2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

Learning Task-Invariant Properties via Dreamer: Enabling Efficient Policy Transfer for Quadruped Robots

Junyang Liang[1], Yuxuan Liu[1], Yabin Chang[1], Junfan Lin[2], Junkai Ji[1], Hui Li[1], Changxin Huang[1*], Jianqiang Li[1]

***Abstract*— Achieving quadruped robot locomotion across diverse and dynamic terrains presents significant challenges, primarily due to the discrepancies between simulation environments and real-world conditions. Traditional sim-to-real transfer methods often rely on manual feature design or costly real-world fine-tuning. To address these limitations, this paper proposes the DreamTIP framework, which incorporates Task-Invariant Properties learning within the Dreamer world model architecture to enhance sim-to-real transfer capabilities. Guided by large language models, DreamTIP identifies and leverages Task-Invariant Properties, such as contact stability and terrain clearance, which exhibit robustness to dynamic variations and strong transferability across tasks. These properties are integrated into the world model as auxiliary prediction targets, enabling the policy to learn representations that are insensitive to underlying dynamic changes. Furthermore, an efficient adaptation strategy is designed, employing a mixed replay buffer and regularization constraints to rapidly calibrate to real-world dynamics while effectively mitigating representation collapse and catastrophic forgetting. Extensive experiments on complex terrains, including Stair, Climb, Tilt, and Crawl, demonstrate that DreamTIP significantly outperforms state-of-the-art baselines in both simulated and real-world environments. Our method achieves an average performance improvement of 28.1% across eight distinct simulated transfer tasks. In the real-world Climb task, the baseline method achieved only a 10% success rate, whereas our method attained a 100% success rate. These results indicate that incorporating Task-Invariant Properties into Dreamer learning offers a novel solution for achieving robust and transferable robot locomotion.**

I. INTRODUCTION

Enabling robots to autonomously operate in complex real-world environments remains a core goal of embodied intelligence [1]. However, real-world training faces challenges such as low data efficiency, high costs, and safety risks [2], [3]. Training in simulation offers a low-cost, scalable alternative [4] , yet dynamics discrepancies, sensor noise, and visual mismatches often cause performance drops when transferring policies to reality [5], [6]. Thus, achieving efficient and stable sim-to-real transfer while maintaining generalization in unseen environments is the key challenge [7]–[9].

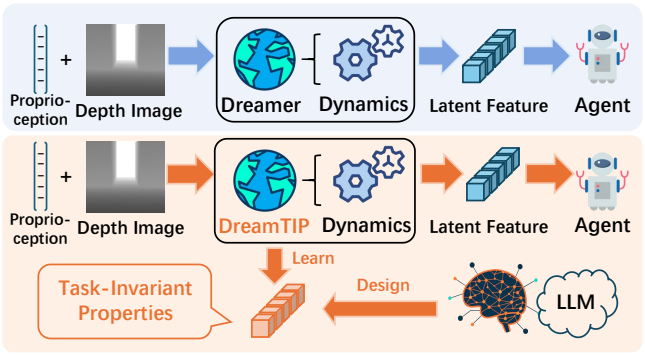

Fig. 1: Different Dreamer learning paradigms. The original Dreamer learns environment dynamics by reconstructing observations. DreamTIP, building upon this, also incorporates Task-Invariant Properties designed by an LLM to reduce its over-reliance on underlying dynamics parameters.

Research on Sim-to-Real transfer focuses on three main approaches: domain randomization, domain adaptation, and simulator enhancement [4]. Domain randomization [6], [10] improves robustness by adding parametric randomness during simulation training, but its predefined distributions often fail to cover real-world complexity, limiting generalization. Simulator enhancement [11] builds high-fidelity simulations to narrow the reality gap, yet it requires accurate modeling, is costly, and struggles with complex dynamics. Domain adaptation [12], [13] aligns feature distributions across domains to reduce discrepancies, but it often faces training instability and high computational costs.

Domain adaptation methods focus on learning domain-invariant features to maintain consistent policy performance across environments [14]–[16]. They incorporate representation constraints to reduce dynamics discrepancies between simulation and reality at the feature level. For example, Lai et al. [17] proposed World Model Perception (WMP), which uses a world model to extract compact dynamic representations from historical visual and proprioceptive data for efficient policy learning. Gu et al. introduced Denoised World Model (DWL) [18], employing privileged information as auxiliary supervision in an encoder-decoder structure to improve dynamics modeling and enhance robotic locomotion generalization in complex terrains.

Despite this, policies from existing methods heavily rely

*Corresponding Author: Changxin Huang (huangchx@szu.edu.cn)
[1]Shenzhen University, Shenzhen, China
[2]Peng Cheng Laboratory, Shenzhen, China

This work is supported in part by the National Natural Science Foundation of China (No. 62403325, No. 62325307, No. 62527809, No. 62203134, No. 62373258, No. 62506180), in part by the Natural Science Foundation of Guangdong Province (No. 2023B1515120038, No. 2026A1515011532), in part by Shenzhen Science and Technology Innovation Commission (No. 20231122104038002, No. KJZD20230923113801004, No. JCYJ20240813141628038, No. KJZD20230923115215032), in part by the Shenzhen Key Industry R&D Program Project (No. ZDCY20250901102300001), in part by China Postdoctoral Science Foundation (No. 2025M771522), in part by the Major Key Project of PCL (No. PCL2024A04, No. PCL2025A17). This work is also supported by the Intelligent Computing Center of Shenzhen University.

on the specific dynamic parameters configured in the simulation. This dependency results in fragile performance when facing real-world dynamic variations. To address this, we propose a world model-based policy transfer framework aimed at reducing the policy's reliance on dynamic characteristics, enabling efficient transfer with minimal real-world samples for fine-tuning the world model.

