Toward Reliable Sim-to-Real Predictability for MoE-based Robust Quadrupedal Locomotion

Author Names Omitted for Anonymous Review. Paper-ID 1098

Page: https://robogauge.github.io/ Code: Train, Evaluate, Deploy

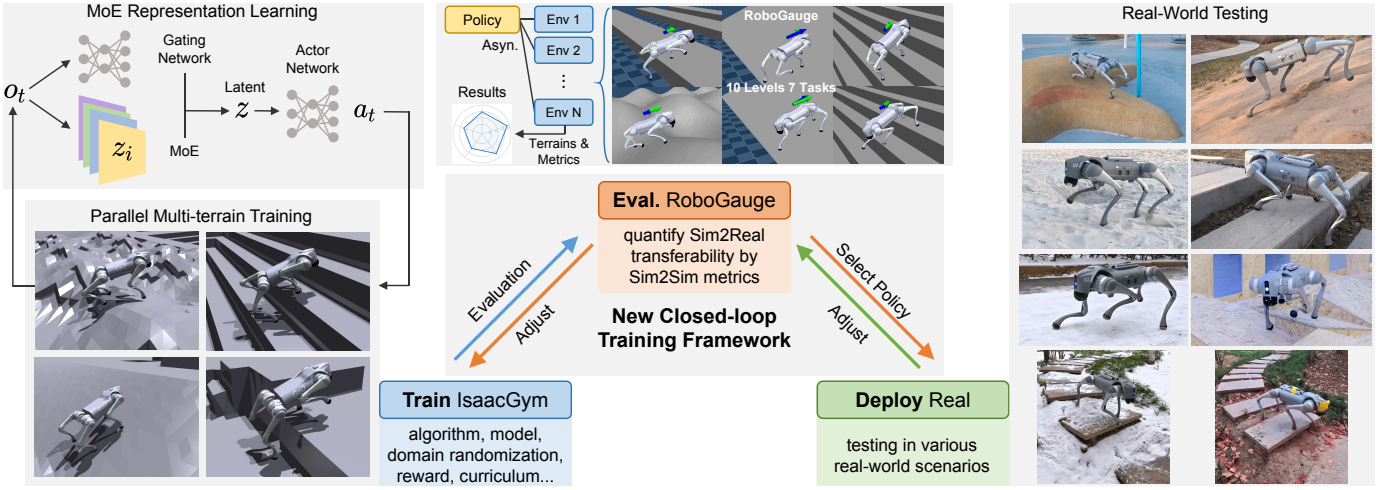


Fig. 1: Our proposed framework integrates a Mixture-of-Experts architecture for terrain and command representation with the RoboGauge assessment suite to quantify sim-to-real transferability through sim-to-sim metrics. This closed-loop design enables reliable policy selection to facilitate robust deployment for agile locomotion across diverse challenging environments based solely on proprioception.

***Abstract*—Reinforcement learning has shown strong promise for quadrupedal agile locomotion, even with proprioception-only sensing. In practice, however, sim-to-real gap and reward overfitting in complex terrains can produce policies that fail to transfer, while physical validation remains risky and inefficient. To address these challenges, we introduce a unified framework encompassing a Mixture-of-Experts (MoE) locomotion policy for robust multi-terrain representation with RoboGauge, a predictive assessment suite that quantifies sim-to-real transferability. The MoE policy employs a gated set of specialist experts to decompose latent terrain and command modeling, achieving superior deployment robustness and generalization via proprioception alone. RoboGauge further provides multi-dimensional proprioception-based metrics via sim-to-sim tests over terrains, difficulty levels, and domain randomizations, enabling reliable MoE policy selection without extensive physical trials. Experiments on a Unitree Go2 demonstrate robust locomotion on unseen challenging terrains, including snow, sand, stairs, slopes, and 30 cm obstacles. In dedicated high-speed tests, the robot reaches 4 m/s and exhibits an emergent narrow-width gait associated with improved stability at high velocity.**

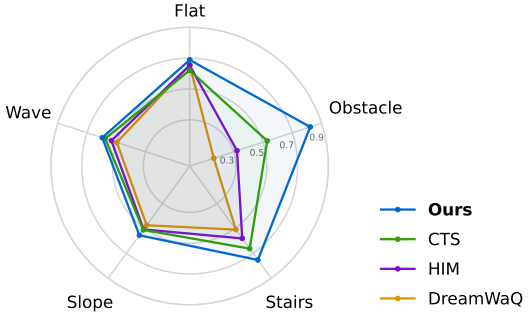


Fig. 2: Comparative analysis against one-stage proprioceptive methods including CTS, HIM, and DreamWaQ. Within the RoboGauge framework, each axis reflects average performance on a specific terrain and serves as a reliable proxy to quantify sim-to-real capability. Our architecture consistently outperforms or matches previous state-of-the-art across all evaluated terrains under RoboGauge's metrics.

I. Introduction

Robots frequently operate in complex and dynamic environments which require high levels of mobility [19, 18, 14]. Quadrupedal robots have garnered significant prominence due to their superior mobility and environmental adaptability [3, 10, 45, 6, 9, 5, 12, 20, 27, 32, 54, 55]. Reinforcement learning has emerged as a potent methodology for motion control by facilitating continuous policy optimization through simulation-based interactions to enhance the robustness of robotic locomotion [24, 23, 46, 16, 51, 45, 52, 41, 33, 36, 28, 31, 8, 30, 53].

The inherent sim-to-real gap remains a primary barrier as

simulation-based performance metrics often prove unreliable for real-world deployment [22, 47, 38, 1, 4]. Specifically, high training rewards across diverse terrains often fail to guarantee physical stability, as policies tend to overfit to the specific dynamics of the simulated robot, thereby degrading generalization to real-world hardware [24, 23, 21]. Moreover, the lack of reliable quantitative proxies compels researchers to rely on direct physical validation, a process that remains prohibitively risky and inefficient [10, 45, 52].

To mitigate these challenges, we propose a training framework that integrates a Mixture-of-Experts (MoE) architecture for terrain and command representation with the RoboGauge assessment suite. This MoE approach improves modeling capabilities by relying exclusively on proprioception to encode unknown terrains and commands while avoiding exteroceptive sensors like cameras, LiDAR, or foot contact sensors, which frequently fail in extreme conditions such as dense smoke and insufficient lighting or violent shaking. Complementing the policy iteration architecture we develop RoboGauge as a predictive evaluation framework designed to quantify sim-to-real stability by utilizing a parallelized sim-to-sim methodology across 6 distinct metrics involving 7 terrains and 10 difficulty levels as well as 3 objectives and 4 domain randomizations.

Fig. 2 illustrates the performance distribution of various models across seven terrains evaluated within the RoboGauge. Our MoE policy outperforms all baseline methods across every terrain category to demonstrate comprehensive superiority. This approach further exhibits exceptional performance during actual deployment on physical robots.

Our contributions are summarized as follows:

- We propose RoboGauge, a comprehensive predictive assessment framework that utilizes a sim-to-sim methodology to quantify sim-to-real transferability, thereby mitigating the risk of hardware damage during direct physical deployment.
- We integrate a Mixture-of-Experts module into the policy to resolve existing deficiencies in multi-terrain representation and demonstrate superior mobility on the physical Unitree Go2 robot.
- We demonstrate that our framework enables the robot to reach a high-speed locomotion of 4 m/s on flat terrain while exhibiting an emergent narrow-width gait associated with improved stability.

II. Related Work

A. Reinforcement Learning for Quadrupedal Locomotion

Reinforcement learning for quadrupedal locomotion in physical environments is hindered by severe sample inefficiency and potential hardware hazards [10, 45, 52]. The predominant sim-to-real approach employs frameworks such as proximal policy optimization [42] or teacher-student training to achieve multi-terrain traversal at velocities under 1 m/s [41, 24, 56]. Adaptability has further advanced through latent parameter estimation via adaptation modules or recurrent belief encoders and contrastive learning within parallelized simulations [23, 11, 33, 36, 28]. Furthermore research pushes agility to peak velocities of 3.9 m/s through command curricula [31, 34] whereas diverse gaits [30, 35, 2] and seamless switching emerge from energy optimization rewards [8, 43, 40] and multi-expert gating architectures [53, 13].

B. Sim-to-Real Evaluation Suites

Evaluation frameworks for locomotion models are currently limited. In contrast, research in robotic manipulation has addressed similar challenges by employing ranking metrics to verify consistency between simulation and reality [26, 49]. High-fidelity digital twins provide closed-loop assessment through environmental reconstruction but often suffer from high costs that restrict their scalability across diverse real-world scenarios [25, 58].

III. MoE Latent Representation Learning

The proposed one-stage reinforcement learning framework centers on Mixture-of-Experts latent representation learning for quadrupedal locomotion, as illustrated in the training phase of Fig. 1. This section describes the mathematical formulation of the motion control task and the internal structural design of the multi-expert neural network architecture, followed by the detailed reward configurations and environment configurations.

A. Locomotion Control in Reinforcement Learning

The core objective of quadrupedal locomotion control is to determine appropriate joint torque commands for all actuated joints based on proprioception. Assuming that proprioceptive information is acquired exclusively via an IMU and joint encoders, the quadrupedal locomotion dynamics are modeled as an infinite-horizon Partially Observable Markov Decision Process (POMDP), defined by the tuple $(\mathcal{S}, \mathcal{A}, \mathcal{O}, P, \Omega, R, \rho_0)$, where $\mathcal{S} \subset \mathbb{R}^n$ denotes the privileged state space including all dynamic information of robot perception and the surrounding environment. $\mathcal{A} \subset \mathbb{R}^m$ represents the action space and $\mathcal{O} \subset \mathbb{R}^o$ signifies the observation space. $P(\boldsymbol{s}'|\boldsymbol{s}, \boldsymbol{a})$ characterizes the state transition probability, $\Omega(\boldsymbol{o}|\boldsymbol{s})$ constitutes the observation function, $R(\boldsymbol{s}, \boldsymbol{a}, \boldsymbol{s}')$ defines the reward function, and $\rho_0(\boldsymbol{s}_0)$ indicates the initial state distribution. Our objective is to acquire an optimal policy π^* that maximizes the expected cumulative discounted reward over the trajectory $\tau = \{\boldsymbol{s}_t, \boldsymbol{a}_t, r_t, \boldsymbol{s}_{t+1}, ...\}$:

$$J(\pi) = \mathbb{E}_{\boldsymbol{s}_0 \sim \rho_0, \tau \sim \pi} \left[\sum_{t=0}^{\infty} \gamma^t R(\boldsymbol{s}_t, \boldsymbol{a}_t, \boldsymbol{s}_{t+1}) \right] \tag{1}$$

where $\gamma \in (0, 1)$ serves as the discount factor.

