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

Teaching to Individual Needs: Bidirectional Teacher-Student Learning for Wheeled-Legged Locomotion

Guangsheng Li[1], Charles Wu[2], Xinhua Zheng[1], Shiyu Zhu[1], Shenglan Liu[1*]

***Abstract*— Reinforcement Learning (RL) enables robust and adaptive locomotion in legged and wheeled-legged robots. A common approach is the Teacher-Student (TS) paradigm, in which a teacher policy with privileged information supervises a proprioceptive student. While the TS paradigm has proven effective on legged robots, we encounter two critical issues when applying it to wheeled-legged robots. One issue is multimodal confusion, where teacher actions become multimodal under the student proprioceptive observations, resulting in the student generating averaged action modes. The other is low imitability of teacher actions, as the teacher overlooks their reproducibility by the student. To address these issues, we propose Teaching to Individual Needs (TIN), a bidirectional TS framework. To mitigate multimodal confusion within the student policy, we design a Highest-Weight Component Mixture Density Network (HWC-MDN). By utilizing HWC-MDN, TIN student can explicitly model multimodal action distributions and outputs the highest-weight component. To improve imitability, we propose an Imitation-Aware Reward (IAR) that encourages the teacher to generate more reproducible actions by the student. Simulation experiments show that TIN significantly improves both training efficiency and traversability. Real-world tests illustrate that TIN enables the wheeled-legged robot MagicDog-W to traverse 45 cm obstacles and ascend 45° slopes.**

I. INTRODUCTION

In recent years, Reinforcement Learning (RL) has achieved substantial progress in legged and wheeled-legged locomotion [1]–[6]. It enables robots to achieve robust and stable locomotion, avoiding the intricate modeling and parameter tuning in traditional model-based methods. Directly training RL policies in the real-world remains challenging [7]. As a result, training is typically conducted in simulation with domain randomization. Many studies [8]–[12] adopt Teacher-Student (TS) paradigm to enhance sim-to-real transfer. To begin with, a teacher policy is trained with RL using privileged observations (e.g., local heightmaps). It then supervises a student policy with noisy proprioceptive observations, allowing the student to learn to reproduce the teacher behavior. This enables efficient transfer of locomotion skills from full-state to limited-observation settings.

Nevertheless, the observation asymmetry between teacher and student policies leads to two critical issues in the knowledge transfer of the conventional TS learning framework, especially for wheeled-legged robots.

*Corresponding author: Shenglan Liu. liusl@dlut.edu.cn
This work was supported by the National Natural Science Foundation of China (No. 62376052).
[1]The authors are with the School of Computer Science, Dalian University of Technology, Dalian 116024, China.
[2]The author is with Magiclab Robotics Technology Co., Ltd., Wuxi, China.

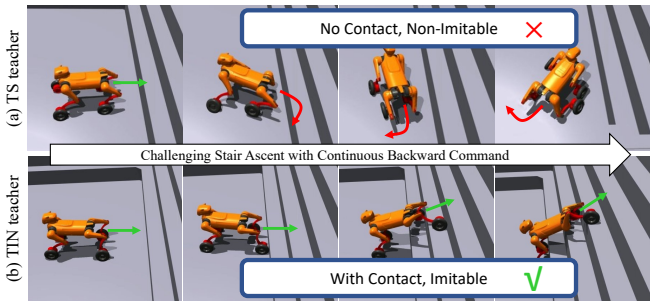

Fig. 1. TIN and TS teachers produce contrasting behaviors during challenging stair ascent with a backward command. Arrows indicate motion trends, where green represents motions that are easy for the proprioceptive student to imitate and red represents motions that are non-imitable. (a) Under traditional TS, the teacher with privileged heightmaps turns to avoid the stairs, generating non-imitable motions. (b) With TIN, the teacher accounts for imitability and continues climbing the stairs as instructed.

(a) Multimodal confusion within the student policy. The student policy often struggles to imitate the teacher actions accurately. With privileged information, the teacher behaves deterministically or follows unimodal action mode. Without it, the same actions appear multimodal under the student limited observations. This effect is amplified on wheeled-legged robots, whose hybrid dynamics yield more diverse action modes. The student tends to average across multimodal action modes [13], [14], leading to inefficient behaviors such as unnecessary energy cost on flat terrain and reduced stability.

(b) Low imitability of teacher actions. As shown in Fig. 1(a), the TS teacher is optimized solely for maximizing environmental rewards, without regard to its actions' imitability with respect to the student. It is more pronounced on wheeled-legged robots due to their fast wheel-assisted motions. Non-imitable teacher actions can induce a large state distribution gap between teacher and student, which reduces training stability and limits generalization [15].