Specifically, this work builds upon the Dreamer framework [19] to construct a world model that learns latent features of the robot dynamics and integrates them into the state space, thereby supporting subsequent reinforcement learning (RL) policy training. To further enhance the policy's generalization and adaptation capabilities in the face of environmental variations, we propose the DreamTIP framework: Learning Task-Invariant Properties via Dreamer (**DreamTIP**). This approach introduces a world model learning method during Dreamer training that guides the agent to acquire Task-Invariant Properties, which are both transferable across tasks and robust to dynamic changes, thereby reducing the policy's reliance on specific dynamic parameters. Taking legged robot locomotion as an example, such properties can manifest as contact-related stability, terrain clearance, and other dynamics-robust attributes that generalize across diverse tasks.

However, a critical and underexplored question remains: how to accurately define and acquire Task-Invariant Properties. Manual design of task-specific intermediate features [20], [21] is costly, requires deep expertise, and is prone to bias, which limits cross-task generalization. In contrast, Large Language Models (LLMs) leverage their vast pre-trained knowledge to reason about and abstract high-dimensional task descriptions and state observations, uncovering fundamental physical and behavioral principles crucial for task success that human experts might overlook. To overcome this, we employ LLMs to build a **Task-Invariant Extractor**. Leveraging their physical and behavioral knowledge, LLMs identify high-level task semantics and extract Task-Invariant Properties from privileged observations. These properties serve as auxiliary prediction targets in the world model, enhancing the robustness of latent representations against sim-to-real physical discrepancies.

Even though Task-Invariant Properties improve robustness, sim-to-real gaps remain. Fine-tuning with limited real data is often needed to calibrate model parameters toward true dynamics and ensure policy performance [22]. However, this process risks representation collapse and catastrophic forgetting [23]. To overcome these issues, we propose an efficient adaptation method for rapid real-world deployment of DreamTIP. Specifically, our method constructs a mixed replay buffer with both simulated and real data to reduce representation collapse from distribution gaps. During adaptation, we duplicate and freeze a pre-trained DreamTIP model as a reference. Updates to the posterior state are constrained by minimizing negative cosine similarity, a metric insensitive to variations in feature scale, thereby enhancing adaptation stability. Additionally, we freeze DreamTIP's recurrent module during fine-tuning to accelerate adaptation to real-world dynamics.

We compared our method with baselines on complex terrains such as stairs, jumps, and scrambles. In simulation, our approach outperformed baselines in nearly all transfer tasks. For instance, in the challenging Crawl task (23 *cm*), our method achieved an average reward of 25.35, far exceeding the baseline's 5.66. Real-world tests on the Unitree Go2 robot further validated our method: it reached a 100% success rate in the Climb task (53 *cm*), compared to the baseline's 10%. These results demonstrate that extracting Task-Invariant Properties and enabling efficient adaptation significantly improves policy generalization and bridge the sim-to-real gap.

II. RELATED WORK

A. Sim to real transfer

Sim-to-real transfer tackles performance decline when simulation-trained policies face real-world deployment [7]–[9]. The key challenge lies in bridging visual and dynamic gaps between domains. Main approaches include: domain randomization [6], [10], improving generalization through varied simulation parameters; domain adaptation [12], aligning feature distributions for zero-shot transfer or efficient fine-tuning; and simulator enhancement [11], building high-fidelity environments. This paper introduces a world model to reduce dynamics-specific dependency, enabling effective adaptation with minimal real-world data.

B. World model for robotics

World models significantly reduce the reliance on real-world interaction data by modeling environmental dynamics to support prediction and planning [24]. For instance, DreamerV3 [19] achieves efficient and stable training across multiple tasks through latent dynamics prediction and multi-scale optimization; DayDreamer [25] enables robots to acquire complex behaviors and achieve online adaptation with limited real-world interactions; and the WMP method proposed by Lai et al. [17] extracts compact representations from multimodal perception to enhance policy learning efficiency. Building on these advances, we further explore the role of world models in cross-domain generalization and propose explicitly learning Task-Invariant Properties to enhance the model's robustness to dynamic variations, thereby better facilitating simulation-to-real transfer.

C. LLM-driven robot skill learning

Large language models are applied in quadruped robots across three main areas: reward modeling, motion control, and representation learning. In reward design, works like Eureka [26] show LLMs can automatically generate reward functions. For motion control, some studies use LLMs to convert language into intermediate commands (e.g., foot contact patterns) executed by reinforcement learning controllers [27], [28], or even directly output joint trajectories [29]. In representation learning, methods such as LESR [30] employ LLMs to improve state representations and intrinsic rewards, enhancing policy generalization and efficiency. In contrast,

our approach utilizes LLMs' rich knowledge and reasoning capabilities to extract Task-Invariant Properties closely tied to task success, ultimately enhancing the generalization ability of robot policies.

III. BACKGROUND

A. Problem Formulation

The learning of locomotion skills for legged robots can be formulated as a Partially Observable Markov Decision Process (POMDP), represented by the tuple $(\mathcal{S}, \mathcal{O}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma)$, where $s_t \in \mathcal{S}$ is the state space, which encapsulates the full dynamical state of the robot and its environment. $o_t \in \mathcal{O}$ is the observation, $a_t \in \mathcal{A}$ is the action space, $\mathcal{P}$ is the state transition function, $\mathcal{R}$ is the reward function, and $\gamma \in (0, 1)$ is the discount factor. The goal of the agent is to learn a policy that maximizes its cumulative discounted return $R_t = \sum_{i=0}^{\infty} \gamma^i r_{t+i}$.

B. Dreamer-Augmented Policy Optimization

Dreamer [19] learns a latent dynamics model to extract abstract representations of environmental dynamics from pixel or state observations. In this work, we employ a variant of Dreamer whose core component is the Recurrent State-Space Model (RSSM), which primarily consists of the following four parts: Recurrent model $h_t = f_\theta(h_{t-1}, z_{t-1}, a_{t-1})$, Encoder $z_t \sim q_\theta(z_t \mid h_t, o_t)$, Dynamic predictor $\hat{z}_t \sim p_\theta(\hat{z}_t \mid h_t)$ and Decoder $\hat{o}_t \sim p_\theta(\hat{o}_t \mid h_t, z_t)$.