Let $\boldsymbol{o}_t \in \mathcal{O}$ and $\boldsymbol{s}_t \in \mathcal{S}$ denote the observation and state at time t, respectively. The observation incorporates the angular velocity $\boldsymbol{\omega}$ measured by the IMU, the projected gravity vector $\boldsymbol{g}_{\text{proj}}$ in the body frame, joint positions $\boldsymbol{q}$, and joint velocities $\dot{\boldsymbol{q}}$, linear velocity commands in the longitudinal and lateral directions v_x^{cmd} and v_y^{cmd}, the yaw rate command ω_z^{cmd}, and the preceding action $\boldsymbol{a}_{t-1}$. Beyond the components of $\boldsymbol{o}_t$, the state $\boldsymbol{s}_t$ encompasses the linear velocity $\boldsymbol{v}_t$, sampled terrain

heights $\boldsymbol{h}_t$, and environmental latent parameters $\boldsymbol{\mu}_t$ representing foot contact forces, joint torques, and joint accelerations. The height measurements are sampled within a 1m $\times$ 1.6m rectangular area centered on the robot's base with a 0.1m interval, providing a comprehensive representation of the local terrain.

The action $a_t \in \mathcal{A}$ denotes the joint position offsets relative to the initial joint positions. For each actuated joint, the model produces target positions, and the required torques are computed through a Proportional-Derivative (PD) controller.

B. Mixture-of-Experts Representation Encoder

To facilitate the acquisition of an optimal policy, privileged observations $\boldsymbol{s}_t$ are commonly employed during training to accelerate learning and elevate performance upper bounds. Given that the model is restricted to observations $\boldsymbol{o}_t$ during deployment, the teacher-student paradigm leverages distillation techniques to transfer advantageous strategies to the student [24]. The Concurrent Teacher-Student (CTS) framework [50] simultaneously optimizes both teacher and student networks. Through this parallel learning process, both entities update actor and critic networks, enabling student feedback to actively refine the teacher's parameters. Such joint optimization typically yields outcomes superior to those achieved through independent training [57]. We observe that the limited expressive capacity of the student model often precludes it from accurately inferring the features encoded by the teacher, which consequently restricts the performance ceiling. To overcome this limitation, we integrate a Mixture-of-Experts (MoE) structure [15, 17] into the student architecture within the CTS framework. This augmentation bolsters the representational capabilities of the student and further elevates the performance upper bound of the overall system.

We substitute the student encoder in the CTS framework with the MoE network. This architecture comprises K parallel expert subnetworks $\{E_k\}_{k=1}^{K}$ where each expert specializes in processing observation data under specific command types or environmental contexts. To coordinate these subnetworks, we incorporate a gating network g that dynamically allocates weights ω_k based on the observation $\boldsymbol{o}_t$. These coefficients determine the relative contribution of each expert to the current state representation. Accordingly, the resulting latent state $\boldsymbol{z}_s$ of the student encoder is formulated as the weighted sum of all expert outputs:

$$\boldsymbol{z}_s = \sum_{k=1}^{K} \omega_k E_k(\boldsymbol{o}_t), \quad \omega_k = \text{softmax}(g(\boldsymbol{o}_t))_k \tag{2}$$

To prevent the gating network from exclusively activating a single expert subnetwork, we incorporate an auxiliary load balancing loss [7, 44]:

$$\mathcal{L}_{\text{load balance}} = \sum_{k=1}^{K} \left(\bar{\omega}_k - \frac{1}{K} \right)^2, \quad \bar{\omega}_k = \frac{1}{B} \sum_{j=1}^{B} \omega_k^{(j)} \tag{3}$$

where B specifies the batch size utilized during training while $\omega_k^{(j)}$ represents the weight allocated to the k-th expert for the j-th sample. This formulation encourages the system to distribute tasks uniformly across all experts to ensure representational diversity and expressive capacity.

C. Reward Design

We utilize a consistent reward function structure for both the multi-terrain and the flat-ground high-speed locomotion models. The fundamental reward configurations are established based on established methodologies [24, 41, 50]. Building upon these foundations, we introduced a hip joint position reward to mitigate outward thigh abduction during rapid locomotion. Appendix Table VIII presents the comprehensive reward specifications. Within this framework, σ denotes the velocity tracking precision parameter initialized to a value of 0.25. Additionally, the reward component r^{fr} adopts the formulation from the CTS model [50] to incentivize adequate foot clearance during high-speed movement. For high-speed locomotion training on flat ground, we introduce an external hip symmetry reward r^{hs} to regularize joint positions while executing longitudinal linear motion commands. This term ensures that the robot maintains symmetrical postures and is defined as follows:

$$r^{\text{hs}} = \frac{|v_x^{\text{cmd}}|}{\|\boldsymbol{v}^{\text{cmd}}\|_2} \cdot \left(|q_{\text{FL}}^{\text{hip}} + q_{\text{FR}}^{\text{hip}}| + |q_{\text{RL}}^{\text{hip}} + q_{\text{RR}}^{\text{hip}}| \right) \tag{4}$$

Since the training curriculum involves diverse terrains, the vertical linear velocity reward weight decays to zero once the robot achieves stable locomotion. This reduction prevents vertical velocity fluctuations caused by terrain irregularities from interfering with the policy optimization process. We observed that augmenting the base height reward weight effectively mitigates body sagging during high-speed locomotion on flat surfaces. For the multi-terrain model, the reference base height is established at 0.38m. In contrast, the high-speed model utilizes a lower reference height of 0.33m to enhance the stability of the center of mass through a reduced posture.

D. Environment Configurations

We utilize the IsaacGym simulation environment [29] to train 8192 agents in parallel across diverse terrains. The experimental platform is the Unitree Go2 quadrupedal robot featuring 12 degrees of freedom. Motor PD control gains are specified as $k_{\text{p}} = 20.0$ and $k_{\text{d}} = 0.5$ for all joints. The system operates with a control frequency of 50Hz and a simulation frequency of 200Hz.

Establishing a proper curriculum difficulty is essential to ensure representational diversity during training. Following [41], we implement a terrain curriculum encompassing seven terrains including flat, wave, slope, rough slope, stairs up, stairs down, and obstacle. Slope inclinations vary from 5.7° to 29.6° and the rough slope terrain incorporates random height fluctuations of 5cm. Stair heights range between 5cm and 25.7cm with a constant tread width of 31cm. The obstacle terrain consists of random cubic structures with heights spanning from 5cm to 27.5cm and widths between 1m and 2m.

To facilitate effective sim-to-real transfer, we introduce domain randomization parameters, the details of which are shown in Table I.

TABLE I: Domain Randomization Specifications

Randomization Term	Range	Unit
Friction	$[0.5, 1.5]$	–
Payload mass	$[-1, 1]$	kg
Link mass	$[0.9, 1.1] \times$ Nominal Value	kg
Base center of mass	$[-3, 3] \times [-3, 3] \times [-3, 3]$	cm
Restitution	$[0.0, 0.5]$	–
Proportional gain k_p	$[0.9, 1.1] \times$ Nominal Value	Nm/rad
Derivative gain k_d	$[0.9, 1.1] \times$ Nominal Value	Nm $\cdot$ s/rad
Actuator strength	$[0.8, 1.2] \times$ Nominal Value	–
Actuator offset	$[-0.035, 0.035]$	rad
Control latency	$[0, 20]$	ms

We identify several training problems within the original framework [41, 36, 28, 50] which are elaborated in Appendix B along with corresponding ablation studies to verify the effectiveness of our improvements. To ensure reward stability on complex terrains we implement a *dynamic velocity tracking precision adjustment* B-A that scales constraints based on terrain difficulty and command magnitude. We further incorporate a comprehensive command design suite including a *command curriculum*, *extreme command sampling* and *dynamic command sampling* B-B to ensure consistent progression through terrain levels. These strategies collectively accelerate convergence and elevate the peak RoboGauge score by 11% while promoting stable locomotion patterns across diverse environments.

IV. The RoboGauge Predictive Assessment Framework

As illustrated in the central evaluation module of Fig. 1, RoboGauge serves as the pivotal assessment engine designed to bridge the gap between simulation training and real-world deployment. This section details the design philosophy of RoboGauge, a comprehensive framework developed to quantitatively validate the performance of reinforcement learning (RL) locomotion controllers.

Built upon the MuJoCo [48] simulation environment, the framework's operational workflow is depicted in Fig. 3, which organizes the evaluation process into three hierarchical stages: (1) the BasePipeline for atomic, single-environment evaluations; (2) the Multi/Level Pipeline for parallelized difficulty assessment and domain randomization; and (3) the Stress Pipeline for synthesizing a unified robustness score. The following subsections detail the formulation of our quantitative metrics, the design of the evaluation environments, and the hierarchical scoring methodology, respectively.

A. Quantitative Performance Metrics

The primary objective of RoboGauge is to derive quantitative indicators solely from proprioceptive feedback that accurately reflect a controller's efficacy during real-world deployment. Drawing from empirical observations of common

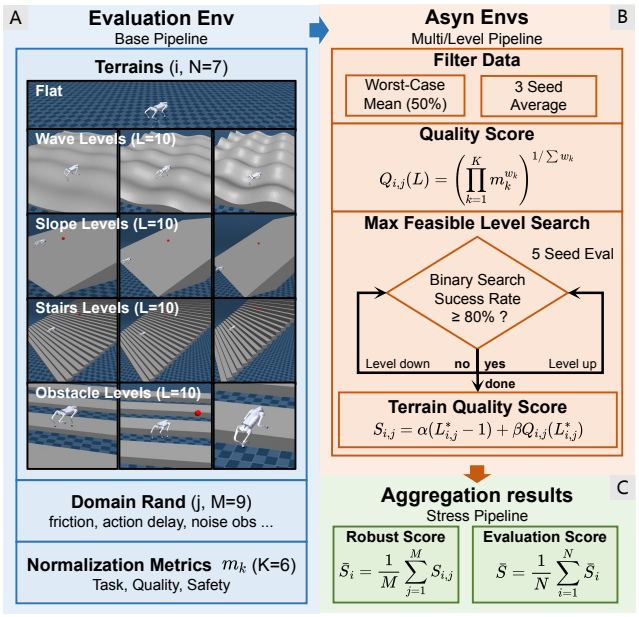


Fig. 3: The RoboGauge evaluation architecture consists of three hierarchical stages. (**A**) Base Pipeline serves as a single evaluation environment by incorporating specific terrains and domain randomization. (**B**) Multi/Level Pipeline highlights the Multi/Level Pipeline where parallel evaluations across diverse random seeds. (**C**) Stress Pipeline depicts the Stress Pipeline which triggers comprehensive testing across the entire terrain suite to synthesize the final score.