To overcome the issues of the conventional TS learning in wheeled-legged robot locomotion, we propose Teaching to Individual Needs (TIN), a bidirectional TS training paradigm. To address issue (a), TIN introduces the Highest-Weight Component Mixture Density Network (HWC-MDN), which employs a Mixture Density Network (MDN) [16] to explicitly capture the multimodal action distributions of the student policy. During inference, HWC-MDN selects the most dominant action mode, thereby preventing the student from collapsing into averaged behaviors. To tackle issue (b),

we design an Imitation-Aware Reward (IAR) within TIN, enabling the teacher and student policies to adapt jointly. IAR incorporates the student multimodal distribution into the teacher optimization, encouraging actions easier for the student to imitate (shown in Fig. 1(b)). This feedback loop effectively narrows the state distribution gap between the teacher and student policies. TIN achieves robust locomotion over challenging terrains using only proprioceptive sensing. Simulation and real-world experiments on the wheeled-legged MagicDog-W platform demonstrate effectiveness and traversability of TIN.

In summary, our contributions are as follows.

- We design the HWC-MDN, which explicitly models the multimodal action distributions induced by partial observability in the student policy. It outputs the latent action mode with the highest probability.
- We propose the IAR, which evaluates the imitability of teacher actions. By feeding back the student multimodal features to the teacher, the framework guides the teacher to generate more reproducible actions for the student.
- We validate our approach in both simulation and real-world experiments. In simulation, TIN enhances both training efficiency and traversability. In the real-world, it allows MagicDog-W to surmount 45 cm obstacles and ascend 45° slopes.

II. RELATED WORK

A. Teacher-Student Learning

Conventional TS learning is often designed as a two-stage process. In this setup, privileged information in simulation is used to train teacher policies, which are then distilled into proprioceptive students. This method has been applied to enhance obstacle crossing ability [8] and agility [10]. RMA [11] leverages a longer history window of proprioceptive inputs to asynchronously infer environmental latent representation encoded from privileged information, achieving rapid adaptation to environmental changes. Moreover, TO-AMP [12] leverages adversarial motion priors to guide the teacher policy in generating more natural motion styles. These motion styles are then distilled into the student policy to improve agility and generalization.

Although effective, multistage TS learning are prone to increased training complexity, distributional shift, and representation mismatch. To address these issues, recent methods [17]–[19] combine teacher and student learning into a single stage and leverage their interactions. ROA [17] aligns the teacher latent encodings with the student through a regularization process. Additionally, some works enhance interaction by feeding student-encoded latent variables into the shared policy network [18] or by returning student-collected trajectories to the teacher RL updates [19].

B. Model Estimation

Some approaches use dynamics learning or environment state variable estimation to support policy learning while avoiding privileged information as inputs. DayDreamer [20] trains a world model that reduces reliance on real-world interaction samples, making it possible to conduct training directly in physical environments. EstNet [21] improves input encoding by explicitly estimating key environment states. DreamWaQ [22] and HIM [23] both pursue dynamics-based approaches to enhance representation learning. DreamWaQ [22] employs variational auto-encoder (VAE) to regress the next state, thereby implicitly capturing the robot dynamics. By contrast, HIM [23] adopts contrastive learning instead of regression, effectively mitigating the risk of network collapse caused by noisy supervision.

III. METHOD

The overall framework of TIN is shown in Fig. 2. In this section, we present TIN for wheeled-legged locomotion.

A. Preliminaries

We model the environment as an Markov Decision Process (MDP), represented by the tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma \rangle$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $\mathcal{P}$ is the transition dynamics, $\mathcal{R}$ are the reward functions, and $\gamma \in [0, 1]$ is the discount factor. At timestep t, the agent observes the state $\boldsymbol{s}_t \in \mathcal{S}$ and selects an action $\boldsymbol{a}_t \in \mathcal{A}$ from its policy $\pi(\cdot \mid \boldsymbol{s}_t)$. The environment then transitions to the next state $\boldsymbol{s}_{t+1} \sim \mathcal{P}(\cdot \mid \boldsymbol{s}_t, \boldsymbol{a}_t)$ and provides a scalar reward $r_t = \mathcal{R}(\cdot \mid \boldsymbol{s}_t, \boldsymbol{a}_t)$. We employ Proximal Policy Optimization (PPO) to optimize the policy parameters π by maximizing the expected discounted return over trajectories τ:

$$J(\pi) = \mathbb{E}_{\tau \sim \pi} \left[\sum_{t=0}^{\infty} \gamma^t r_t \right]. \tag{1}$$