Dreamer jointly learns the entire model by minimizing the negative Evidence Lower Bound (ELBO):

$$\begin{aligned}\mathcal{L}_D(\theta) \doteq \mathbb{E}_{q_\theta}\Big[\sum_{t=1}^{T}\big(-\ln p_\theta(o_t \mid h_t, z_t) \\ + \beta\,\mathrm{KL}\big[q_\theta(z_t \mid h_t, o_t) \parallel p_\theta(\hat{z}_t \mid h_t)\big]\big)\Big],\end{aligned} \tag{1}$$

where β is a hyperparameter.

PPO [31] is a policy optimization algorithm based on the Actor-Critic framework, which aims to learn a policy $\pi_\theta(a_t \mid o_t)$ that maximizes cumulative returns by optimizing policy gradients. Inspired by WMP [17], we first input the observation o_{t-1}, which consists of proprioceptive data and depth images, into Dreamer. And then the hidden state h_t is computed by the recurrent model (RNN) based on the previous deterministic state h_{t-1}, stochastic state s_{t-1}, and action a_{t-1}. It encodes the deterministic history of the environment's dynamics, capturing the complete temporal evolution from the initial state to the current time step. Subsequently, incorporate the h_t along with the current observation o_t (without depth image) as inputs to the Actor-Critic network. Consequently, the optimization objective of PPO is redefined as learning a policy $\pi_\theta(a_t \mid h_t, o_t)$ that maximizes cumulative returns.

IV. METHOD

A. Overview

In this section, we present the proposed framework, termed Learning Task-Invariant Properties via Dreamer. As shwon in Fig. 2, this framework first leverages a large language model to analyze task descriptions and state observation spaces, constructing a **TIP Extractor** to convert privileged states into corresponding **Task-Invariant Properties**. Subsequently, we introduce an additional predictor on the Dreamer architecture, developing an improved version termed **DreamTIP**, to explicitly learn these Task-Invariant Properties. During the deployment phase, the framework duplicates and freezes the pre-trained DreamTIP parameters to serve as a reference model. During the adaptation process, a mixed replay buffer comprising both real and simulated data is utilized for offline updates, while regularization constraints are applied to the adaptation process. This approach enables efficient adaptation with minimal real-world data requirements.

B. Task-Invariant Properties

Manually designing these properties entails high costs and inherent limitations. To overcome these constraints, this work leverages the prior knowledge and reasoning capabilities of LLMs to construct a **TIP Extractor**. This module converts raw privileged observational information into **Task-Invariant Properties** that are closely correlated with task success. Specifically, as illustrated in Fig. 2, prior to training, we provide the high-level task description I_{text} and the complete state observation space I_{priv} containing privileged information as inputs to guide the LLM in reasoning and outputting a properties transformation function $TIP_{extrator}$. This function $TIP_{extrator}$ is capable of extracting Task-Invariant Properties f_t from the raw privileged observations s_t, which are both generalizable across tasks and insensitive to variations in dynamics. The process is formally defined as:

$$TIP_{extrator} = \mathrm{LLM}(I_{text}, I_{priv}), \tag{2}$$

$$f_t = TIP_{extrator}(s_t). \tag{3}$$

Building upon this foundation, and to integrate the extracted Task-Invariant Properties f_t into the world model's learning process while enhancing its transfer capabilities, we propose DreamTIP, an improved version of the Dreamer framework [19]. The core innovation of this architecture lies in the introduction of a properties predictor, designed to enable the dreamer to infer the same Task-Invariant Properties f_t from its own latent states. This predictor, implemented as a Multilayer Perceptron (MLP), takes as input the concatenation of DreamTIP's recurrent state ht and stochastic state representation z_t at time step t, and outputs an estimate $\hat{f}_t$ of the current Task-Invariant Properties, then Eq. 1 can be rewritten as:

$$\mathcal{L}_{train}(\theta) \doteq \mathcal{L}_D(\theta) - \mathbb{E}_{q_\theta}\Big[\sum_{t=1}^{T}\ln p_\theta(f_t \mid h_t, z_t)\Big], \tag{4}$$

where the second term is L_{MLE}.

By incorporating Task-Invariant Properties, this approach encourages the world model to learn representations that are both generalizable across tasks and robust to dynamic disturbances, thereby enhancing its transfer capability and