TABLE II: Metrics for the RoboGauge Framework

Metric	**Description**
Lin. Velocity Error	Linear velocity ℓ_2 tracking error
Ang. Velocity Error	Angular velocity ℓ_2 tracking error
Dof Power	Motor power consumption
Dof Limits	Joint angles exceeding soft limits
Orientation Stability	Gravity projection on the lateral (y) axis
Torque Smoothness	Temporal smoothness of motor torques

failure modes in physical testing, we formulated 6 metrics, as detailed in Table II, addressing three critical aspects of sim-to-real transfer. First, to ensure **hardware safety and efficiency**, we evaluate *dof limits* and *dof power*, preventing actuator damage or thermal failure caused by sub-optimal motor operation. Second, **tracking precision** is quantified by the *velocity error*, measuring the controller's fidelity in following linear and angular commands. Finally, we assess **motion stability** via *torque smoothness* and *orientation stability* to mitigate structural vibrations and ensure robust attitude control. To facilitate a unified assessment, all raw measurements are normalized and transformed via the function $f(x) = 1 - x$, ensuring that a higher score consistently signifies superior performance.

B. Evaluation Environment and Randomization

To ensure a rigorous and holistic assessment, the framework establishes a systematic evaluation matrix integrating diverse

motion goals, complex terrain structures, and extensive domain randomizations.

Motion Goals: We devised motion goals to stress-test the control policy as detailed in Appendix Table VII. These tasks cover maximum command execution, rapid emergency stops, and abrupt diagonal velocity step changes. Furthermore, the evaluation incorporates a specific target position task regulated by a proportional error controller. This task serves as the pass criterion for terrain traversal. It enables a binary search strategy to identify the maximum difficulty level the model can navigate.

Terrain Configuration: The evaluation suite features 5 distinct terrain categories: flat, wave, slopes, stairs, and obstacles. Excluding the flat surface, each terrain type is subdivided into 10 discrete difficulty levels to probe the limits of the controller's mobility. Fig. 3 explicitly illustrates the environmental complexity for difficulty levels 3, 5, and 10. Beyond difficulty scaling, navigation on slopes and stairs presents unique directional challenges. Therefore, we explicitly evaluate both ascending and descending configurations to ensure robust performance regardless of the incline direction.

Domain Randomization: We implement domain randomization across two primary dimensions environmental factors and inherent robot properties. Specifically, environmental factors include variations such as payloads and friction coefficients, while robot properties encompass motor response latency and observation noise. Collectively, these perturbations simulate the imperfections of physical hardware, preventing the policy from overfitting to ideal simulation dynamics and ensuring robust real-world transfer.

C. Hierarchical Scoring Methodology

We denote the set of $N = 7$ terrain configurations as $\mathcal{T} = \{T_1, \ldots, T_N\}$, expanding the five distinct terrain categories by treating ascending and descending directions on slopes and stairs as separate evaluation environments. For each terrain $T \in \mathcal{T}$, we apply $M = 9$ distinct domain randomizations, denoted by $\mathcal{D} = \{d_1, \ldots, d_M\}$. The terrain difficulty is stratified into 10 levels, represented as $L \in \{1, 2, \ldots, 10\}$. Each evaluation session yields $K = 6$ performance metrics, designated as $\mathcal{M} = \{m_1, \ldots, m_K\}$.

Next, we formalize the composite scoring methodology for evaluating the model. For a given terrain T_i, domain randomization d_j, and difficulty level L, we aggregate $K = 6$ normalized metrics $\{m_1, \ldots, m_6\}$, where each $m_k \in [0, 1]$ denotes the average result across three stochastic seeds. To penalize imbalanced performance, specifically to prevent high scores when a critical dimension fails, we employ a weighted geometric mean to compute the execution quality score:

$$Q_{i,j}(L) = \left(\prod_{k=1}^{K} m_k^{w_k} \right)^{1/\sum_{k=1}^{K} w_k} \tag{5}$$

We adopt a Worst-Case Mean aggregation strategy to evaluate performance across motion goals. This method involves averaging the lowest 50% of scores within each goal, effectively discounting high scores from non-challenging commands to concentrate the assessment on challenging maneuvers such as obstacle negotiation and gait transitions. Additionally, we compute the global mean and the average of the top 25% for broader reference as detailed in Appendix Table XII.

We employ a binary search strategy to identify the maximum attainable difficulty level $L^*_{i,j} \in \mathcal{L}$ for each terrain under the specified domain randomization parameters. For a given level, the model is evaluated across five stochastic seeds to verify whether it successfully reaches the goal. A difficulty level is deemed passable if the success rate in the goal-reaching task surpasses 80%.

Let $Q_{i,j}(L^*_{i,j})$ denote the execution quality score at the highest passable difficulty level. To balance task difficulty and execution quality across diverse terrains, the terrain quality score $S_{i,j}$ for a specific terrain T_i and domain randomization d_j is formulated using the following overlapping scoring function:

$$S_{i,j} = \alpha(L^*_{i,j} - 1) + \beta Q_{i,j}(L^*_{i,j}) \tag{6}$$

By setting $\beta > \alpha$, this design ensures that high-quality performance at a lower difficulty level approximates the score of mediocre performance at a higher level, facilitating a smooth transition across difficulty tiers.

The framework results are aggregated through arithmetic averaging. Initially, we calculate the robust score $\bar{S}_i$ for each terrain T_i by averaging the results over M domain randomizations. The final framework score $\bar{S}$ is subsequently obtained by averaging these robust scores across all N terrains:

$$\bar{S}_i = \frac{1}{M} \sum_{j=1}^{M} S_{i,j}, \quad \bar{S} = \frac{1}{N} \sum_{i=1}^{N} \bar{S}_i \tag{7}$$

Given the extensive combinations of terrain types, randomization parameters, and random seeds, performing a full evaluation sequentially is prohibitively time-consuming. We consequently adopt multiprocessing acceleration to run concurrent environment instances. This efficiency fulfills the necessity for rapid performance feedback throughout the training phase. Further implementation specifics and all specific hyperparameter values are elaborated in Appendix A.

V. Framework Validation and Ablation Studies

In this section, we present experiments aimed at addressing the following research questions:

- **Q1**: Does RoboGauge provide metrics that correlate closely with real-world performance?
- **Q2**: How do state-of-the-art methods perform under our evaluation framework?
- **Q3**: Can the Mixture of Experts architecture effectively differentiate between various encoded terrains?

A. Metric Reliability of RoboGauge

We deployed the proposed model and baselines on a Unitree Go2 quadruped robot. We utilize a high-precision motion capture system operating at 90Hz to acquire real-time linear and angular velocity data across flat terrain and 10cm stairs

by mounting five markers on the robot base. At the same time, we gather proprioceptive feedback and motor torques to derive the six specific metrics in Table II. To quantify the fidelity of these assessment methods, we compare the metric errors from both the training environment and our proposed framework against real-world ground truth. We specifically evaluate a model that exhibited high performance during training but suffered from significant sim-to-real degradation. As presented in Table III, the training environment consistently yields larger errors. Comprehensive scoring data provided in Table XI in the Appendix further confirms that errors obtained through our framework are markedly lower than those from standard training evaluations. These results demonstrate that our evaluation framework more accurately reflects real-world performance and provides a more dependable basis for model selection.

TABLE III: Metrics Error Comparison

Env.	Cmd.	Tracking ↓	Safety ↓	Quality ↓
MuJoCo (**Ours**)	Longitudinal	0.0573	0.0253	0.0246
	Lateral	0.0541	0.0049	0.0079
	Angular	0.0560	0.0050	0.0035
	Average	**0.0558**	**0.0117**	**0.0120**
IsaacGym (Training)	Longitudinal	0.1365	0.0844	0.0678
	Lateral	0.0572	0.0052	0.0125
	Angular	0.0713	0.0103	0.0337
	Average	0.0883	0.0333	0.0380

B. Comparison of Baselines under RoboGauge

To facilitate a rigorous comparative evaluation, we benchmark our proposed approach against several state-of-the-art one-stage training algorithms based solely on proprioception:

1) DreamWaQ [36]: The policy utilizes an asymmetric actor-critic scheme with a variational estimator to jointly predict body velocity and terrain latents.
2) HIM [28]: The policy incorporates a hybrid internal model to explicitly estimate robot responses using contrastive learning.
3) CTS [50]: The policy employs an asymmetric teacher-student setup to optimize the agent via reinforcement learning and supervised reconstruction.

We implement all aforementioned methods using a consistent configuration, with 8192 parallel agents training in IsaacGym [29]. Because DreamWaQ and HIM do not support terrain-specific velocity command ranges, we set their maximum limit to 1 m/s. We apply this same constraint within the RoboGauge assessment for these models to reduce the difficulty of command tracking. Conversely, both CTS and our proposed model utilize a command range of 2 m/s for both training and evaluation. Each algorithm is trained with three independent random seeds and we select the model achieving the highest RoboGauge score for subsequent analysis. The outcomes summarized in Table IV demonstrate that our method significantly outperforms the other approaches across the entire set of metrics.

TABLE IV: RoboGauge results for baselines

Model	Score	Tracking ↑	Safety ↑	Quality ↑	Level
Ours	**0.67**	**0.66 ± 0.24**	**0.78 ± 0.25**	**0.77 ± 0.25**	**7.81**
CTS	0.58	0.57±0.23	0.70±0.25	0.69±0.25	6.81
HIM	0.53	0.51±0.26	0.63±0.28	0.63±0.28	5.94
DWaQ	0.47	0.45±0.27	0.57±0.31	0.56±0.30	5.15

As indicated in the training curves in Fig. 4, our model does not necessarily achieve the highest terrain levels during the training phase compared to other baselines. Nevertheless, the predictability assessment framework provides precise scores that accurately reflect the underlying performance. Fig. 5 illustrates the maximum terrain levels attained across a variety of friction coefficients. Details of the terrain levels are provided in Fig. 15 of the Appendix. Our model consistently exhibits superior terrain level proficiency across the entire range of friction values. These findings are further corroborated by the real-world deployment data in Table VI, which confirms that the controller possesses the capability to navigate such challenging environments in physical settings.

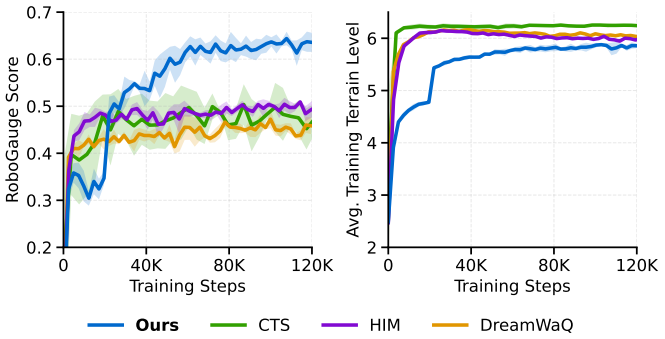


Fig. 4: Comparison of RoboGauge scores and terrain level curves across various baselines during training. Stable RoboGauge scores despite fluctuating terrain levels demonstrate that training levels fails to accurately represent model performance.

