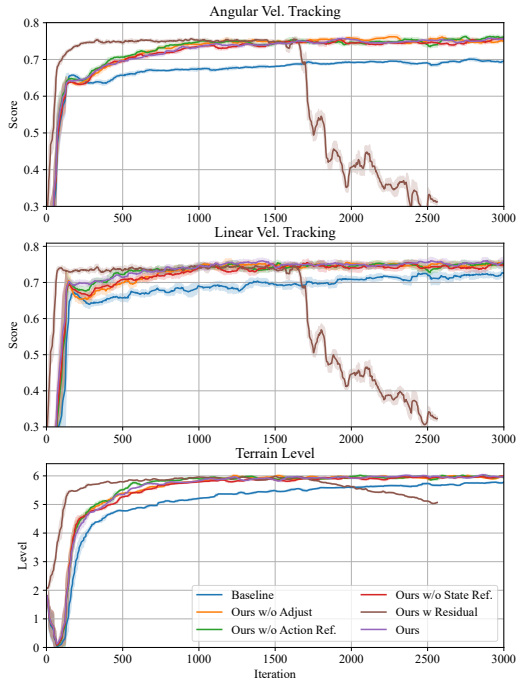


Fig. 4. **Ablation studies with learning curves.** From top to bottom: (1) normalized angular velocity tracking score, (2) normalized linear velocity tracking score, and (3) maximum reachable terrain level in training.

calculated by $\exp(-\frac{\|\omega_{\text{yaw}}-\omega_{\text{yaw}}^{\text{target}}\|^2}{0.25})$ and $\exp(-\frac{\|v_{x,y}-v_{x,y}^{\text{target}}\|^2}{0.25})$ respectively.

As shown in Fig. 4, our method consistently improves tracking and terrain traversal over the baseline. The residual-action variant accelerates early learning but later becomes unstable, suggesting that direct residual use of imagined transitions is vulnerable to OOD reference errors. The proposed adjustment mechanism mitigates this issue.

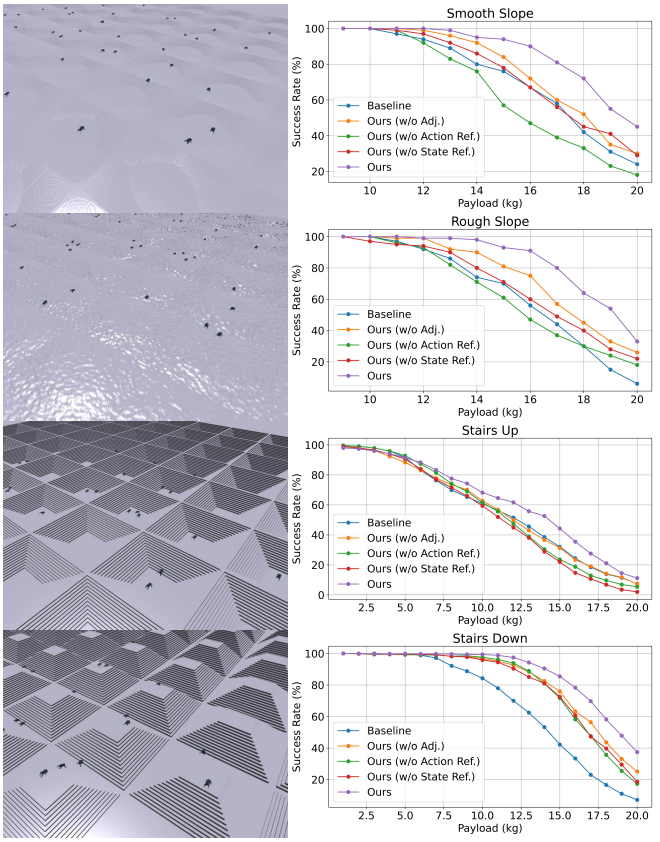


Fig. 5. **Success rate of different algorithms under various payloads.** The success rate for payloads from 1kg to 9kg on both smooth and rough slopes is 100%.

*3) Payload Shift:* We evaluate robustness to unseen payloads on four terrains: smooth slopes, rough slopes, stairs up, and stairs down. The A1 robot is commanded to walk forward at $[1,0,0]$ m/s for 200 steps while the payload increases from 1 kg to 20 kg. As shown in Fig. 5, our method achieves higher survival rates under heavy payloads. This indicates that the adjustment mechanism and reference inputs mainly improve robustness in OOD conditions, even when their in-distribution learning curves are similar.