State Space. The state space formulation distinguishes between proprioceptive and environmental components. Formally, the full state contains two parts $\boldsymbol{s}_t = \langle \boldsymbol{o}_t, \boldsymbol{e}_t \rangle$, where $\boldsymbol{o}_t$ denotes the embodied proprioceptive observation vector and $\boldsymbol{e}_t$ represents the environment privileged information vector. Specifically, $\boldsymbol{o}_t$ integrates onboard sensory measurements, including the base linear velocity command $\boldsymbol{v}_t^{\text{cmd}}$, the base angular velocity command $\boldsymbol{\omega}_t^{\text{cmd}}$, joint positions $\boldsymbol{q}_t$ and velocities $\dot{\boldsymbol{q}}_t$, the previous action $\boldsymbol{a}_{t-1}$, the gravity projected vector $\boldsymbol{g}_t$, and base angular velocity $\boldsymbol{\omega}_t$ in the body frame. Correspondingly, $\boldsymbol{e}_t$ contains environment privileged variable including local terrain heightmap, friction coefficients, motor strength, body mass, and center of mass (COM). The teacher actor π_θ^{tea} operates on the full state $\boldsymbol{s}_t$, while the student policy π_ϕ^{stu} receives only the proprioceptive history windows $\boldsymbol{o}_{t-N:t}$, where N denotes the history length.

Action Space. The action vector $\boldsymbol{a}_t$ serves as the reference commands for the underlying proportional-derivative (PD) controllers and consists of two components, leg joint actions $\boldsymbol{a}_t^{\text{leg}}$ and wheel joint actions $\boldsymbol{a}_t^{\text{wheel}}$:

$$\boldsymbol{a}_t = \langle \boldsymbol{a}_t^{\text{leg}}, \boldsymbol{a}_t^{\text{wheel}} \rangle. \tag{2}$$

For the leg joints, $\boldsymbol{a}_t^{\text{leg}}$ specifies the target joint angles relative to a fixed reference pose. The corresponding torques $\mathbf{T}_t^{\text{leg}}$ are computed as follow:

$$\mathbf{T}_t^{\text{leg}} = M_p^{\text{leg}} \left(\boldsymbol{q}^{\text{ref}} + C^{\text{leg}} \boldsymbol{a}_t^{\text{leg}} - \boldsymbol{q}_t^{\text{leg}} \right) - M_d^{\text{leg}} \dot{\boldsymbol{q}}_t^{\text{leg}}, \tag{3}$$

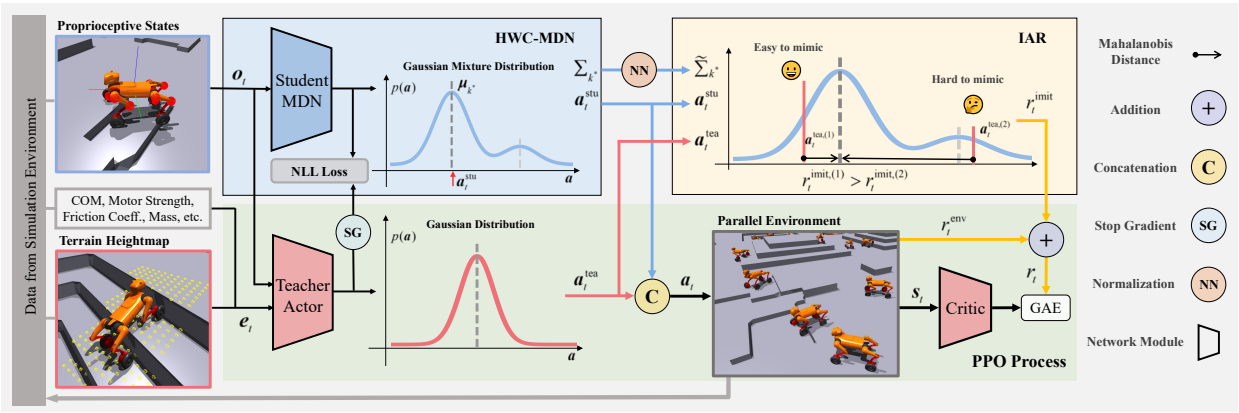

Fig. 2. Overview of the TIN framework. Red lines represent teacher actions, and blue curves represent student distributions in the IAR implementation block diagram. $\boldsymbol{a}_t^{\text{tea},(1)}$ and $\boldsymbol{a}_t^{\text{tea},(2)}$ denote two teacher-sampled actions, and $r_t^{\text{imit},(1)}$ and $r_t^{\text{imit},(2)}$ indicate their corresponding IAR values. The IAR, computed as the Mahalanobis distance between teacher actions and the student distribution, guides the teacher toward actions that are easier for the student to imitate. The teacher actor-critic is optimized by PPO with both environment rewards and IAR, and the student HWC-MDN learns under the teacher supervision.

where $\boldsymbol{q}_t^{\text{leg}}$ and $\dot{\boldsymbol{q}}_t^{\text{leg}}$ are the measured leg joint angles and angular velocities, M_p^{leg} and M_d^{leg} denote the proportional and derivative gains, $\boldsymbol{q}^{\text{ref}}$ is the reference joint configuration for standing, and C^{leg} is a constant scaling factor.