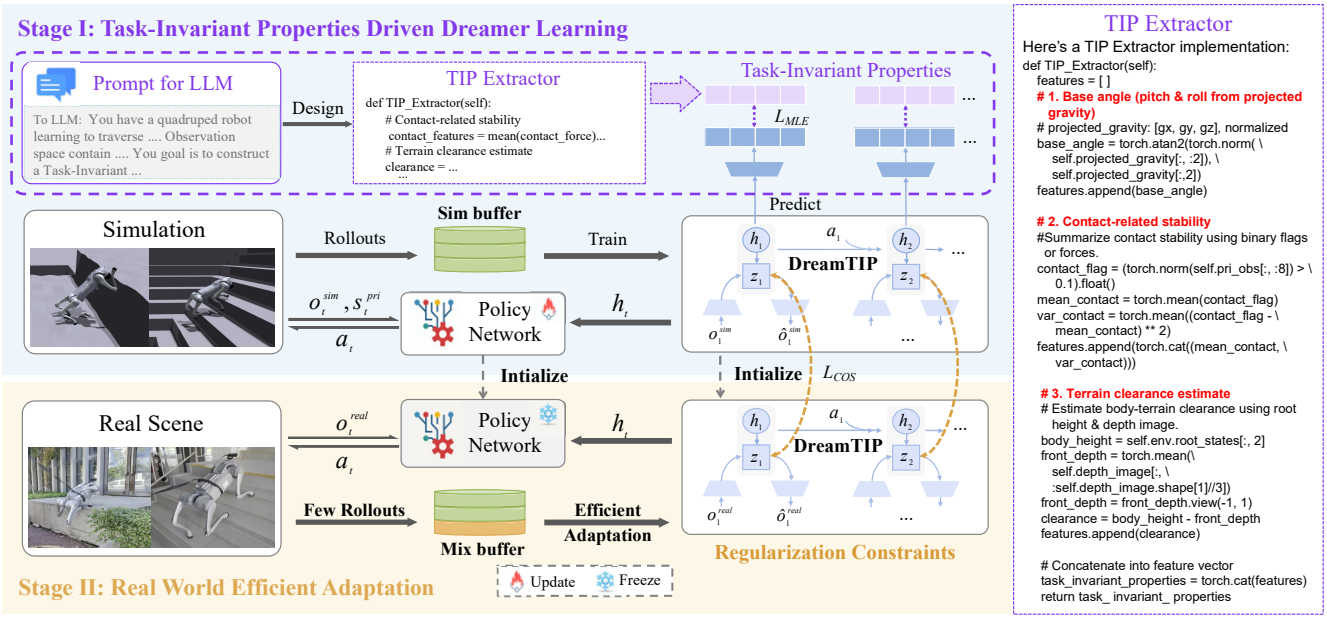

Fig. 2: Overview of the proposed framework. The framework consists of two stages: In the first stage, DreamTIP is employed in a simulation environment to learn Task-Invariant Properties; In the second stage, it adapts to the dynamics distribution in the physical environment with only a few rollouts.

behavioral consistency in previously unknown real-world environments.

C. Real-World Efficient Adaptation

During the simulation training phase of DreamTIP, we concurrently collect simulated trajectories to construct a simulated experience replay buffer (**Sim buffer**). Upon model convergence, its parameters are duplicated and frozen to obtain a fixed DreamTIP model M_{sg}, which is deployed on the quadruped robot alongside a frozen policy model π. Subsequently, through interaction in the real environment, real trajectory data is continuously collected and incrementally merged into the original sim buffer to form a mixed replay buffer (**Mix buffer**). This mix buffer is utilized for subsequent offline adaptation updates in DreamTIP. The hybrid buffer mechanism is designed to balance the distribution of old and new data, thereby constraining the magnitude of model updates. This approach avoids over-optimization on limited real samples, which could otherwise disrupt or cause the forgetting of previously learned dynamic transition representations. Thereby, it mitigates issues such as catastrophic forgetting, representation collapse, and overfitting, and consistently enhances the model's adaptability to real-world dynamics.

Although the world model demonstrates high data efficiency during adaptation, its fine-tuning performance is still highly dependent on the quality and scale of real-world data, while real-world data collection is costly and sample availability is typically restricted. Furthermore, full parameter fine-tuning under limited data conditions often results in suboptimal outcomes. To enable rapid adaptation of DreamTIP to real dynamics using a small number of real trajectories, we freeze the Recurrent Model module within DreamTIP during the adaptation update process. This strategy is primarily motivated by the need to accelerate the world model's alignment with the distribution of real data, thus promoting faster adaptation to real-world dynamics.

To further enhance the stability of adaptation training, inspired by the teacher–student distillation framework in TWIST [32] and the source model supervision strategy in LS-UNN [33], we propose a Regularization Constraints-based stable adaptation method. The key idea is to stabilize model adaptation by introducing regularization constraints. Specifically, we first obtain a pre-trained DreamTIP model M from simulation, then duplicate and freeze its parameters to construct a reference world model M_{sg}. At each timestep t, the stochastic state representation of M_{sg} serves as a supervisory signal to guide the adaptation of M on real data.

Concretely, during the adaptation process, for each observation o_t at timestep t, both the frozen reference model M_{sg} and the adaptable model M encode it into stochastic state representations, denoted as z_t^{sg} and z_t respectively. A negative cosine similarity loss is minimized to align their directions, enforcing semantic consistency while remaining insensitive to feature scales [34]. This regularization term plays a critical role in complementing the reconstruction loss. While the reconstruction objective encourages the world model to fit real-world data distributions, limited data availability may drive the learned representation towards collapseor cause it to deviate from the well-structured latent space established during pre-training. The regularization constraint addresses this issue by stabilizing the adaptation process and preserving the quality of the latent representations.

According to Eq. 1, the complete loss function for the adaptation process can be refined as:

$$\mathcal{L}_{Adapt}(\theta) \doteq \mathcal{L}_D(\theta) - \mathbb{E}_{q_\theta}\Big[\sum_{t=1}^{T} \frac{z_t \cdot z_t^{sg}}{\|z_t\| \cdot \|z_t^{sg}\|}\Big]. \quad (5)$$

where the second term is L_{COS}.

During adaptation, the policy network π remains frozen. With this mechanism, our approach achieves stable and efficient world model adaptation using only a small amount of real-world data.

V. EXPERIMENTS

A. Experiment Setting

To validate the effectiveness of the proposed method in transfer tasks, experiments were conducted in both simulated and real-world environments. Fig. 4 illustrates the configuration of terrain tasks employed in our study, covering both simulation and real-world evaluation. The simulation experiments were built on the Isaac Gym environment. We employed Unitree Go2 robots for the experiments, whose action space is 12-dimensional, corresponding to the target positions of the 12 joints. The observation space includes proprioceptive information such as base angular velocity, direction of gravity projection, joint positions and velocities, along with depth images. Beyond the observation variables mentioned above, the privileged information also incorporates physical states such as linear velocity, elevation maps, friction coefficients, center of mass position, and foot contact forces. In real-world evaluation, all methods were deployed and executed directly on the onboard Orin Nano of the Go2 robot, utilizing depth images captured by the D435i camera, which were preprocessed with spatial and temporal filters to mitigate the visual sim-to-real gap [35].

TIP Extractor: In quadruped robot locomotion tasks mentioned in this paper, the TIP generated by the LLM, as shown in the right part of Fig. 2, reveal that tasks such as climb, stairs, and gaps share below critical common constraints: maintaining sufficient terrain clearance to avoid physical collisions, and preserving foot contact stability to prevent slipping and instability.

Reward functions: We adopt a reward function similar to that of Cheng et al. [6], which encourages the robot to follow the commanded velocities while penalizing velocities along other axes, excessive joint torques, accelerations, and collisions. In addition, we introduce two extra reward terms: penalizing joint deviations from the normal standing posture, and encouraging smoothness of joint torques [36]. We find that these designs are beneficial for sim-to-real transfer.