*4) External push disturbance:* Pushing a robot with external force is a widely used method for evaluating the robustness of its locomotion. In this experiment, the push force was applied by adding impulse-like velocity perturbation to the quadruped's base along the *xy*-plane of the

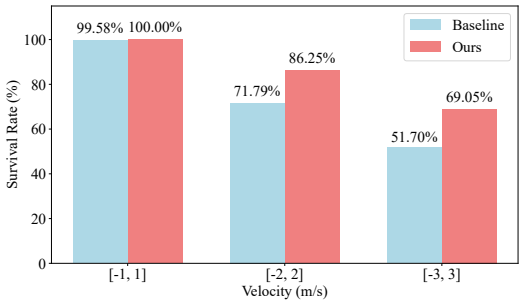


Fig. 6. **Comparison of survival rate under external pushing.** The experimental results are categorized into three classes based on the magnitude of the external push velocity. A higher survival rate indicates better robustness.

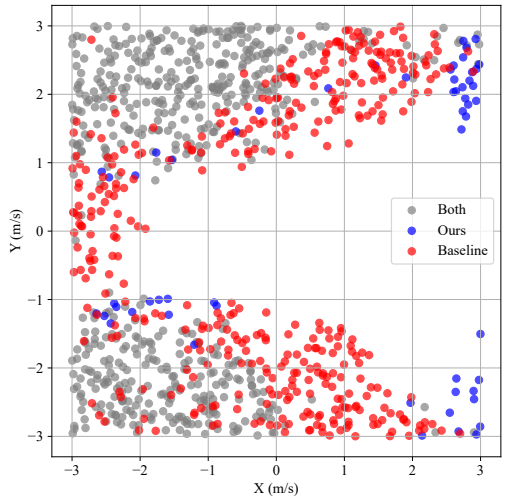


Fig. 7. **Fall resistant visualization with external push.** 'Both' represents cases where both ours and baseline fall. 'Ours' denotes cases where only our method falls, while 'Baseline' denotes cases where only the baseline falls.

world frame. The robot was instructed to walk forward with a command of $[1,0,0]$ m/s, while simultaneously responding to the unpredicted random push. The magnitude of the added velocity determines the intensity of the push, with larger magnitudes corresponding to more forceful pushes. To assess the robustness of our approach in comparison to the baseline, pushes were randomly sampled 2000 times within a velocity range of $[-3,3]$ m/s. The results in Figure 6 illustrate that our method outperforms the baseline in terms of fall resistance when subjected to external pushes.

Figure 7 provides a detailed visualization of the standing versus falling status under various push magnitudes and directions for different algorithms. Given the command $[1,0,0]$ m/s, it can be observed that Figure 7 exhibits near-symmetry along the $x$-axis, but not along the $y$-axis. All methods include push perturbations within the range $[-1,1]$ m/s during the robust training phase, and consequently, they perform well within this range. However, as the perturbation range increases, the advantage of our approach becomes increasingly pronounced. Our method outperforms the baseline by surviving more than 300 additional times in 2000 randomized tests.

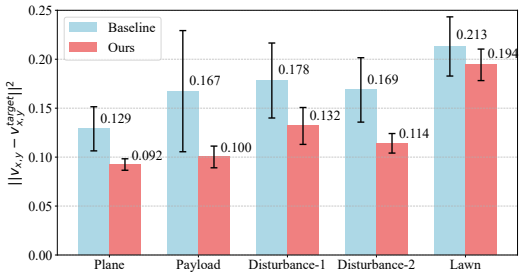


Fig. 8. **Comparison of command tracking performance in real-world scenarios.** The results are averaged over ten trials. Error bars represent standard error. A lower tracking error indicates better deployed performance.

### C. Evaluation on Real World

To test the effectiveness of policy deployment in the real world, we designed a variety of challenging scenarios as shown in Figure 3. We deployed on the Unitree A1 robot the policy trained for 2000 iterations in simulation, with the PD controller's parameters set to $Kp = 40.0$ and $Kd = 1.0$. The robot was instructed to walk forward with a command of $[1,0,0]$ m/s in five scenarios. We measured the tracking errors $||v_{x,y} - v_{x,y}^{target}||^2$ as the performance metric to evaluate the robot's ability to track the linear velocity command, where the body velocity $v_{x,y}$ is estimated by onboard state estimator. The results in Figure 8 show that our method has a lower tracking error in five scenarios compared to the baseline.