For the wheel joints, $\boldsymbol{a}_t^{\text{wheel}}$ directly denotes the desired angular velocities. The torque command $\mathbf{T}_t^{\text{wheel}}$ is produced by a proportional controller:

$$\mathbf{T}_t^{\text{wheel}} = M_p^{\text{wheel}} \left(C^{\text{wheel}} \boldsymbol{a}_t^{\text{wheel}} - \dot{\boldsymbol{q}}_t^{\text{wheel}} \right), \tag{4}$$

where $\dot{\boldsymbol{q}}_t^{\text{wheel}}$ is the measured angular velocity of the wheel joints, M_p^{wheel} is the proportional gain, and C^{wheel} is a scaling constant.

B. Highest-Weight Component Mixture Density Network

During the imitation learning stage of the TS framework, the student policy lacks access to the privileged information available to the teacher, resulting in a substantial mismatch between their observation spaces. This asymmetry causes unimodal action distribution of the teacher $\pi_\theta^{\text{tea}}(\boldsymbol{a}_t \mid \boldsymbol{s}_t)$ to appear multimodal from the perspective of the student policy $\pi_\phi^{\text{stu}}(\boldsymbol{a}_t \mid \boldsymbol{o}_{t-N:t})$. Even when the student incorporates historical observations to enrich context, a single observation may still correspond to multiple plausible action modes.

For instance, before any part of the robot makes contact with an obstacle, the student cannot reconstruct or reliably infer the obstacle features along the path. As a result, it cannot determine whether the correct action is a wheeled walking mode for flat terrain or a high-stepping mode for obstacle traversal. Conventional deterministic regression or unimodal Gaussian assumptions are inadequate to represent such complex multimodal conditional distributions. To address this challenge, we adopt a MDN to explicitly model the multimodal action distributions of the student policy.

$$\pi_\phi^{\text{stu}}(\boldsymbol{a}_t \mid \boldsymbol{o}_{t-N:t}) = \sum_{k=1}^{K} \beta_k \mathcal{N}_k(\boldsymbol{a}_t \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k), \tag{5}$$

where $\mathcal{N}_k(\cdot \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ denotes the probability density function of the k-th Gaussian component, with K specifying the number of mixture components. Each component is parameterized by a coefficient β_k, a mean vector $\boldsymbol{\mu}_k$, and a covariance matrix $\boldsymbol{\Sigma}_k$, all predicted by a neural network. For computational tractability, we assume $\boldsymbol{\Sigma}_k$ is diagonal, with $\beta_k \geq 0$ and $\sum_{k=1}^{K} \beta_k = 1$.

The student policy, parameterized by ϕ, is trained to imitate the teacher demonstrated actions $\boldsymbol{a}_t^{\text{tea}} \sim \pi_\theta^{\text{tea}}(\cdot \mid \boldsymbol{s}_t)$ by minimizing the Negative Log-Likelihood (NLL) loss:

$$\mathcal{L}(\phi) = -\log \sum_{k=1}^{K} \beta_k \mathcal{N}_k\left(\text{sg}(\boldsymbol{a}_t^{\text{tea}}) \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right), \tag{6}$$

where $\text{sg}(\cdot)$ denotes the stop-gradient operator.

Within the MDN framework, the student policy outputs a Gaussian mixture distribution. Each component represents a latent action mode, and its mixture weight β_k indicates the relative importance of that mode under the current observation. The NLL loss encourages the student to assign high probability density to the teacher actions. In practice, components with larger weights often correspond to more frequent or more salient action modes. For action execution, the student policy selects the mean vector with the highest mixture weight.

$$k^* = \arg\max_k \beta_k, \tag{7}$$

$$\boldsymbol{a}_t^{\text{stu}} = \boldsymbol{\mu}_{k^*}, \tag{8}$$

where $\boldsymbol{\mu}_{k^*}$ is the mean vector of the dominant mode in the student. The resulting action is then given by policy.

This choice of selecting the highest-weight mixture component is motivated by two considerations. On one hand, for a student without privileged information, the teacher action labels may reflect suboptimal decisions. The highest-weight component usually represents the most consistent and concentrated action patterns during training, serving as an implicit denoising mechanism. This helps the student policy learn more robust representations by focusing on reproducible supervision. On the other hand, stochastic action sampling can introduce uncertainty or risky behaviors

during online execution. Using the mean of the highest-weight component yields more reliable and stable control.