Simulation settings: A total of eight transfer tasks were constructed in the simulation environment for comprehensive evaluation. During training, the center of mass was randomized within [-0.05 *m*, 0.05 *m*], and the velocity command varied over [0, 1] *m/s*. Across the five terrain transfer tasks (stair, gap, climb, crawl, and tilt), each task is evaluated under difficulty levels not encountered during training. To further evaluate the robot's adaptation capability under unseen variations in mass distribution and velocity commands, the CoM Transfer and Velocity Transfer tasks were conducted on rough flat terrain. Specifically, in the CoM Transfer task, a mass-center offset Δa was introduced, resulting in an adjusted range: [-0.05-Δa, 0.05-Δa]∪[-0.05+Δa, 0.05+Δa]. In the Compound Task, we set a mass center offset (Δa = 0.1 *m*) and a fixed velocity command (1.1 *m/s*) for all four tasks: Gap (75 *cm*), Climb (40 *cm*), Crawl (25 *cm*), and Stair (18 *cm*). This configuration induced a more challenging scenario.

Methods	Stair (16*cm*)	Climb (52*cm*)	Tilt (33*cm*)	Crawl (25*cm*)
WMP	100%	10%	40%	70%
Ours w/o Adapt	100%	90%	50%	80%
Ours	**100%**	**100%**	**80%**	**100%**

TABLE I: Real-world evaluation. The success rate was employed as the evaluation metric in this study. The results were statistically derived from 10 independent trials conducted for each task.

B. Simulation Evaluation

The methods involved in the experiments conducted in this paper are as follows. WMP [17] and DreamTIP-DWL [18] are the two baseline methods. The specific descriptions are as follows:

WMP: Following the training paradigm proposed by Lai et al. [17], this method differs from DreamTIP by omitting both Task-Invariant Properties and adaptation updates.

DreamTIP-DWL: According to the training framework introduced by Gu et al. [18], DreamTIP predicts privileged information during DreamTIP training instead of Task-Invariant Properties, and performs no adaptation updates.

WMP w/ Finetune: Based on WMP, this variant utilizes adaptation updates. The sequence model is frozen during adaptation, and no regularization constraints are applied.

Ours w/o TIP: The proposed method without learning Task-Invariant Properties during the training phase.

Ours w/o Adapt: The proposed method without DreamTIP Adaptation during the transfer process.

Ours: The proposed method.

As illustrated in Fig. 3, the performance of the aforementioned methods across eight transfer tasks is evaluated. Our method achieves an average performance improvement of 28.1% across eight distinct simulated transfer tasks. The figure shows that performance differences are small under low task difficulty but become pronounced as difficulty increases, demanding greater adaptability. Our method proves most robust, with the least performance degradation as tasks grow harder. It consistently outperforms all baseline methods in nearly every task. For example, in the Crawl task with gap widths varying from 26 *cm* to 23 *cm*, the WMP method achieves an average reward of approximately 33.51 at the easiest level, which sharply decreases to 5.66 at the highest difficulty level, corresponding to a performance drop of about 83.1%. In contrast, our method declines from 36.58 to 25.35, resulting in a significantly more moderate reduction of only 30.6%, even under conditions where the baseline approach almost completely fails.

On tasks such as Stair, Gap, Climb, and Compound Transfer, the proposed method without Task-Invariant Properties (**Ours w/o TIP**) consistently outperforms the Weighted Model Predictive Control with Finetuning (**WMP w/ Finetune**) approach, which lacks explicit regularization constraints. Notably, the latter exhibits even worse performance on gap and stair tasks within the Compound Transfer setting

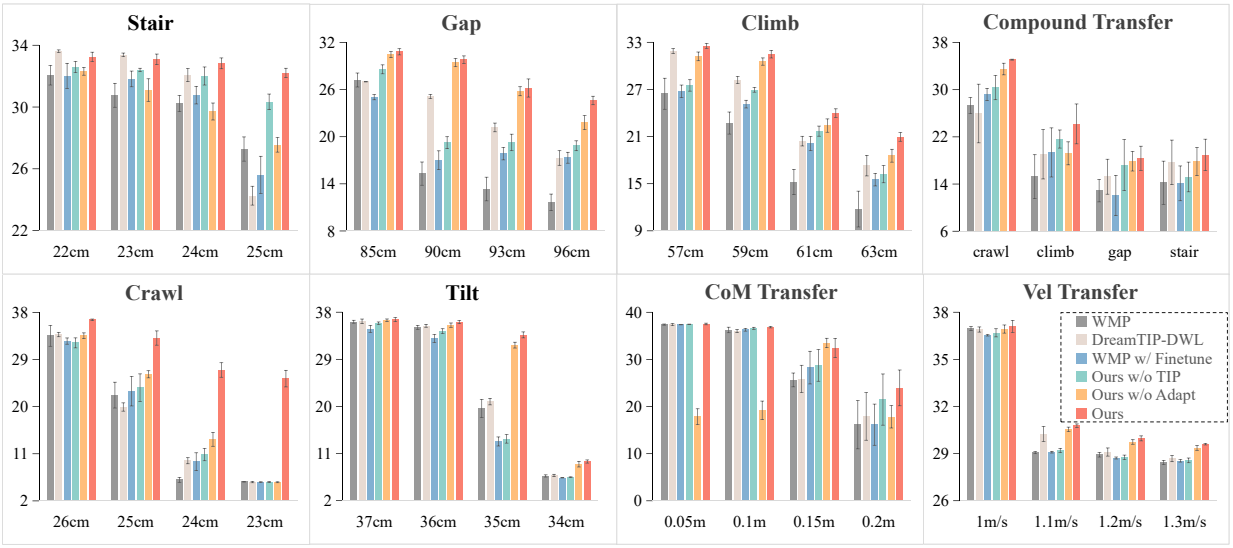

Fig. 3: Performance comparison of various methods on eight transfer tasks in simulation. The evaluation metric is the average cumulative reward over 100 trajectories per task. Our method outperformed all other baselines across the board. The vertical axis represents the average trajectory reward, while the horizontal axis indicates the varying levels of task difficulty. The results are obtained through testing over 100 trajectories with 3 different random seeds.