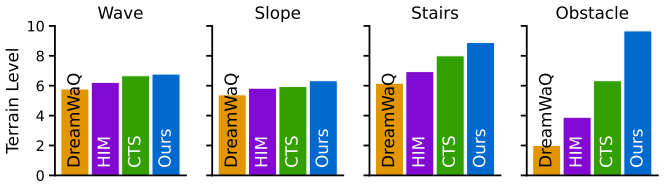


Fig. 5: Comparison of maximum terrain levels across varying friction coefficients as evaluated by RoboGauge.

C. Ablation and Latent Representation of MoE

We designed various ablation studies to investigate the integrated MoE structure, including the following variants:

1) MoE-NG: The command information is excluded from the MoE input, utilizing only observation information to the expert networks.
2) AC-MoE: Following MoE-Loco [13], the MoE structure is applied to the Actor-Critic networks rather than the

student encoder.

3) MCP [39]: A multiplicative composition strategy is employed for the actions output by the Actor.

As shown in Table V, our proposed method achieved the best performance across all evaluation metrics. Furthermore, during training, we observed that modifications to the action network, such as AC-MoE and MCP, were prone to loss divergence. This instability likely originates from the expert combination acting directly within the action space. The concurrent adaptation of the gating network and individual experts can yield volatile control signals that induce hazardous maneuvers and consequently undermine training stability.

TABLE V: RoboGauge Results for MoE Ablation

Model	Score	Tracking	Safety	Quality	Level
MoE (Ours)	**0.6745**	**0.6574**	**0.7765**	**0.7722**	**7.81**
MoE-NG	0.6637	0.6450	0.7651	0.7615	7.67
AC-MoE [13]	0.6589	0.6402	0.7601	0.7538	7.56
MCP [39]	0.6513	0.6343	0.7559	0.7504	7.52

We subsequently visualize the MoE latent space by applying Principal Component Analysis [37] to reduce the dimensionality of the student encoder hidden states. Fig. 6 contrasts the state distributions during 5 s of forward locomotion across diverse terrains to evaluate the impact of the MoE module. Similarly, Fig. 17 in the Appendix illustrates the hidden state distributions across all terrains under various commands including forward, backward, left, and right turns over a 5 s duration. These results indicate that the MoE architecture achieves superior discrimination of encoding features across various terrains and motion commands.

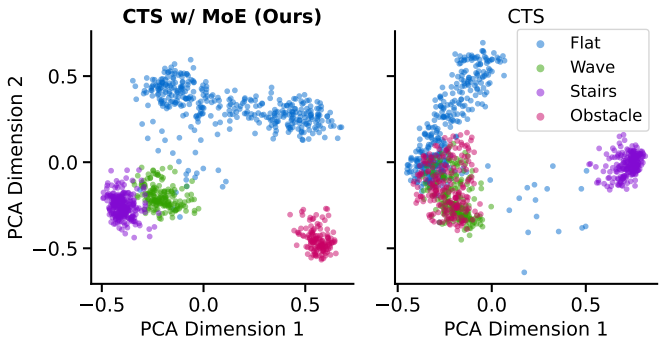


Fig. 6: PCA visualization of the student encoder latent space in different terrains with forward command.

VI. Physical Deployment and Generalization

In this section, our real-world experiments are designed to address the following research questions.

- **Q4**: Does the proposed framework outperform more challenging terrain compared to other baselines?
- **Q5**: How accurate is its tracking of velocity commands?
- **Q6**: Can the model perform reliably in diverse complex environments not encountered during training?

TABLE VI: Real-World Survival Rate Comparison

Model	Survival Rate (%) ↑		
	Lat. Impulse (80–100 N)	**Tile Stairs 15.5cm** ($\mu = 0.38$)	**Obstacle 30cm** ($\mu = 0.85$)
Ours	**18/20**	**85/85**	**17/20**
Built-in RL	5/20	**85/85**	0/20
CTS	11/20	18/85	0/20
HIM	8/20	24/85	0/20
DreamWaQ	7/20	12/85	0/20

A. Comparison on Terrain Challenges

We deployed the proposed model and baselines on a Unitree Go2 quadruped robot to evaluate its real-world performance as summarized in Table VI. The experimental validation comprises three robustness scenarios including sudden lateral pulls between 80 N and 100 N then 15.5 cm smooth tile stairs and 30 cm obstacle climbing where Appendix Fig. 18 depicts the specific setups. Only our model successfully surmounted the 30 cm obstacle while also exhibiting the most effective disturbance rejection during lateral pulls. Although both our approach and the built-in reinforcement learning controller conquered the stairs, our model completed the 85 steps 17 s faster than the baseline.

B. Velocity Tracking Precision

We employed a motion capture system to assess velocity tracking accuracy across both flat terrain and stair scenarios. Fig. 7 depicts the robot traversing stairs at an average speed of 1.31 m/s with a tracking error of 0.15 m/s, which confirms the robust tracking proficiency of the framework even when tackling complex environments. We further evaluated the locomotion performance on a 30 degree wooden slope where the robot maintains an average velocity of 1.53 m/s. This efficiency reduces the traversal duration by 1.7 s compared to the built-in reinforcement learning baseline as documented in Fig. 8 of the Appendix.

Fig. 9 illustrates the tracking performance during high-speed locomotion on flat ground. Restricted by an 8 m indoor runway, the robot attains a peak velocity of 4.01 m/s within 2.16 s with a tracking error of 0.20 m/s, which demonstrates exceptional acceleration and braking capabilities. Notably the model autonomously develops a stable narrow-base gait despite the absence of explicit motion constraints to minimize lateral center-of-mass oscillations and bolster stability during high-speed maneuvers.

C. Stability and Generalization

We validated the emergency recovery capabilities of the proposed model across two challenging real-world scenarios. First, the robot is subjected to external forces such as strong pushes or pulls where it shows great disturbance rejection by changing its center of mass and creating gaits to offset the impact. Fig. 10 and 18 show that the robot remains stable under continuous lateral pulls between 25 N and 40 N as well as sudden impulses of 85 N to 100 N where established baselines almost entirely fail to maintain balance. Second,

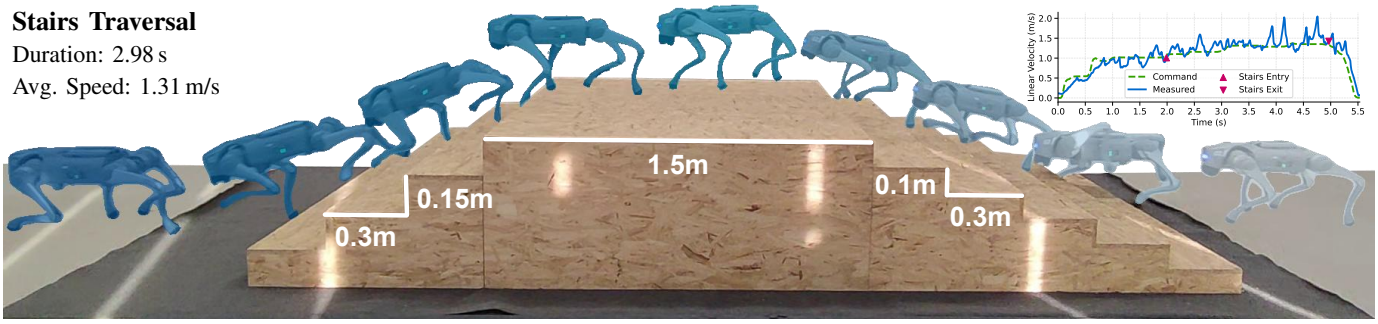


Fig. 7: Experiment on wooden stairs with a 10 cm rise and 15 cm drop. The upper-right plot depicts the velocity tracking curve captured through a motion capture system where the tracking error is 0.15 m/s.

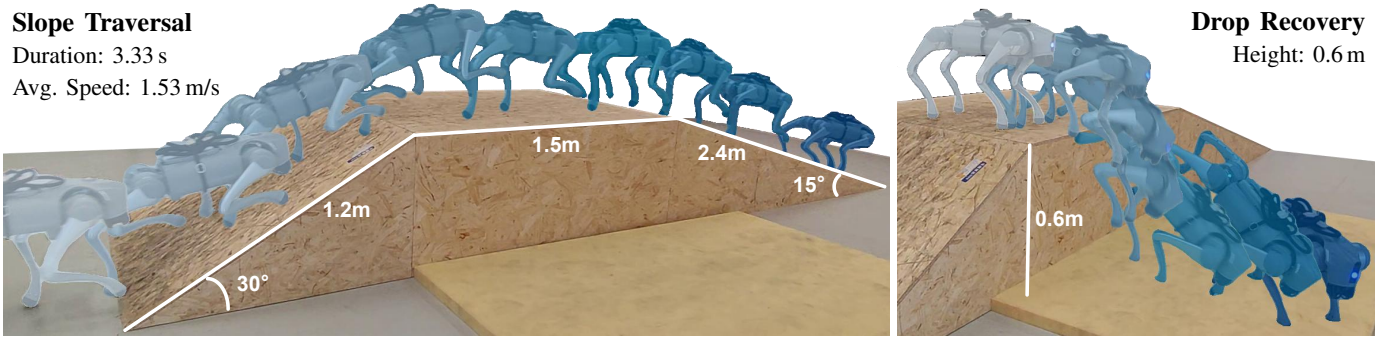


Fig. 8: Robust locomotion during slope traversal and drop recovery. The left panel highlights a 1.7 s efficiency gain on $\mu = 0.55$ slopes compared to the built-in RL baseline and the right frame verifies reliable recovery from 60 cm drops.

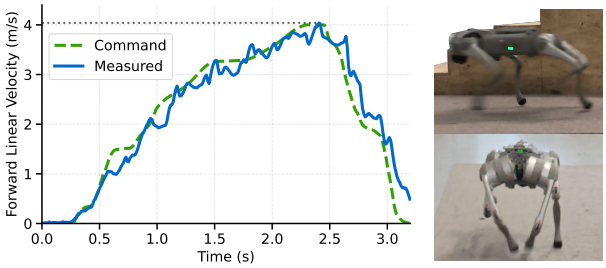


Fig. 9: Velocity tracking and gait on a $\mu = 0.6$ surface. The left plot exhibits command following reaching 4.01 m/s within 2.16 s with a 0.20 m/s error. The upper-right image captures transient flight phases while the lower-right image highlights a stable narrow-base gait.

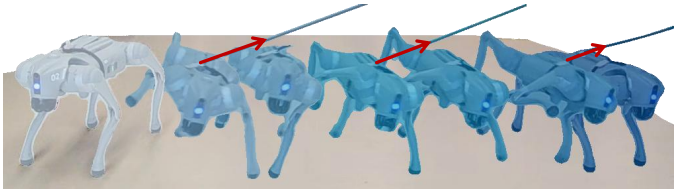


Fig. 10: Continuous lateral pull disturbance rejection experiment on flat terrain. The robot endures repeated lateral pulls of approximately 25 $\sim$ 40N while maintaining stable locomotion.

when encountering a sudden loss of support the robot rapidly reconfigures its gait to secure its footing and prevent forward tumbling. Fig. 8 illustrates a successful recovery sequence from a 60 cm drop while Fig. 14 depicts the natural transition to a stable posture after an unexpected fall from flat ground onto stairs.