C. Imitation-Aware Reward

In conventional two-stage TS learning, the teacher does not account for whether its actions are actually imitable by a student without privileged information. This can force the student into unfamiliar states that the teacher rarely visits, resulting in poor supervision. For example, a teacher with a local heightmap can avoid obstacles by lifting its legs in advance, whereas a student lacking this sensing ability may collide with them. These collision cases are so rare for the teacher that it cannot provide effective guidance.

To tackle this issue, we introduce the IAR, which encourages the teacher to produce actions that the student can effectively imitate. It estimates the imitation difficulty of the teacher actions using mixture density distribution of the student, and incorporates this measure as a regularization term within reward function of the teacher RL training. We implement the IAR within a parallel TS training framework, similar to CTS [18]. This design partitions the training environments into teacher and student groups, allowing for simultaneous optimization. The proposed IAR not only simplifies learning for the student but also enhances the teacher's adaptability to the student's proprioceptive limitations.

A straightforward approach to implement the IAR would be to calculate the probability density of $\boldsymbol{a}_t^{\text{tea}}$ under $\mathcal{N}_{k^*}$. However, the density estimates are numerically unstable in high-dimensional action spaces and unsuitable as reward signals. To address the issue, we adopt a normalized Mahalanobis distance as a robust substitute, providing a stable approximation of the density. The IAR, based on the Mahalanobis distance, is formally defined as:

$$\begin{aligned} r^{\text{imit}} &= -\frac{1}{\sqrt{D}} \left\| \boldsymbol{a}_t^{\text{tea}} - \boldsymbol{a}_t^{\text{stu}} \right\|_{\widetilde{\boldsymbol{\Sigma}}_{k^*}^{-1}} \\ &= -\frac{1}{\sqrt{D}} \sqrt{(\boldsymbol{a}_t^{\text{tea}} - \boldsymbol{a}_t^{\text{stu}})^\top \widetilde{\boldsymbol{\Sigma}}_{k^*}^{-1} (\boldsymbol{a}_t^{\text{tea}} - \boldsymbol{a}_t^{\text{stu}})}, \end{aligned} \tag{9}$$

where D denotes the action dimension. $\widetilde{\boldsymbol{\Sigma}}_{k^*}$ is the highest-weight normalized covariance matrix:

$$\widetilde{\boldsymbol{\Sigma}}_{k^*} = \operatorname{diag}\left(\widetilde{\sigma}_1^2, \widetilde{\sigma}_2^2, \ldots, \widetilde{\sigma}_D^2\right), \tag{10}$$

$$\widetilde{\sigma}_d = \alpha + \eta \cdot \operatorname{sigmoid}\left(\frac{\sigma_d - \operatorname{mean}(\sigma_d)}{\operatorname{std}(\sigma_d)}\right), \tag{11}$$

where σ_d is the original predicted standard deviation, $\operatorname{mean}(\cdot)$ and $\operatorname{std}(\cdot)$ denote the mean and standard deviation operators along the specified dimension within the current batch, and $\operatorname{sigmoid}(\cdot)$ is the sigmoid function. The normalization maps $\widetilde{\sigma}_d$ into the interval $[\alpha, \alpha + \eta]$, where we set $\alpha = 0.3$ and $\eta = 0.7$.

Additionally, the Mahalanobis distance offers a physically intuitive measure, as it accounts for the scaling and correlation among different actuated joints. In the HWC-MDN, a smaller variance $\widetilde{\sigma}_d$ indicates a joint requires finer control and should be assigned a tighter tolerance range, while a larger variance corresponds to dimensions with more relaxed tolerances. The normalized covariance matrix $\widetilde{\boldsymbol{\Sigma}}_{k^*}$ embodies this property. It highlights the relative importance of joint actions and adaptively adjusts the reward weights across different dimensions.

TABLE I
ENVIRONMENT REWARD TERMS.

Reward Term	Equation	Weight
X Vel. Tracking	r^{tx}	0.75
Y Vel. Tracking	$\exp\{-4(v_y^{\text{cmd}} - v_y)\}$	0.75
Yaw Rate Tracking	$\exp\{-4(\omega_z^{\text{cmd}} - \omega_z)\}$	0.75
Z Velocity	v_z^2	-1.0
Roll-Pitch Rate	ω_{xy}^2	-0.05
Orientation	g_z^2	-0.5
Wheel Spin	r^{spin}	-0.1
Joint Acc.	$\|\ddot{\boldsymbol{q}}\|_2^2$	-2.5×10^{-7}
Joint Power	$\|\dot{\boldsymbol{q}}\mathbf{T}\|$	-2.0×10^{-5}
Joint Reference	$\left\|\boldsymbol{q}^{\text{leg}} - \boldsymbol{q}^{\text{ref}}\right\|_2^2$	-0.03
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

IV. Implementation Details

A. Environment Reward

The terms of the environment reward are summarized in Table I. Two task specific rewards designed for wheeled-legged locomotion are introduced in the following.