Methods	Climb				Tilt			
-	57cm	59cm	61cm	63cm	37cm	36cm	35cm	34cm
DreamTIP-DWL	31.89 ± 0.61	28.20 ± 0.90	20.42 ± 1.23	17.30 ± 2.62	36.06 ± 0.87	35.40 ± 0.55	20.92 ± 1.24	6.82 ± 0.43
DreamTIP-DeepSeekV3	**32.15 ± 0.21**	**31.06 ± 0.55**	20.74 ± 1.52	17.89 ± 1.66	36.13 ± 0.28	35.46 ± 0.13	25.79 ± 1.29	7.57 ± 0.88
DreamTIP-GPT5	31.21 ± 1.11	30.55 ± 0.93	**22.40 ± 1.70**	**18.56 ± 1.64**	**36.47 ± 0.48**	**35.53 ± 0.83**	**31.71 ± 1.11**	**8.97 ± 1.07**

TABLE II: Performance comparison of different Task-Invariant Properties design methods in simulation. **Bolded** numbers indicate the best performance.

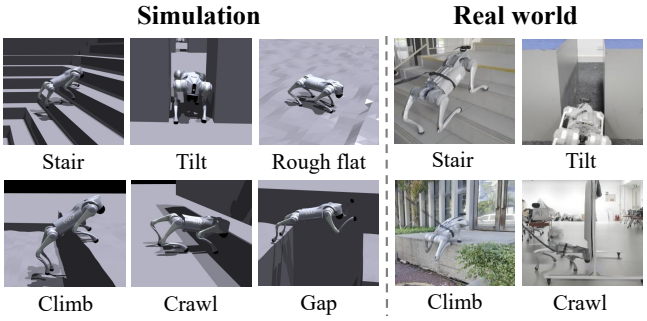

Fig. 4: Illustrations of terrain settings in simulation and real-world evaluation.

compared to its pre-finetuned version. These results suggest that the integration of regularization constraints effectively mitigates issues such as representation drift and knowledge forgetting during the adaptation process, thereby enabling more robust and stable performance across tasks.

Furthermore, comparisons between the baseline (**WMP**) and our method without adaptation mechanisms (**Ours w/o Adapt**) reveal that learning with Task-Invariant Properties significantly enhances the task transfer capacity of the world model. The proposed approach surpasses the baseline in almost all transfer tasks, confirming the effectiveness of leveraging Task-Invariant Properties to improve transfer capability. This design allows the model to better capture structural and dynamic features that remain consistent across environments and tasks, thereby maintaining reliable prediction and decision-making capabilities even when confronted with unseen or highly challenging scenarios.

C. Real-World Evaluation

We deployed our method (**Ours**), its non-adaptive variant (**Ours w/o Adapt**), and the baseline (**WMP**) on a Unitree Go2 robot, evaluating their performance across four terrains: Stair (16 *cm*), Climb (52 *cm*), Tilt (33 *cm*), and Crawl (25 *cm*). Success rates were computed over 10 trials per task. To test robustness under dynamic changes, a 2 kg counterweight was attached to the robot's right side, shifting its center of mass. All tests used a velocity command of 0.6 m/s to ensure consistent motion.

The results presented in Tab. I indicate that while the baseline method performed competently in Stair and Crawl tasks, it exhibited substantially inferior performance in Climb and Tilt scenarios compared to our proposed approach. Our method exhibits stronger transfer capability, especially in the 52 *cm* Climb task: the baseline achieved a mere 10% success rate, whereas the ablated version without adaptation (**Ours w/o Adapt**) and the full method (**Ours**) attained success rates of 90% and 100%, respectively. The baseline method exhibited lower performance on the Go2 platform compared to Go1 [17], likely due to Go2's greater mass, larger size, and consequently higher control difficulty and sim-to-real transfer requirements. In contrast, our approach

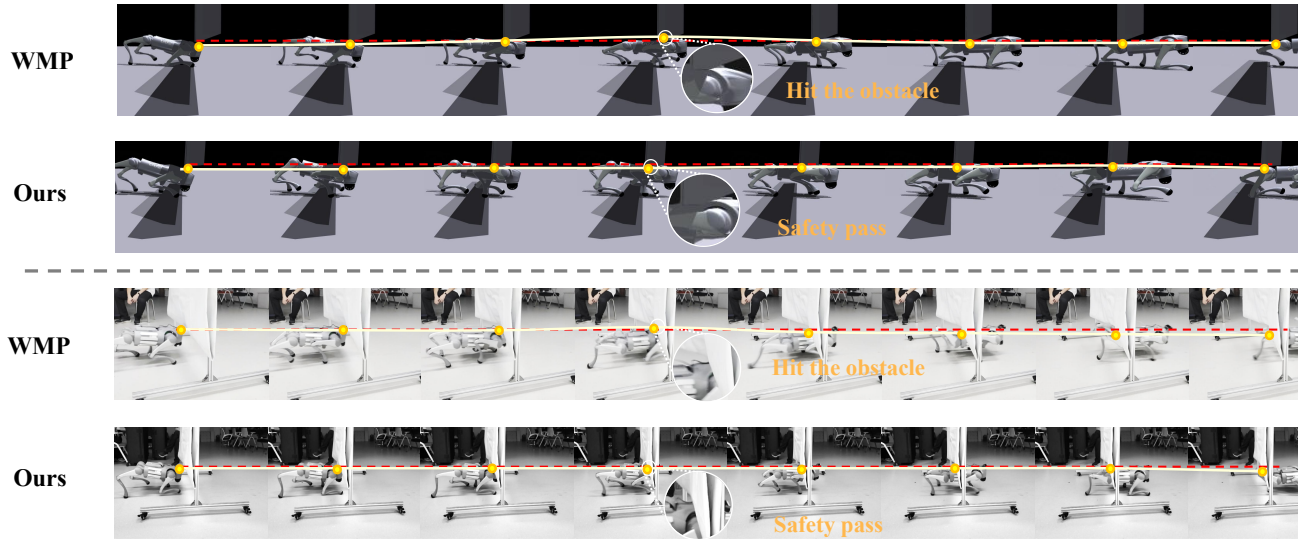

Fig. 5: Performance comparison of various methods on the crawl task across simulated and real environments. Simulation environment (top, above gray dashed line) and real-world environment (bottom). Red line: obstacle height; Yellow dots: robot dog's traversal height at the obstacle. With the obstacle height set to 25 *cm* in both environments, the Baseline method encounters collisions with its head when passing through the obstacles, whereas our method traverses safely. This demonstrates the superior task transfer performance of our method, as well as the consistency in its sim-to-real effectiveness.