Finally we conducted field tests in diverse outdoor environments to evaluate the generalization capabilities of the framework. The right panel of Fig. 1 illustrates the performance across various terrains such as sand and ice as well as slopes and uneven terrains. The robot completed all trials with a 100% success rate and zero unexpected terminations which highlights the exceptional robustness of the learned policy.

VII. Conclusions and Future Work

In this work, we presented a training framework comprising the RoboGauge assessment suite and an MoE locomotion policy which enables robust multi-terrain locomotion relying solely on proprioception. Physical experiments on a Unitree Go2 robot demonstrate that our framework successfully surmounts challenging environments including 30 cm obstacles and 100 N impulses, while utilizing the identical training configuration on flat ground to attain a peak velocity of 4.01 m/s. The framework consistently outperforms established baselines in both tracking precision and recovery stability with a 100% success rate in diverse outdoor field tests. This synergy between predictive assessment and modular architecture provides a reliable and efficient way to bridge the gap between simulation results and actual physical performance.

Future research will extend RoboGauge to broader morphologies like humanoid robots and integrate exteroceptive perception with the MoE representation to further improve the crossing of complex structural obstacles.

References

[1] OpenAI: Marcin Andrychowicz, Bowen Baker, Maciek Chociej, Rafal Jozefowicz, Bob McGrew, Jakub Pachocki, Arthur Petron, Matthias Plappert, Glenn Powell, Alex Ray, et al. Learning dexterous in-hand manipulation. *The International Journal of Robotics Research*, 39 (1):3–20, 2020.

[2] Guillaume Bellegarda, Milad Shafiee, and Auke Ijspeert. Allgaits: Learning all quadruped gaits and transitions. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 15929–15935. IEEE, 2025.

[3] Russell Buchanan, Lorenz Wellhausen, Marko Bjelonic, Tirthankar Bandyopadhyay, Navinda Kottege, and Marco Hutter. Perceptive whole-body planning for multilegged robots in confined spaces. *Journal of Field Robotics*, 38 (1):68–84, 2021.

[4] Yevgen Chebotar, Ankur Handa, Viktor Makoviychuk, Miles Macklin, Jan Issac, Nathan Ratliff, and Dieter Fox. Closing the sim-to-real loop: Adapting simulation randomization with real world experience. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 8973–8979. IEEE, 2019.

[5] Jia-Ruei Chiu, Jean-Pierre Sleiman, Mayank Mittal, Farbod Farshidian, and Marco Hutter. A collision-free mpc for whole-body dynamic locomotion and manipulation. In *2022 international conference on robotics and automation (ICRA)*, pages 4686–4693. IEEE, 2022.

[6] Thomas Dudzik, Matthew Chignoli, Gerardo Bledt, Bryan Lim, Adam Miller, Donghyun Kim, and Sangbae Kim. Robust autonomous navigation of a small-scale quadruped robot in real-world environments. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3664–3671. IEEE, 2020.

[7] William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(120):1–39, 2022.

[8] Zipeng Fu, Ashish Kumar, Jitendra Malik, and Deepak Pathak. Minimizing energy consumption leads to the emergence of gaits in legged robots. In *Conference on Robot Learning*, pages 928–937. PMLR, 2022.

[9] Magnus Gaertner, Marko Bjelonic, Farbod Farshidian, and Marco Hutter. Collision-free mpc for legged robots in static and dynamic scenes. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 8266–8272. IEEE, 2021.

[10] Sehoon Ha, Peng Xu, Zhenyu Tan, Sergey Levine, and Jie Tan. Learning to walk in the real world with minimal human effort. In *Conference on Robot Learning*, pages 1110–1120. PMLR, 2021.

[11] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997. doi: 10.1162/neco.1997.9.8.1735.

[12] David Hoeller, Lorenz Wellhausen, Farbod Farshidian, and Marco Hutter. Learning a state representation and navigation in cluttered and dynamic environments. *IEEE Robotics and Automation Letters*, 6(3):5081–5088, 2021.

[13] Runhan Huang, Shaoting Zhu, and Yilun Du. Moe-loco: Mixture of experts for multitask locomotion. In *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 14218–14225, 10 2025. doi: 10.1109/IROS60139.2025.11246585.

[14] Jemin Hwangbo, Joonho Lee, Alexey Dosovitskiy, Dario Bellicoso, Vassilios Tsounis, Vladlen Koltun, and Marco Hutter. Learning agile and dynamic motor skills for legged robots. *Science Robotics*, 4(26):eaau5872, 2019.

[15] Robert A Jacobs, Michael I Jordan, Steven J Nowlan, and Geoffrey E Hinton. Adaptive mixtures of local experts. *Neural computation*, 3(1):79–87, 1991.

[16] Gwanghyeon Ji, Juhyeok Mun, Hyeongjun Kim, and Jemin Hwangbo. Concurrent training of a control policy and a state estimator for dynamic and robust legged locomotion. *IEEE Robotics and Automation Letters*, 7 (2):4630–4637, 2022.

[17] Michael I Jordan and Robert A Jacobs. Hierarchical mixtures of experts and the em algorithm. *Neural computation*, 6(2):181–214, 1994.

[18] Sertac Karaman and Emilio Frazzoli. Sampling-based algorithms for optimal motion planning. *The international journal of robotics research*, 30(7):846–894, 2011.

[19] Oussama Khatib. Real-time obstacle avoidance for manipulators and mobile robots. *The international journal of robotics research*, 5(1):90–98, 1986.

[20] Donghyun Kim, Daniel Carballo, Jared Di Carlo, Benjamin Katz, Gerardo Bledt, Bryan Lim, and Sangbae Kim. Vision aided dynamic exploration of unstructured terrain with a small-scale quadruped robot. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2464–2470. IEEE, 2020.

[21] Yunho Kim, Hyunsik Oh, Jeonghyun Lee, Jinhyeok Choi, Gwanghyeon Ji, Moonkyu Jung, Donghoon Youm, and Jemin Hwangbo. Not only rewards but also constraints: Applications on legged robot locomotion. *IEEE Transactions on Robotics*, 40:2984–3003, 2024.

[22] Sylvain Koos, Jean-Baptiste Mouret, and Stéphane Doncieux. The transferability approach: Crossing the reality gap in evolutionary robotics. *IEEE Transactions on Evolutionary Computation*, 17(1):122–145, 2012.

[23] Ashish Kumar, Zipeng Fu, Deepak Pathak, and Jitendra Malik. Rma: Rapid motor adaptation for legged robots. *Robotics: Science and Systems XVII*, 2021.

[24] Joonho Lee, Jemin Hwangbo, Lorenz Wellhausen, Vladlen Koltun, and Marco Hutter. Learning quadrupedal locomotion over challenging terrain. *Science robotics*, 5 (47):eabc5986, 2020.

[25] Xinhai Li, Jialin Li, Ziheng Zhang, Rui Zhang, Fan Jia, Tiancai Wang, Haoqiang Fan, Kuo-Kun Tseng, and Ruiping Wang. Robogsim: A real2sim2real robotic gaussian splatting simulator. *arXiv preprint arXiv:2411.11839*, 2024.

[26] Xuanlin Li, Kyle Hsu, Jiayuan Gu, Karl Pertsch, Oier

Mees, Homer Rich Walke, Chuyuan Fu, Ishikaa Lunawat, Isabel Sieh, Sean Kirmani, et al. Evaluating real-world robot manipulation policies in simulation. In *RSS 2024 Workshop: Data Generation for Robotics*.

[27] Qiayuan Liao, Zhongyu Li, Akshay Thirugnanam, Jun Zeng, and Koushil Sreenath. Walking in narrow spaces: Safety-critical locomotion control for quadrupedal robots with duality-based optimization. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2723–2730. IEEE, 2023.

[28] Junfeng Long, Zirui Wang, Quanyi Li, Liu Cao, Jiawei Gao, and Jiangmiao Pang. Hybrid internal model: Learning agile legged locomotion with simulated robot response. In *ICLR*, 2024.

[29] Viktor Makoviychuk, Lukasz Wawrzyniak, Yunrong Guo, Michelle Lu, Kier Storey, Miles Macklin, David Hoeller, Nikita Rudin, Arthur Allshire, Ankur Handa, et al. Isaac gym: High performance gpu based physics simulation for robot learning. In *NeurIPS Datasets and Benchmarks*, 2021.

[30] Gabriel B Margolis and Pulkit Agrawal. Walk these ways: Tuning robot control for generalization with multiplicity of behavior. In *Conference on Robot Learning*, pages 22–31. PMLR, 2023.

[31] Gabriel B Margolis, Ge Yang, Kartik Paigwar, Tao Chen, and Pulkit Agrawal. Rapid locomotion via reinforcement learning. In *Robotics: Science and Systems*, 2022.

[32] Matias Mattamala, Nived Chebrolu, and Maurice Fallon. An efficient locally reactive controller for safe navigation in visual teach and repeat missions. *IEEE Robotics and Automation Letters*, 7(2):2353–2360, 2022.

[33] Takahiro Miki, Joonho Lee, Jemin Hwangbo, Lorenz Wellhausen, Vladlen Koltun, and Marco Hutter. Learning robust perceptive locomotion for quadrupedal robots in the wild. *Science robotics*, 7(62):eabk2822, 2022.

[34] Prakhar Mishra, Amir Hossain Raj, Xuesu Xiao, and Dinesh Manocha. Hacl: History-aware curriculum learning for fast locomotion. *arXiv preprint arXiv:2505.18429*, 2025.

[35] Alexander Luis Mitchell, Wolfgang Merkt, Aristotelis Papatheodorou, Ioannis Havoutis, and Ingmar Posner. Gaitor: Learning a unified representation across gaits for real-world quadruped locomotion. In *8th Annual Conference on Robot Learning*, 2024.

[36] I Made Aswin Nahrendra, Byeongho Yu, and Hyun Myung. Dreamwaq: Learning robust quadrupedal locomotion with implicit terrain imagination via deep reinforcement learning. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5078–5084. IEEE, 2023.

[37] Karl Pearson. Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin philosophical magazine and journal of science*, 2 (11):559–572, 1901.

[38] Xue Bin Peng, Marcin Andrychowicz, Wojciech Zaremba, and Pieter Abbeel. Sim-to-real transfer of robotic control with dynamics randomization. In *2018 IEEE international conference on robotics and automation (ICRA)*, pages 3803–3810. IEEE, 2018.