Owing to the structural properties of wheeled-legged robots, the learned policy tends to favor pure wheeled locomotion during the early stages of training on simple terrains. When adopting the commonly used Gaussian cost function for x-axis velocity $\exp\{-4(v_x^{\text{cmd}} - v_x)\}$ tracking, as in prior work, the robot exhibits unintended behavior: upon facing an obstacle, it deliberately moves backward to gain momentum and then exploits the inertia from colliding with the obstacle to vault over it. Although this strategy still provides positive rewards under the velocity-tracking objective, it deviates substantially from the intended design. To address this, we formulate the x-axis velocity tracking reward as follows:

$$r^{\text{tx}} = -\rho \frac{\Delta v_x^2}{\sqrt{\Delta v_x^2 + \epsilon}}, \tag{12}$$

where $\Delta v_x = v_x^{\text{cmd}} - v_x$. Here, ρ regulates the decay rate of the linear region, and ϵ defines the effective range of the nonlinear region. We set $\rho = 2.5$ and $\epsilon = 0.2$ in our implementation. The reward function is designed to be smoothly nonlinear near zero but gradually approaches linear decay as the distance increases, thereby ensuring that behaviors such as moving backward to build momentum and colliding with obstacles are penalized with negative rewards.

For stable locomotion, wheeled-legged robots generally rely on sustained wheel-ground contact. Yet in practice, we observed a pronounced sim-to-real gap, particularly when traversing complex terrains. During vertical obstacle climbing, the learned simulation policy tends to press the wheels against the vertical face of the obstacle and exploit

wheel-induced friction to lift the legs. In real environments, however, this strategy often causes the wheels to bounce off the obstacle and oscillate, resulting in persistent wheel spinning and failure to lift. To address this gap, we introduce a wheel free-spinning reward r^{spin} that penalizes inconsistencies between wheel angular velocity and actual translational velocity. r^{spin} implicitly guides the policy to switch appropriately between wheeled and legged locomotion modes:

$$r^{\text{spin}} = \| \operatorname{ReLU}(\boldsymbol{\xi}) \|_1, \tag{13}$$

where $\operatorname{ReLU}(\cdot)$ denotes the rectified linear unit function, and $\boldsymbol{\xi}$ represents the criterion for detecting wheel free-spinning.

$$\boldsymbol{\xi} = |\lambda L \dot{\boldsymbol{q}}^{\text{wheel}}| - \|\boldsymbol{v}_{\text{xy}}^{\text{wheel}}\|_2 - \delta, \tag{14}$$

where $|\cdot|$ denotes the element-wise absolute value, λ is a scaling factor for the velocity error, δ is the allowable error threshold, L is the wheel radius, and $\boldsymbol{v}_{\text{xy}}^{\text{wheel}}$ represents the wheel linear velocity in the world frame along the x-y plane. In our implementation, we set $\lambda = 0.8$ and $\delta = 0.1$.

B. Training Details

To achieve robust locomotion, we train policies in Isaac Gym [24] on five terrain tasks: stair ascent, stair descent, uphill slopes, downhill slopes, and discrete obstacles, similar to [23]. The terrain is divided into 400 square sub-terrains, each 10 m wide, covering 20 terrain types and 20 difficulty levels. Stair steps have widths in the range $[0.25, 0.55]$ m and heights in $[0.05, 0.4]$ m. Slopes include random bumps with heights in $[0.01, 0.15]$ m and gradients between $[0°, 45°]$. Discrete obstacles are randomly generated with lengths and widths in $[1.0, 2.0]$ m, heights in $[0.05, 0.45]$ m. Commands for the x-axis velocity v_x^{cmd} and yaw angular velocity ω_z^{cmd} are uniformly sampled from $[-1.0, 1.0]$ m/s and $[-1.0, 1.0]$ rad/s, respectively.

Lateral climbing is infeasible on stairs and obstacles with high difficulty levels. Therefore, y-axis velocity v_y^{cmd} commands are sampled uniformly from $[-1.0, 1.0]$ m/s for difficulty levels below 10, and set to 0 m/s otherwise.

The policy outputs actions at 50 Hz, while the PD controller runs at 200 Hz. Training is performed with 8192 agents in parallel. To reduce the sim-to-real gap, we adopt domain randomization by referring to the parameter terms used in [18], [22], [23] and additionally introducing variations in body rotational inertia and reference joint positions detailed ranges are provided in Table II. Each episode lasts up to 20 seconds and is terminated early if the z-axis gravity vector projected $g_z > 0$.