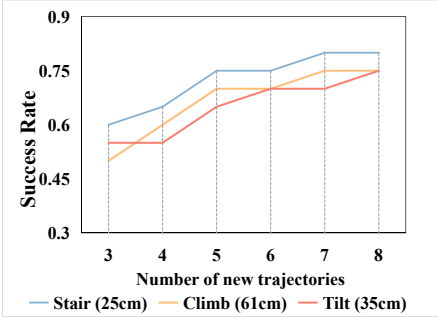

Fig. 6: Ablation on the number n of trajectories added to the mix buffer. We evaluated the performance of our approach in a simulation environment on the Stair (25 *cm*), Climb (61 *cm*), and Tilt (35 *cm*) tasks, collecting additional trajectories on these transfer tasks for adaptation after pre-training DreamTIP. Success rates are calculated over twenty trials.

demonstrates stronger adaptability. The results demonstrate that our method effectively narrows the sim-to-real gap and enables robust policy transfer under challenging real-world conditions, confirming its capacity to handle dynamic environmental variations while maintaining strong cross-task adaptability.

As shown in Fig. 5, the proposed method and the baseline are compared in both simulation and real-world crawl tasks. The results demonstrate that while the baseline causes collisions when the robot dog traverses obstacles, our method achieves a safe pass.

D. Ablation Study

We evaluate three designs of Task-Invariant Properties in the Climb and Tilt transfer tasks under simulation, using average trajectory reward as the metric. The methods include: **DreamTIP-DWL** (predicting privileged information directly), **DreamTIP-GPT5** (our main LLM-driven properties designed method), and **DreamTIP-DeepSeekV3** (another LLM for comparison). As Tab. I shows, LLM-based methods that construct Task-Invariant Properties as auxiliary targets significantly outperform direct privileged information prediction. This result represents the best outcome from three independent TIP generations produced by different LLMs, indicating that the properties designed by LLMs effectively capture essential features that enhance the transfer performance of the world model.

The number of trajectories n used during fine-tuning significantly affects model performance. While more data generally improves robustness by covering broader real-world dynamics, collecting such data is costly. Using our method, we varied n from 3 to 8 and evaluated performance on three simulated transfer tasks: Stair (25 *cm*), Climb (61 *cm*), and Tilt (35 *cm*). As shown in Fig. 6, performance improves noticeably when the number of trajectories increases from 3 to 5, but exhibits diminishing returns beyond this point. We therefore set $n = 5$ in practice to balance effectiveness and computational cost.

VI. CONCLUSIONS

This paper introduces DreamTIP, an extension of the Dreamer framework designed to improve sim-to-real transfer in quadruped robot locomotion through Task-Invariant Properties learning. By leveraging large language models, DreamTIP learns dynamics-robust and task-invariant properties , such as contact stability and terrain clearance, to reduce the reliance on specific dynamic parameters. To further narrow the sim-to-real gap, we propose an efficient adaptation strategy integrating a mix buffer with regularization constraints, which enables stable calibration to real-world dynamics while alleviating representation collapse and catastrophic forgetting. Extensive evaluations across various transfer tasks show that DreamTIP consistently outperforms baselines in both simulated and real-world settings. These

results underscore the value of Task-Invariant Properties in enhancing policy generalization and sim-to-real transfer. However, this work still has some limitations. For example, prolonged operation leads to a certain degree of performance degradation due to the compounding errors in the world model. Future work will explore leveraging richer simulated and real-world data to improve the world model's long-term prediction accuracy and robustness, thereby mitigating performance degradation from error accumulation and offering a robust and scalable framework for adaptive robot learning.

REFERENCES

[1] H. Kim, H. Oh, J. Park, Y. Kim, D. Youm, M. Jung, M. Lee, and J. Hwangbo, "High-speed control and navigation for quadrupedal robots on complex and discrete terrain," *Science Robotics*, vol. 10, no. 102, p. eads6192, 2025.

[2] H. Lai, W. Zhang, X. He, C. Yu, Z. Tian, Y. Yu, and J. Wang, "Sim-to-real transfer for quadrupedal locomotion via terrain transformer," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 5141–5147.

[3] J. Long, Z. Wang, Q. Li, J. Gao, L. Cao, and J. Pang, "Hybrid internal model: Learning agile legged locomotion with simulated robot response," *The Twelfth International Conference on Learning Representations (ICLR)*, 2024.

[4] W. Zhao, J. P. Queralta, and T. Westerlund, "Sim-to-real transfer in deep reinforcement learning for robotics: a survey," in *2020 IEEE symposium series on computational intelligence (SSCI)*. IEEE, 2020, pp. 737–744.

[5] J. Wu, G. Xin, C. Qi, and Y. Xue, "Learning robust and agile legged locomotion using adversarial motion priors," *IEEE Robotics and Automation Letters (RA-L)*, vol. 8, no. 8, pp. 4975–4982, 2023.

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

[7] J. He, C. Zhang, F. Jenelten, R. Grandia, M. Bächer, and M. Hutter, "Attention-based map encoding for learning generalized legged locomotion," *Science Robotics*, vol. 10, no. 105, p. eadv3604, 2025.

[8] C. Zhang, N. Rudin, D. Hoeller, and M. Hutter, "Learning agile locomotion on risky terrains," in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2024, pp. 11 864–11 871.

[9] D. Kim, H. Kwon, J. Kim, G. Lee, and S. Oh, "Stage-wise reward shaping for acrobatic robots: A constrained multi-objective reinforcement learning approach," in *2025 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2025, pp. 10 268–10 274.