[39] Xue Bin Peng, Michael Chang, Grace Zhang, Pieter Abbeel, and Sergey Levine. Mcp: Learning composable hierarchical control with multiplicative compositional policies. *Advances in neural information processing systems*, 32, 2019.

[40] Skand Peri, Akhil Perincherry, Bikram Pandit, and Stefan Lee. Non-conflicting energy minimization in reinforcement learning based robot control. In *9th Annual Conference on Robot Learning*, 2025.

[41] Nikita Rudin, David Hoeller, Philipp Reist, and Marco Hutter. Learning to walk in minutes using massively parallel deep reinforcement learning. In *Conference on robot learning*, pages 91–100. PMLR, 2022.

[42] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[43] Milad Shafiee, Guillaume Bellegarda, and Auke Ijspeert. Viability leads to the emergence of gait transitions in learning agile quadrupedal locomotion on challenging terrains. *Nature Communications*, 15(1):3073, 2024.

[44] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*, 2017.

[45] Laura Smith, J Chase Kew, Xue Bin Peng, Sehoon Ha, Jie Tan, and Sergey Levine. Legged robots that keep on learning: Fine-tuning locomotion policies in the real world. In *2022 international conference on robotics and automation (ICRA)*, pages 1593–1599. IEEE, 2022.

[46] Zhi Su, Xiaoyu Huang, Daniel Ordoñez-Apraez, Yunfei Li, Zhongyu Li, Qiayuan Liao, Giulio Turrisi, Massimiliano Pontil, Claudio Semini, Yi Wu, et al. Leveraging symmetry in rl-based legged locomotion control. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6899–6906. IEEE, 2024.

[47] Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter 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)*, pages 23–30. IEEE, 2017.

[48] Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5026–5033, 2012. doi: 10.1109/IROS.2012.6386109.

[49] Wei-Cheng Tseng, Jinwei Gu, Qinsheng Zhang, Hanzi Mao, Ming-Yu Liu, Florian Shkurti, and Lin Yen-Chen. Scalable policy evaluation with video world models. *arXiv preprint arXiv:2511.11520*, 2025.

[50] Hongxi Wang, Haoxiang Luo, Wei Zhang, and Hua Chen. Cts: Concurrent teacher-student reinforcement learning

for legged locomotion. *IEEE Robotics and Automation Letters*, 2024.

[51] Jinze Wu, Guiyang Xin, Chenkun Qi, and Yufei Xue. Learning robust and agile legged locomotion using adversarial motion priors. *IEEE Robotics and Automation Letters*, 8(8):4975–4982, 2023.

[52] Philipp Wu, Alejandro Escontrela, Danijar Hafner, Pieter Abbeel, and Ken Goldberg. Daydreamer: World models for physical robot learning. In *Conference on robot learning*, pages 2226–2240. PMLR, 2023.

[53] Chuanyu Yang, Kai Yuan, Qiuguo Zhu, Wanming Yu, and Zhibin Li. Multi-expert learning of adaptive legged locomotion. *Science Robotics*, 5(49):eabb2174, 2020.

[54] Ruihan Yang, Minghao Zhang, Nicklas Hansen, Huazhe Xu, and Xiaolong Wang. Learning vision-guided quadrupedal locomotion end-to-end with cross-modal transformers. In *Deep RL Workshop NeurIPS 2021*.

[55] Chong Zhang, Jin Jin, Jonas Frey, Nikita Rudin, Matías Mattamala, Cesar Cadena, and Marco Hutter. Resilient legged local navigation: Learning to traverse with compromised perception end-to-end. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 34–41. IEEE, 2024.

[56] Chong Zhang, Nikita Rudin, David Hoeller, and Marco Hutter. Learning agile locomotion on risky terrains. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 11864–11871. IEEE, 2024.

[57] Ying Zhang, Tao Xiang, Timothy M Hospedales, and Huchuan Lu. Deep mutual learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4320–4328, 2018.

[58] Shaoting Zhu, Linzhan Mou, Derun Li, Baijun Ye, Runhan Huang, and Hang Zhao. Vr-robo: A real-to-sim-to-real framework for visual robot navigation and locomotion. *IEEE Robotics and Automation Letters*, 2025.

APPENDIX A
ROBOGAUGE SUPPLEMENTARY MATERIAL

A. Implementation Details

The operational logic for each pipeline is delineated below.

The `BasePipeline` orchestrates the interaction between the simulation engine `sim`, the evaluator `gauge` responsible for control commands and metric computation, and the locomotion model `robot`. Additionally, it manages exception handling, domain randomization, and the application of observation noise.

The `MultiPipeline` leverages multiprocessing to execute the `BasePipeline` across diverse seeds and domain randomization configurations while aggregating the output files. To determine the maximum navigable difficulty for a given terrain, the `LevelPipeline` identifies the highest level that the model traverses successfully across three separate random seeds.

In the quality score calculation Eq. 5, the metric weights are set to $w_k = 2$ for task-completion metrics and $w_k = 1$ to others. For the overlapping scoring function Eq. 6, the hyperparameters are set to $\alpha = 0.09$ and $\beta = 0.19$, which ensures the performance score is bounded within the range $[0, 1]$.

TABLE VII: Configuration of Locomotion Control Objectives

Goal Name	Description	Reset Condition	Max Trials
Max Velocity	Evaluation of peak linear or angular velocity in a single dimension.	Sudden stop after each directional command.	6
Diagonal Velocity	Tracking of coupled diagonal velocity vectors (combined linear and angular).	Completion of each pair of diagonal commands.	8
Target Pos. Velocity	Position-based tracking using a Proportional controller to reach targets.	Goal reached or time limit exceeded.	1

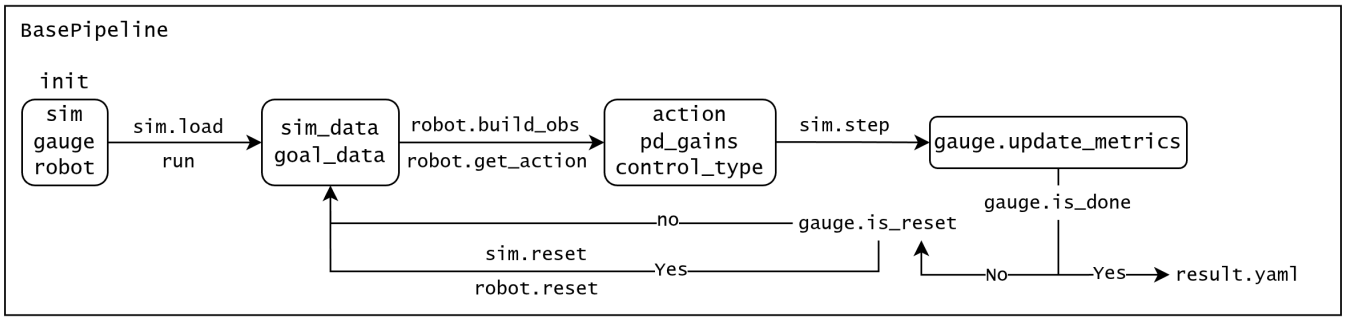


Fig. 11: Operational workflow of the `BasePipeline`.

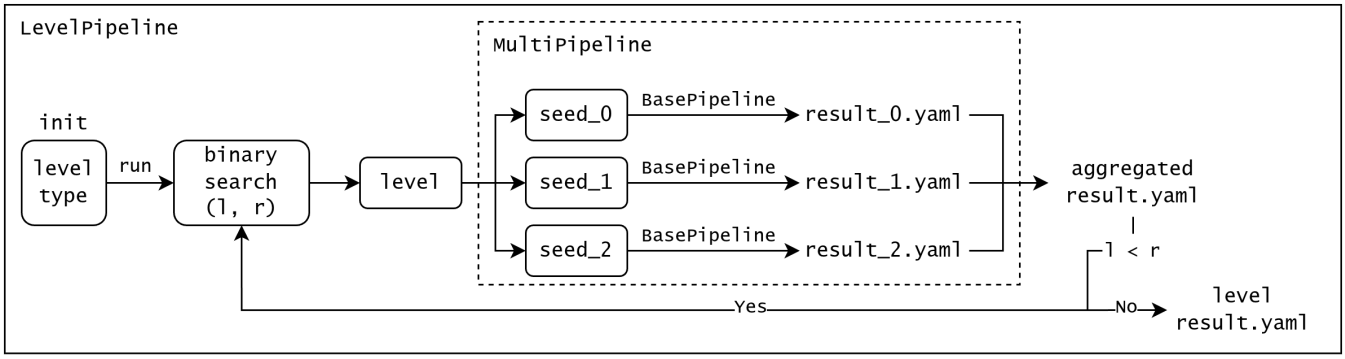


Fig. 12: Operational workflow of the `LevelPipeline`.

APPENDIX B
TRAINING DETAILS

A. Dynamic Velocity Tracking Precision Adjustment

To adapt the velocity tracking precision σ according to terrain characteristics and difficulty levels, we implement a dynamic scaling adjustment. We observe that as the maximum command range expands from 0.5 to 1.5, locomotion on challenging terrains such as wave, stairs, and obstacle often fails to accurately track the commanded linear velocity. Consequently, we scale the tracking coefficients to relax the tracking constraints for these scenarios.

We define $[v_{min}, v_{max}]$ as the velocity magnitude range designated for the dynamic adjustment of σ. The parameter $\sigma_{\text{max}}^{T_i}$ denotes the maximum velocity tracking coefficient assigned to the i-th terrain type. Given a commanded velocity v for the i-th terrain, the intermediate coefficient σ_{vel} is formulated as follows:

$$\begin{cases} \sigma, & v \in [0, v_{min}), \\ \sigma(v - v_{min}) + \sigma_{\text{max}}^{T_i}(v_{max} - v), & v \in [v_{min}, v_{max}), \\ \sigma_{\text{max}}^{T_i}, & v \in [v_{max}, \infty). \end{cases} \tag{8}$$

The final adaptive tracking coefficient σ_{now} incorporates the terrain difficulty level L as delineated below:

$$\sigma_{\text{now}} = \sigma + \min(e^{\frac{L}{10}} - 1, 1)(\sigma_{\text{vel}} - \sigma) \tag{9}$$

The velocity commands v pertain to longitudinal and lateral linear velocities as well as angular velocity commands. Table IX in the Appendix details the maximum velocity tracking coefficients $\sigma_{\text{max}}^{T_i}$ and the associated velocity adjustment ranges across diverse terrains.

B. Command Design

Direct training with the full command range of $[-1, 1]$ m/s across all terrains enables rapid progression through difficulty levels but frequently yields unstable gaits. Specifically, the robot often demonstrates erratic behaviors such as leaping and high-frequency leg motions. Conversely, training from low-speed commands facilitates the acquisition of stable locomotion patterns. We therefore introduce a *command curriculum* to address these issues, as detailed in Table X.