In the network architecture implementation, the teacher employs a Convolutional Neural Network (CNN) to encode the local terrain heightmap. The remaining parts of the teacher network and the entire student policy are implemented as Multi-Layer Perceptrons (MLPs). The number of mixture components K in the HWC-MDN is set to 3, and the diagonal entries of the covariance matrices Σ_k are parameterized using a softplus activation. Additionally, we integrate the velocity estimation module in [21] to enhance the accuracy of linear velocity estimation under proprioceptive observations.

TABLE II
DOMAIN RANDOMIZATION TERMS.

Randomization Term	Range	Unit
Rotational Inertia factor	$[0.8, 1.2]$	-
Joint Reference Position	$[-0.05, 0.05]$	rad
Payload	$[-3.0, 3.0]$	kg
COM of base	$[-0.05, 0.05]$	m
Wheel Friction	$[0.1, 2.0]$	-
Wheel Restitution	$[0.5, 1.0]$	-
Motor Strength factor	$[0.9, 1.1]$	-
M_p factor	$[0.9, 1.1]$	-
M_d factor	$[0.9, 1.1]$	-
Action Delay	$[0.0, 10.0]$	ms

V. EXPERIMENTS

To comprehensively evaluate the effectiveness of TIN and validate its advantages over existing approaches, we conduct extensive and systematic simulation experiments that include quantitative performance comparisons with representative methods and detailed ablation studies on key components, followed by real-world deployment evaluations.

A. Simulation Comparison Experiments

To conduct a comprehensive evaluation, we benchmark TIN against the following representative methods:

1) **Oracle**: Represents the theoretical upper bound, where both the Actor and Critic have full access to privileged states and are trained with PPO.
2) **Baseline**: A classical asymmetric Actor-Critic architecture. The Critic receives privileged inputs, while the Actor observes only proprioceptive states, and both are trained with PPO.
3) **TS Student**: A two-stage TS method, adapted from [8], where the **Oracle** policy serves as the teacher (TS Teacher).
4) **ROA** [17]: A one-stage TS method with extra regularization process. For fair comparison, we extend the original method by integrating the same CNN heightmap encoder used in TIN.
5) **DreamWaQ** [22]: A method that learns dynamics by predicting future states from historical proprioceptive sequences. Our implementation follows the description provided in the original paper.
6) **HIM** [23]: A method that learns dynamics by contrastive learning. It treats inaccessible privileged states as external disturbances.

Fig. 3 shows the terrain levels reached during training. Each experiment is repeated with five random seeds, and shaded regions indicate standard deviations. The results demonstrate that TIN achieves the best performance among deployable policies and validate its capability to traverse complex terrains. Compared with the Oracle method, which optimizes only for environmental rewards, TIN Teacher shows slightly lower performance because it also considers action imitability. Nevertheless, TIN Student outperforms TS Student, which lacks this feedback mechanism. In addition,

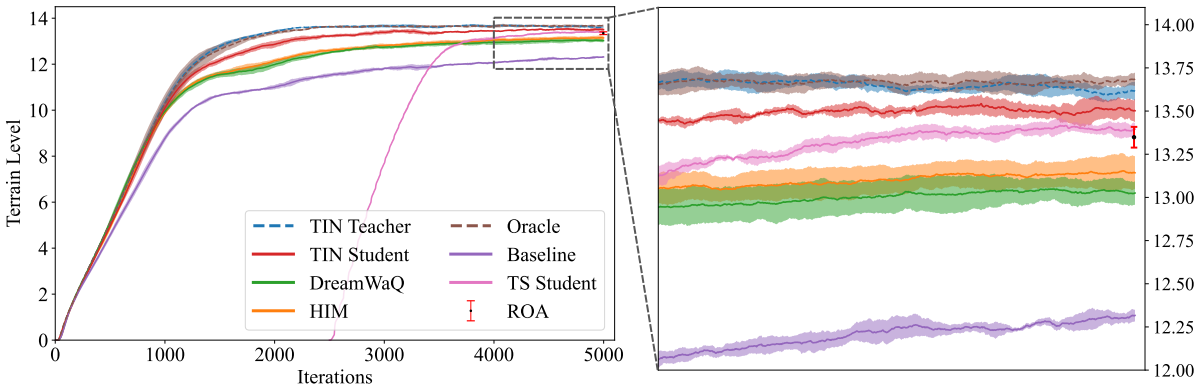

Fig. 3. Average terrain level during training. Solid lines indicate deployable policies, and dashed lines indicate non-deployable ones.

TABLE III
COMPARISON OF METHODS ON LOCOMOTION METRICS.