[10] J. Long, J. Ren, M. Shi, Z. Wang, T. Huang, P. Luo, and J. Pang, "Learning humanoid locomotion with perceptive internal model," in *2025 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2025, pp. 9997–10 003.

[11] A. Wagenmaker, K. Huang, L. Ke, K. Jamieson, and A. Gupta, "Overcoming the sim-to-real gap: Leveraging simulation to learn to explore for real-world rl," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 37, pp. 78 715–78 765, 2024.

[12] K. Bousmalis, A. Irpan, P. Wohlhart, Y. Bai, M. Kelcey, M. Kalakrishnan, L. Downs, J. Ibarz, P. Pastor, K. Konolige, *et al.*, "Using simulation and domain adaptation to improve efficiency of deep robotic grasping," in *2018 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2018, pp. 4243–4250.

[13] R. P. Poudel, H. Pandya, S. Liwicki, and R. Cipolla, "Recore: Regularized contrastive representation learning of world model," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024, pp. 22 904–22 913.

[14] M. Laskin, A. Srinivas, and P. Abbeel, "Curl: Contrastive unsupervised representations for reinforcement learning," in *International conference on machine learning (ICML)*. PMLR, 2020, pp. 5639–5650.

[15] S. Gao, S. Zhou, Y. Du, J. Zhang, and C. Gan, "Adaworld: Learning adaptable world models with latent actions," *arXiv preprint arXiv:2503.18938*, 2025.

[16] P. Mazzaglia, T. Verbelen, B. Dhoedt, A. Courville, and S. Rajeswar, "Genrl: Multimodal-foundation world models for generalization in embodied agents," *Advances in neural information processing systems (NeurIPS)*, vol. 37, pp. 27 529–27 555, 2024.

[17] H. Lai, J. Cao, J. Xu, H. Wu, Y. Lin, T. Kong, Y. Yu, and W. Zhang, "World model-based perception for visual legged locomotion," in *2025 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2025, pp. 11 531–11 537.

[18] X. Gu, Y.-J. Wang, X. Zhu, C. Shi, Y. Guo, Y. Liu, and J. Chen, "Advancing humanoid locomotion: Mastering challenging terrains with denoising world model learning," *arXiv preprint arXiv:2408.14472*, 2024.

[19] D. Hafner, J. Pasukonis, J. Ba, and T. Lillicrap, "Mastering diverse control tasks through world models," *Nature*, pp. 1–7, 2025.

[20] X. Yang, Z. Ji, J. Wu, and Y.-K. Lai, "Recent advances of deep robotic affordance learning: a reinforcement learning perspective," *IEEE Transactions on Cognitive and Developmental Systems (TCDS)*, vol. 15, no. 3, pp. 1139–1149, 2023.

[21] D. Liu, T. Zhang, J. Yin, and S. See, "Unified locomotion transformer with simultaneous sim-to-real transfer for quadrupeds," *arXiv preprint arXiv:2503.08997*, 2025.

[22] Y. Feng, N. Hansen, Z. Xiong, C. Rajagopalan, and X. Wang, "Finetuning offline world models in the real world," *arXiv preprint arXiv:2310.16029*, 2023.

[23] S. Lee, Y. Seo, K. Lee, P. Abbeel, and J. Shin, "Offline-to-online reinforcement learning via balanced replay and pessimistic q-ensemble," in *Conference on Robot Learning (Corl)*. PMLR, 2022, pp. 1702–1712.

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

[25] P. Wu, A. Escontrela, D. Hafner, P. Abbeel, and K. Goldberg, "Daydreamer: World models for physical robot learning," in *Conference on robot learning (Corl)*. PMLR, 2023, pp. 2226–2240.

[26] Y. J. Ma, W. Liang, G. Wang, D.-A. Huang, O. Bastani, D. Jayaraman, Y. Zhu, L. Fan, and A. Anandkumar, "Eureka: Human-level reward design via coding large language models," *arXiv preprint arXiv:2310.12931*, 2023.

[27] Y. Tang, W. Yu, J. Tan, H. Zen, A. Faust, and T. Harada, "Saytap: Language to quadrupedal locomotion," *arXiv preprint arXiv:2306.07580*, 2023.

[28] A.-C. Cheng, Y. Ji, Z. Yang, Z. Gongye, X. Zou, J. Kautz, E. Bıyık, H. Yin, S. Liu, and X. Wang, "Navila: Legged robot vision-language-action model for navigation," *arXiv preprint arXiv:2412.04453*, 2024.

[29] Y.-J. Wang, B. Zhang, J. Chen, and K. Sreenath, "Prompt a robot to walk with large language models," in *2024 IEEE 63rd Conference on Decision and Control (CDC)*. IEEE, 2024, pp. 1531–1538.

[30] B. Wang, Y. Qu, Y. Jiang, J. Shao, C. Liu, W. Yang, and X. Ji, "Llm-empowered state representation for reinforcement learning," *arXiv preprint arXiv:2407.13237*, 2024.

[31] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[32] J. Yamada, M. Rigter, J. Collins, and I. Posner, "Twist: Teacher-student world model distillation for efficient sim-to-real transfer," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 9190–9196.

[33] S. Beaussant, S. Lengagne, B. Thuilot, and O. Stasse, "Towards zero-shot cross-agent transfer learning via latent-space universal notice network," *Robotics and Autonomous Systems (RAS)*, vol. 184, p. 104862, 2025.

[34] B. Barz and J. Denzler, "Deep learning on small datasets without pre-training using cosine loss," in *Proceedings of the IEEE/CVF winter conference on applications of computer vision (WACV)*, 2020, pp. 1371–1380.

[35] Z. Zhuang, Z. Fu, J. Wang, C. Atkeson, S. Schwertfeger, C. Finn, and H. Zhao, "Robot parkour learning," *arXiv preprint arXiv:2309.05665*, 2023.

[36] A. Kumar, Z. Fu, D. Pathak, and J. Malik, "Rma: Rapid motor adaptation for legged robots," *arXiv preprint arXiv:2107.04034*, 2021.