We observed that when the maximum command magnitude exceeds $[-1, 1]$ m/s, the robot fails to accurately track the target linear velocity on complex terrains such as wave, stairs, and obstacle. This tracking discrepancy induces instability during the training process. Therefore, we impose specific constraints on the maximum command range for individual terrains as detailed in Table IX. Notably, although these limits are strictly enforced during the training phase, no such restrictions are applied during hardware testing on the physical robot. Despite this discrepancy, the model follows commands that lie beyond the training distribution and demonstrates robust generalization capabilities.

Our empirical analysis indicates that uniform sampling distributions are suboptimal because boundary values exhibit an exceptionally low probability of occurrence despite being frequently encountered during hardware deployment. To address this issue, we introduced an *extreme command sampling* strategy. This methodology allocates a 10% probability to stationary commands and a 20% probability to command combinations that represent maximum velocity limits across all three dimensions. Furthermore, when the linear velocity is zero, the framework maintains a 20% probability of sampling the maximum angular velocity to enhance robustness during pivot turns.

At the start of training, the linear velocity command range is restricted to $[-0.5, 0.5]$ m/s with a 10% probability of remaining stationary. Such a narrow distribution frequently produces command sequences that fail the terrain level-up condition, which necessitates a final horizontal distance relative to the initial position exceeding 4m, a value equivalent to half the terrain length [41]. This limitation prevents the agent from exploring higher difficulty levels. To guarantee that the cumulative command length surpasses the required threshold, we implement a *dynamic command sampling* strategy.

Let n_r represent the number of sampled commands and $\boldsymbol{v}_i^{\text{cmd}}$ denote the i-th linear velocity command. Given that T_r signifies the sampling interval and T_{ep} is the episode duration, the sampling range for the $(n_r + 1)$-th command is restricted to the intervals between $(v^{\text{min}}, -v^*) \cup (v^*, v^{\text{max}})$ where v^* is formulated as follows:

$$v^* := \text{clip}\left(\frac{5 - ||\sum_{i=1}^{n_r} \boldsymbol{v}_i^{\text{cmd}}||_2 T_r}{T_{\text{ep}} - n_r T_r}, 0, \min(|v^{\text{min}}|, |v^{\text{max}}|)\right) \tag{10}$$

Should a stationary command be selected for the (n_r+1)-th sample, its specific duration is determined as follows:

$$T^{zero} = \text{clip}\left(T_{\text{ep}} - n_r T_r - \frac{5 - ||\sum_{i=1}^{n_r} \boldsymbol{v}_i^{\text{cmd}}||_2 T_r}{0.8 \times \max(v_x^{\text{max}}, v_y^{\text{max}})}, 0, T_r\right) \tag{11}$$

The integration of the aforementioned command curriculum, extreme command sampling, and dynamic command sampling promotes the development of more stable locomotion gaits while ensuring a steady advancement across terrain difficulty levels. Additionally, these strategies markedly raise the performance ceiling for models evaluated with the RoboGauge.

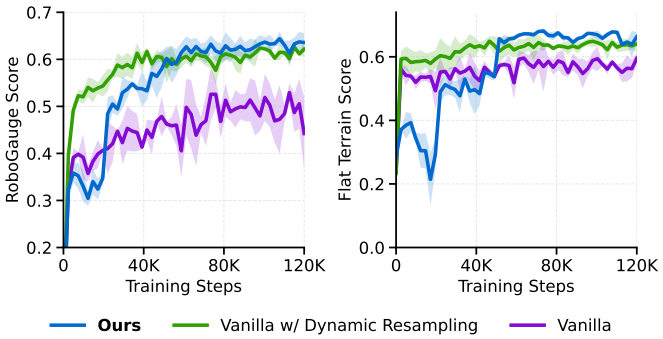


Fig. 13: Ablation study on training strategies.

We conducted ablation studies on the training configurations where Fig. 13 illustrates the impact of dynamic command

sampling. Activating this feature accelerates convergence and elevates the peak reward by 11% relative to the version without dynamic sampling. The final training curve is achieved by further incorporating dynamic velocity tracking precision adjustment and a command curriculum. These additions significantly bolster training stability and improve performance on flat terrain.

APPENDIX C
TRAIN CONFIGURATION

TABLE VIII: Reward Function Specifications

Reward Term	Equation	Weight
Lin. velocity tracking	$\exp(-\sigma\|\|\boldsymbol{v}_{xy}^{\text{cmd}} - \boldsymbol{v}_{xy}\|\|_2^2)$	$1.0/2.0$
Ang. velocity tracking	$\exp(-\sigma\|\omega_z^{\text{cmd}} - \omega_z\|^2)$	0.5
Lin. velocity (z)	v_z^2	-2.0
Ang. velocity (xy)	$\|\|\boldsymbol{\omega}_{xy}\|\|_2^2$	-0.05
Joint acceleration	$\ddot{q}^2$	-2.5×10^{-7}
Joint power	$\|\boldsymbol{\tau}\|\|\dot{q}\|^T$	-2×10^{-5}
Joint torque	$\|\|\boldsymbol{\tau}\|\|_2^2$	-1×10^{-4}
Base height	$(h^{\text{des}} - h)^2$	-1.0
Action rate	$\|\|\boldsymbol{a}_t - \boldsymbol{a}_{t-1}\|\|_2^2$	-0.01
Action smoothness	$\|\|\boldsymbol{a}_t - 2\boldsymbol{a}_{t-1} + \boldsymbol{a}_{t-2}\|\|_2^2$	-0.01
Collision	$n_{\text{collision}}$	-1.0
Joint limit	$n_{\text{limitation}}$	-2.0
Foot regulation	r^{fr}	-0.05
Hip regulation	$\|\boldsymbol{q}^{\text{hip}} - \boldsymbol{q}_{\text{default}}^{\text{hip}}\|$	-0.05
Hip symmetry	r^{hs}	-1

Black: Reward terms utilized for the multi-terrain model.
Red: Flat-ground high-speed model modified weights.

TABLE IX: Maximum Velocity Tracking Coefficients and Command Limits Across Terrains

Terrain Type	σ_{max}^i	v_x [m/s]	v_y [m/s]	ω_z [rad/s]
Flat	1/4	$\pm$ 2.0	$\pm$ 1.0	$\pm$ 2.0
Wave	5/12	$\pm$ 1.5	$\pm$ 1.0	$\pm$ 1.5
Slope	1/4	$\pm$ 1.5	$\pm$ 1.0	$\pm$ 1.5
Stairs Up	1/2	$\pm$ 1.0	$\pm$ 1.0	$\pm$ 1.5
Stairs Down	1/2	$\pm$ 1.0	$\pm$ 1.0	$\pm$ 1.5
Obstacle	3/4	$\pm$ 1.0	$\pm$ 1.0	$\pm$ 1.5

Note: Velocity ranges are defined as $v^{\text{lin}} \in [0.5, 1.5]$ m/s and $v^{\text{ang}} \in [1.0, 2.0]$ rad/s.

TABLE X: Command Curriculum Stages and Velocity Limits

Stage	**Training Steps**	v_x [m/s]	v_y [m/s]	ω_z [rad/s]
Initial	$[0, 2\times10^4]$	$\pm$ 0.5	$\pm$ 0.5	$\pm$ 1.0
Intermediate	$[2\times10^4, 5\times10^4]$	$\pm$ 1.0	$\pm$ 1.0	$\pm$ 1.5
Advanced	$[5\times10^4, \infty]$	$\pm$ 2.0	$\pm$ 1.0	$\pm$ 2.0

APPENDIX D
SUPPLEMENTARY EXPERIMENT

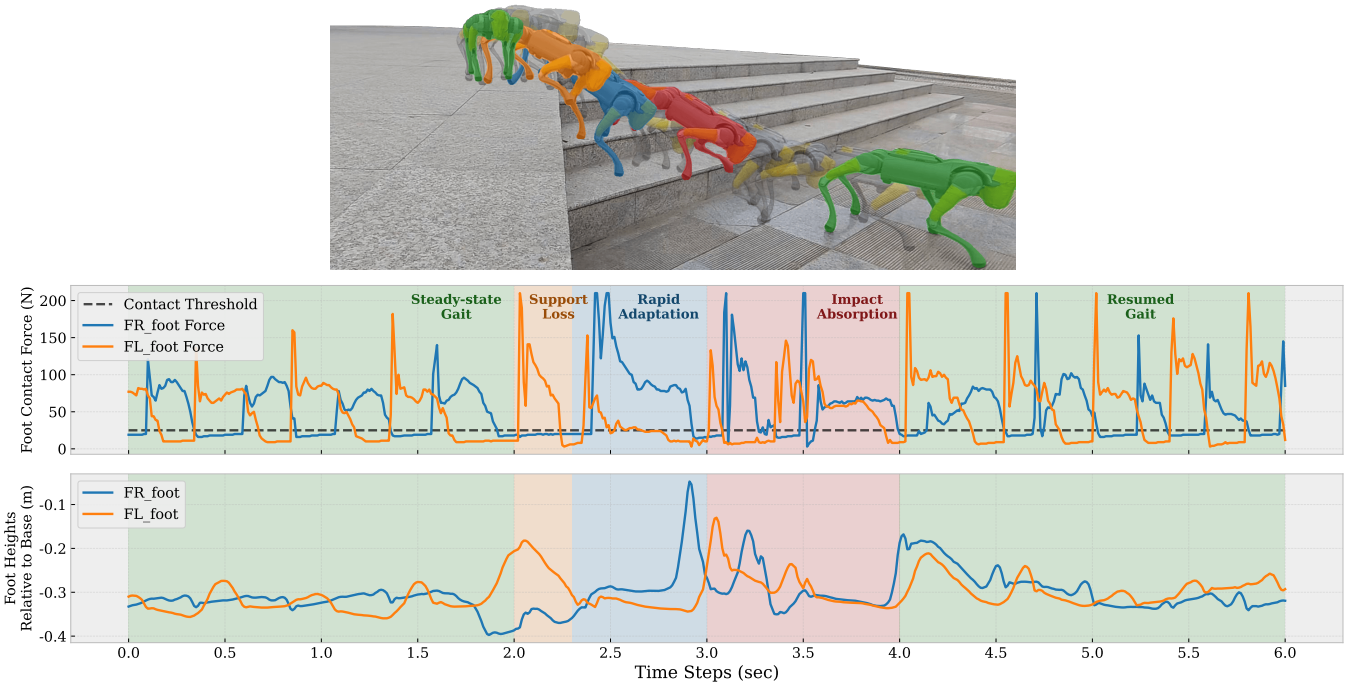


Fig. 14: The top panel shows the robot quickly adjusting its posture to safely descend when the ground ends at the edge. The middle plot depicts the contact force signals measured by the foot sensors. The bottom image illustrates the front foot height relative to the base calculated from forward kinematics. These results confirm the robustness of the policy and its capacity for adaptive gait transitions across diverse challenges.