### D. Implementation Details

**(i)** Reward Design: The reward follows [4] and is summarized in Table II. Here $h^{\text{target}}$ is the desired base height, while $p_z^{\text{target}}$, $p_z^i$, and $v_{xy}^i$ denote desired foot height, actual foot height, and planar foot velocity in the robot frame; **(ii)** Dynamics Randomization: We randomize physical parameters, actuation, delay, external force, and initial joint states during robust policy learning, ranges are listed in Table III; **(iii)** Training Curriculum: We use terrain and command curricula following [12], [32], [6]. The terrain map contains 200 terrains in a $20 \times 10$ grid with increasing difficulty. Difficulty increases when the robot reaches 80% linear tracking reward and decreases when it fails to traverse half of the terrain. Commands are sampled every 25 steps, with wider velocity ranges for slopes and rough terrains than for stairs and discrete obstacles.

TABLE II
REWARD FUNCTIONS.

| Reward | Equation ($r_i$) | Weight ($w_i$) |
|---|---|---|
| Linear velocity tracking | $\exp\left\{-\frac{\|\mathbf{v}_{xy}^{\text{cmd}} - \mathbf{v}_{xy}\|_2^2}{0.25}\right\}$ | 1.0 |
| Angular velocity tracking | $\exp\left\{-\frac{\left(\omega_{\text{yaw}}^{\text{cmd}} - \omega_{\text{yaw}}\right)^2}{0.25}\right\}$ | 0.5 |
| Lin. velocity ($z$) | $v_z^2$ | -2.0 |
| Ang. velocity ($xy$) | $\|\boldsymbol{\omega}_{xy}\|_2^2$ | -0.05 |
| Orientation | $\|\mathbf{g}\|_2^2$ | -0.2 |
| Joint accelerations | $\|\ddot{\boldsymbol{\theta}}\|^2$ | $-2.5 \times 10^{-7}$ |
| Joint power | $\|\boldsymbol{\tau}^\top \dot{\boldsymbol{\theta}}\|$ | $-2 \times 10^{-5}$ |
| Body height | $(h^{\text{target}} - h)^2$ | -1.0 |
| Foot clearance | $\Sigma_{i=0}^{3}\left(p_z^{\text{target}} - p_z^i\right)^2 \cdot v_{xy}^i$ | -0.01 |
| Action rate | $\|\mathbf{a}_t - \mathbf{a}_{t-1}\|_2^2$ | -0.01 |
| Smoothness | $\|\mathbf{a}_t - 2\mathbf{a}_{t-1} + \mathbf{a}_{t-2}\|_2^2$ | -0.01 |

TABLE III

DOMAIN RANDOMIZATION.

| Parameters | Range | Unit |
|---|---|---|
| CoM | $[-0.05, 0.05]$ | m |
| Payload Mass Offset | $[-1, 2]$ | Kg |
| Ground Friction | $[0.2, 1.25]$ | - |
| Motor Strength | $[0.9, 1.1]\times$ motor torque | Nm |
| Joint $K_p$ | $[0.9, 1.1]\times$ 40.0 | Nm/rad |
| Joint $K_d$ | $[0.9, 1.1]\times$ 1.0 | Nms/rad |
| Initial Joint Positions | $[0.5, 1.5]\times$ nominal value | rad |
| System Delay | $[0, 15]$ | ms |
| External Force | $[-30, 30]$ | N |

## VI. CONCLUSION

We propose LIT, a two-stage framework that mitigates the robustness–optimality trade-off in RL-based locomotion by conditioning robust policy learning on imagined transitions. Simulation and real-world experiments show improved velocity tracking, training efficiency, and robustness under disturbances. Future work will explore broader robotic tasks where precision and robustness conflict.

## ACKNOWLEDGMENT

This work was supported by the National Natural Science Foundation of China (Grant No. 62573362) and the National Science and Technology Innovation 2030-Major Projects (Grant No. 2022ZD0208800).

## REFERENCES

[1] Z. Xie, X. Da, M. van de Panne, B. Babich, and A. Garg, "Dynamics randomization revisited: A case study for quadrupedal locomotion," in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 4955–4961.

[2] 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.

[3] A. Kumar, Z. Fu, D. Pathak, and J. Malik, "Rma: Rapid motor adaptation for legged robots," in *Robotics: Science and Systems*, 2021.

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

[5] G. B. Margolis and P. Agrawal, "Walk these ways: Tuning robot control for generalization with multiplicity of behavior," in *Conference on Robot Learning (CoRL)*, 2023.

[6] 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.

[7] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, "Domain randomization for transferring deep neural networks from simulation to the real world," in *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 2017, pp. 23–30.

[8] B. Mehta, M. Diaz, F. Golemo, C. J. Pal, and L. Paull, "Active domain randomization," in *Proceedings of the Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, L. P. Kaelbling, D. Kragic, and K. Sugiura, Eds., vol. 100. PMLR, 30 Oct–01 Nov 2020, pp. 1162–1176. [Online]. Available: https://proceedings.mlr.press/v100/mehta20a.html

[9] G. Tiboni, P. Klink, J. Peters, T. Tommasi, C. D'Eramo, and G. Chalvatzaki, "Domain randomization via entropy maximization," 2023.