TABLE XI: Comprehensive Evaluation: Real-World Measurements, Predicted Values, and Absolute Errors

Source	**Movement**	**Lin. Trk.**	**Ang. Trk.**	**DOF Power**	**DOF Limits**	**Orient.**	**Smooth.**
Real (Ground Truth)	Linear ($x = 1$)	0.9185	0.5808	0.8527	0.9159	0.9675	0.7739
	Lateral ($y = 0.5$)	0.9552	0.7037	0.9696	0.9384	0.9661	0.8985
	Angular ($z = 1$)	0.9552	0.8010	0.9659	0.9439	0.9614	0.9020
	Stairs ($x = 1$)	0.8554	0.2732	0.6721	0.8395	0.8749	0.5944
RoboGauge (Predicted)	Linear ($x = 1$)	0.8217	0.5669	0.8330	0.9386	0.9592	0.7853
	Lateral ($y = 0.5$)	0.8685	0.6822	0.9704	0.9293	0.9627	0.8861
	Angular ($z = 1$)	0.8763	0.7679	0.9734	0.9414	0.9647	0.8983
	Stairs ($x = 1$)	0.7596	0.2507	0.6211	0.8472	0.8625	0.6606
RoboGauge (Ours) (Abs. Error) ↓	$x = 1$ (Merge)	0.0963	0.0182	0.0354	0.0152	0.0103	0.0388
	Average	0.0873	**0.0243**	**0.0145**	**0.0089**	**0.0057**	**0.0183**
IsaacGym (Predicted)	Linear ($x = 1$)	0.8977	0.7826	0.9155	0.9361	0.9737	0.8289
	Lateral ($y = 0.5$)	0.9694	0.8039	0.9598	0.9378	0.9707	0.8781
	Angular ($z = 1$)	0.9853	0.9134	0.9751	0.9325	0.9798	0.9510
	Stairs ($x = 1$)	0.8786	0.5732	0.8635	0.9027	0.9339	0.7454
IsaacGym (Abs. Error) ↓	$x = 1$ (Merge)	0.0220	0.2509	0.1271	0.0417	0.0326	0.1030
	Average	**0.0221**	0.1545	0.0487	0.0179	0.0185	0.0575

TABLE XII: RoboGauge detailed metrics for baselines

Model	ang vel err			lin vel err			dof limits		
	mean	mean@25	mean@50	mean	mean@25	mean@50	mean	mean@25	mean@50
Our	**0.66**	**0.6241**	**0.6402**	**0.6919**	**0.6561**	**0.6746**	**0.8057**	**0.8006**	**0.803**
CTS	0.5777	0.5369	0.5554	0.6005	0.5527	0.5774	0.7269	0.7218	0.7241
HIM	0.5098	0.4644	0.4857	0.5551	0.5148	0.5355	0.6574	0.6517	0.6538
DreamWaQ	0.4522	0.4092	0.4288	0.4954	0.4548	0.4755	0.5953	0.5899	0.5919

Model	dof power			orientation stability			torque smoothness		
	mean	mean@25	mean@50	mean	mean@25	mean@50	mean	mean@25	mean@50
Our	**0.7633**	**0.7382**	**0.75**	**0.8036**	**0.7986**	**0.8009**	**0.7546**	**0.736**	**0.7435**
CTS	0.6857	0.6604	0.6723	0.7237	0.7165	0.72	0.6777	0.6585	0.6666
HIM	0.6201	0.6008	0.6095	0.6541	0.6439	0.6493	0.6135	0.5975	0.6038
DreamWaQ	0.5611	0.5426	0.5512	0.5929	0.5835	0.5884	0.5508	0.5354	0.5415

TABLE XIII: RoboGauge detailed terrain scores for baselines

Model	flat			wave			slope forward			slope backward		
	mean	mean@25	mean@50	mean	mean@25	mean@50	mean	mean@25	mean@50	mean	mean@25	mean@50
Our	**0.7851**	**0.586**	**0.6885**	**0.6344**	**0.5569**	**0.5961**	**0.5896**	**0.5175**	**0.5536**	0.5916	0.5205	0.5562
CTS	0.7454	0.4846	0.6211	0.6183	0.5389	0.5765	0.5629	0.4905	0.5227	0.5377	0.4655	0.5001
HIM	0.7719	0.523	0.6536	0.579	0.4975	0.5338	0.5087	0.4334	0.4689	0.5842	0.5086	0.547
DreamWaQ	0.7706	0.528	0.6547	0.5401	0.4629	0.497	0.408	0.3421	0.3734	**0.6081**	**0.5411**	**0.5753**

Model	stairs forward			stairs backward			obstacle		
	mean	mean@25	mean@50	mean	mean@25	mean@50	mean	mean@25	mean@50
Our	0.8102	0.7246	**0.7578**	**0.7958**	**0.7103**	**0.7472**	**0.8737**	**0.7858**	**0.8219**
CTS	**0.8106**	**0.7309**	0.7561	0.6185	0.5373	0.568	0.5785	0.4907	0.5289
HIM	0.4654	0.38	0.4125	0.7972	0.7153	0.7475	0.3646	0.2876	0.3227
DreamWaQ	0.3742	0.2991	0.3274	0.7419	0.6608	0.6916	0.206	0.1243	0.1645

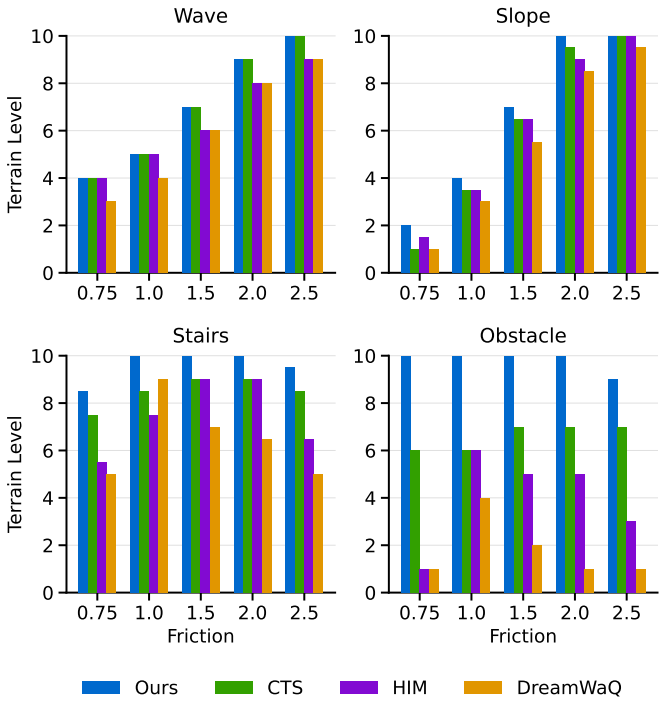
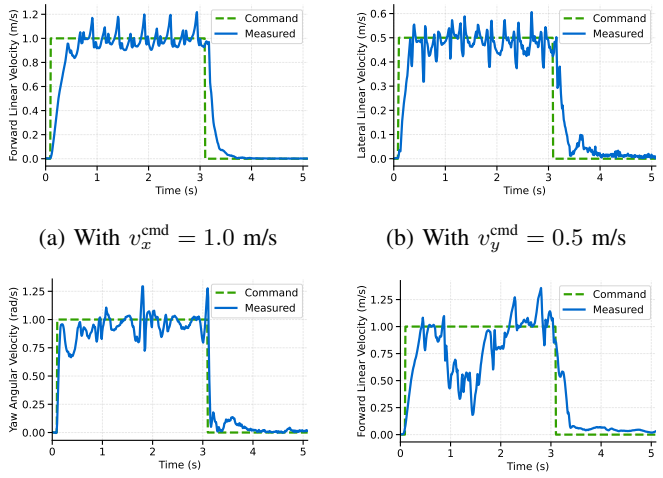


Fig. 15: Maximum terrain difficulty levels achieved by various models under a subset of friction coefficients (ranging from 0.5 to 2.5 in increments of 0.25).



(a) With $v_x^{\text{cmd}} = 1.0$ m/s

(b) With $v_y^{\text{cmd}} = 0.5$ m/s

(c) With $\omega_z^{\text{cmd}} = 1.0$ rad/s

(d) 10 cm stairs with $v_x = 1.0$ m/s

Fig. 16: The green dashed lines represent the ground-truth velocities measured by the motion capture system at a sampling frequency of 90 Hz, while the blue solid lines denote the corresponding target command values automatically transmitted to the Unitree Go2 via a pre-defined evaluation program.

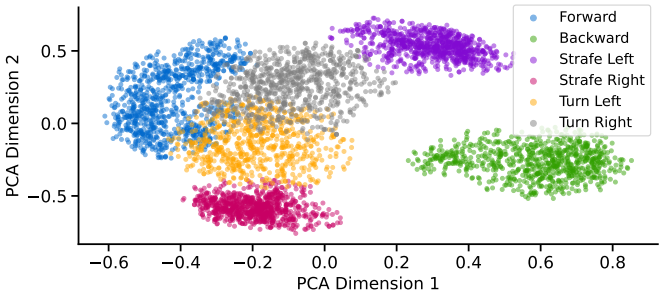


Fig. 17: PCA visualization of the student encoder latent space in different commands with all terrains.

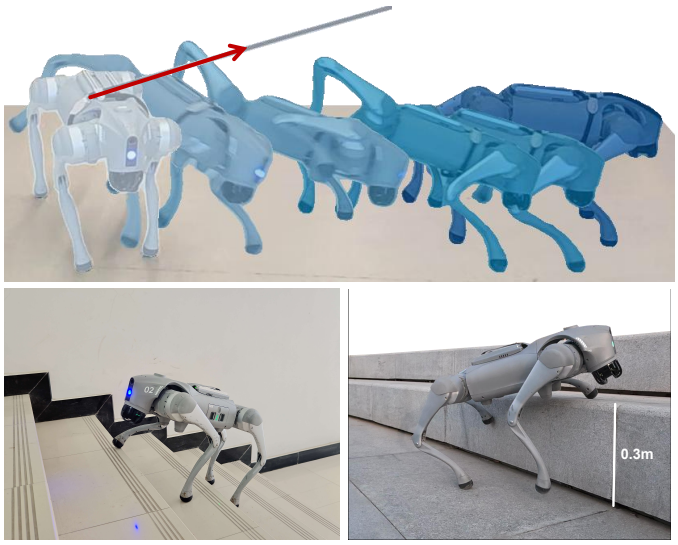


Fig. 18: Locomotion performance of the Unitree Go2 across three challenging scenarios. Top image illustrates the robot maintaining balance against a lateral impulse between 80 N and 100 N. Bottom-left image depicts the stable ascent of 15.5 cm tile stairs with $\mu = 0.38$. Bottom-right image showcases the successful traversal of a 30 cm obstacle where $\mu = 0.85$.