[10] E. Todorov, T. Erez, and Y. Tassa, "Mujoco: A physics engine for model-based control," in *2012 IEEE/RSJ international conference on intelligent robots and systems*. IEEE, 2012, pp. 5026–5033.

[11] 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," *Advances in neural information processing systems*, 2021.

[12] 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 (CoRL)*, 2022.

[13] 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. [Online]. Available: https://www.science.org/doi/abs/10.1126/scirobotics.abc5986

[14] A. Agarwal, A. Kumar, J. Malik, and D. Pathak, "Legged locomotion in challenging terrains using egocentric vision," in *Proceedings of The 6th Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, K. Liu, D. Kulic, and J. Ichnowski, Eds., vol. 205. PMLR, 14–18 Dec 2023, pp. 403–415. [Online]. Available: https://proceedings.mlr.press/v205/agarwal23a.html

[15] X. Cheng, K. Shi, A. Agarwal, and D. Pathak, "Extreme parkour with legged robots," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 2024, pp. 11 443–11 450.

[16] A. Kumar, Z. Li, J. Zeng, D. Pathak, K. Sreenath, and J. Malik, "Adapting rapid motor adaptation for bipedal robots," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 1161–1168.

[17] G. Ji, J. Mun, H. Kim, and J. Hwangbo, "Concurrent training of a control policy and a state estimator for dynamic and robust legged locomotion," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 4630–4637, 2022.

[18] S. Lyu, H. Zhao, and D. Wang, "A composite control strategy for quadruped robot by integrating reinforcement learning and model-based control," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2023, pp. 751–758.

[19] S. Lyu, X. Lang, H. Zhao, H. Zhang, P. Ding, and D. Wang, "Rl2ac: Reinforcement learning-based rapid online adaptive control for legged robot robust locomotion," in *Proceedings of the Robotics: Science and Systems*, 2024.

[20] D. Kim, J. Di Carlo, B. Katz, G. Bledt, and S. Kim, "Highly dynamic quadruped locomotion via whole-body impulse control and model predictive control," *arXiv preprint arXiv:1909.06586*, 2019.

[21] P. A. Ioannou and J. Sun, *Robust adaptive control*. PTR Prentice-Hall Upper Saddle River, NJ, 1996, vol. 1.

[22] D. Hafner, T. Lillicrap, J. Ba, and M. Norouzi, "Dream to control: Learning behaviors by latent imagination," *arXiv preprint arXiv:1912.01603*, 2019.

[23] D. Hafner, T. Lillicrap, M. Norouzi, and J. Ba, "Mastering atari with discrete world models," *arXiv preprint arXiv:2010.02193*, 2020.

[24] D. Hafner, J. Pasukonis, J. Ba, and T. Lillicrap, "Mastering diverse domains through world models," *arXiv preprint arXiv:2301.04104*, 2023.

[25] N. Hansen, X. Wang, and H. Su, "Temporal difference learning for model predictive control," *arXiv preprint arXiv:2203.04955*, 2022.

[26] N. Hansen, H. Su, and X. Wang, "Td-mpc2: Scalable, robust world models for continuous control," *arXiv preprint arXiv:2310.16828*, 2023.

[27] P. Wu, A. Escontrela, D. Hafner, P. Abbeel, and K. Goldberg, "Daydreamer: World models for physical robot learning," in *Conference on Robot Learning (CoRL)*, 2023.

[28] A. Shirwatkar, N. Saxena, K. Chandra, and S. Kolathaya, "Pip-loco: A proprioceptive infinite horizon planning framework for quadrupedal robot locomotion," *arXiv preprint arXiv:2409.09441*, 2024.

[29] I. Nahrendra, B. Yu, M. Oh, D. Lee, S. Lee, H. Lee, H. Lim, and H. Myung, "Obstacle-aware quadrupedal locomotion with resilient multi-modal reinforcement learning," *arXiv preprint arXiv:2409.19709*, 2024.

[30] T. Silver, K. Allen, J. Tenenbaum, and L. Kaelbling, "Residual policy learning," *arXiv preprint arXiv:1812.06298*, 2018.

[31] X. Yuan, T. Mu, S. Tao, Y. Fang, M. Zhang, and H. Su, "Policy decorator: Model-agnostic online refinement for large policy model," *arXiv preprint arXiv:2412.13630*, 2024.

[32] J. Wu, G. Xin, C. Qi, and Y. Xue, "Learning robust and agile legged locomotion using adversarial motion priors," *IEEE Robotics and Automation Letters*, 2022.