	Tracking Error			Stability Metric			Performance Metrics	
	X Vel. ↓	Y Vel. ↓	Yaw Rate ↓	Z Vel. ↓	Roll Rate ↓	Pitch Rate ↓	Free-spin. Freq. ↓	Success Rate ↑
Baseline	0.086	0.059	0.046	0.130	0.556	0.516	0.063	0.609
DreamWaQ	0.096	0.073	0.044	0.126	**0.440**	0.550	0.060	0.641
HIM	0.066	0.051	**0.041**	0.122	0.455	**0.470**	0.057	0.653
ROA	0.075	0.046	0.058	0.134	0.542	0.562	0.055	0.669
TS Student	0.098	0.057	0.067	0.141	0.770	0.760	0.073	0.675
TIN	**0.052**	**0.026**	0.049	**0.120**	0.571	0.576	**0.051**	**0.711**

TS-based learning approaches (TIN, TS Student, ROA) consistently outperform dynamics-based methods (DreamWaQ, HIM) in terms of maximum terrain level reached. It suggests that in challenging wheeled-legged locomotion tasks, teacher policies with privileged information can rapidly acquire effective strategies. Such policies provide more informative supervision to students and substantially reduce failure rates on difficult terrains. In contrast, dynamics-based methods that lack privileged guidance struggle to discover effective action sequences, which limits their ability to learn robust representations of complex dynamics.

As shown in Table III, we evaluate each method on terrains of varying types and difficulty using over 20 seed velocity command combinations, with each episode lasting 20 seconds. Tracking Error is assessed by the mean squared errors (MSE) of linear velocity tracking along the x and y axes, computed as $\|v_{x,y}^{\text{cmd}} - v_{x,y}\|_2^2$, and yaw rate tracking around the z-axis, computed as $\|\omega_z^{\text{cmd}} - \omega_z\|_2^2$. A run is considered successful if the agent reaches beyond the terrain boundary within the time limit. For all successful runs, we further evaluate stability using the mean absolute error (MAE) of vertical velocity $\|v_z\|_1$, roll and pitch rates $\|\omega_{x,y}\|_1$, and the frequency of wheel free-spinning $\text{ReLU}(|L\dot{\boldsymbol{q}}^{\text{wheel}}| - \|\boldsymbol{v}_{xy}^{\text{wheel}}\|_2)$, while the overall success rate indicates task completion performance.

The results show that TIN achieves the highest traversal success rate and overall state-of-the-art performance across most metrics. In particular, TIN attains a success rate of 0.711, which surpasses the second-best method by 3.6 percentage points. TIN also yields the lowest tracking MAE in x-axis and y-axis velocities, reducing the errors by 21.2% and 43.5% compared to the second-best method. These results demonstrate that TIN effectively improves locomotion accuracy, stability, and robustness. HIM and DreamWaQ, however, exhibit notably strong performance in yaw rate tracking and associated stability measures (MAEs of roll and pitch rates). Possibly due to their relatively low traversal success rates on high difficulty terrains, the training process emphasizes angular-velocity precision and system stability over overall traversal capability. In addition, HIM and DreamWaQ directly optimize deployable policies through RL, which inherently improves control stability.

B. Simulation Ablation Study

We performed ablation experiments to assess the impact of the IAR, HWC-MDN, local heightmap CNN encoder, linear velocity estimation module, and wheel free-spinning reward. The results of these experiments are summarized in Table IV.

Removing the IAR (TIN w/o IAR) resulted in the pronounced performance degradation across all metrics. This removal led to a 25.5% increase in wheel free-spinning frequency, a 26.7% increase in z-axis velocity error, and increases in both joint velocity and power consumption by 14.1% and 30.0%, respectively. Based on TIN w/o IAR results, we can infer why performance degraded. The absence of the IAR means the teacher policy is not penalized for generating actions that are difficult for the student to replicate. This leads the teacher to produce actions that are less compatible with the student limited perceptual and motor capabilities. When faced with non-imitable actions,

TABLE IV

ABLATION STUDY RESULTS ON LOCOMOTION METRICS.

	Stability Metric			Joint-level Metrics			Performance Metrics	
	Z Vel. ↓	Roll Rate ↓	Pitch Rate ↓	Vel. ↓	Torque ↓	Power ↓	Free-spin. Freq. ↓	Success Rate ↑
TIN w/o Spin Rew.	**0.118**	0.680	0.641	43.0	81.5	154.9	0.084	0.663
TIN w/o Lin. Est.	0.129	0.617	0.608	43.2	79.4	160.0	0.060	0.678
TIN w/o CNN	0.124	0.650	0.587	41.4	80.5	152.4	0.066	0.667
TIN w/o HWC-MDN	0.134	0.620	0.620	41.8	78.8	155.0	0.060	0.684
TIN w/o IAR	0.152	0.675	0.629	44.6	85.3	188.1	0.064	0.708
TIN	0.120	**0.571**	**0.576**	**39.1**	**79.2**	**144.7**	**0.051**	**0.711**

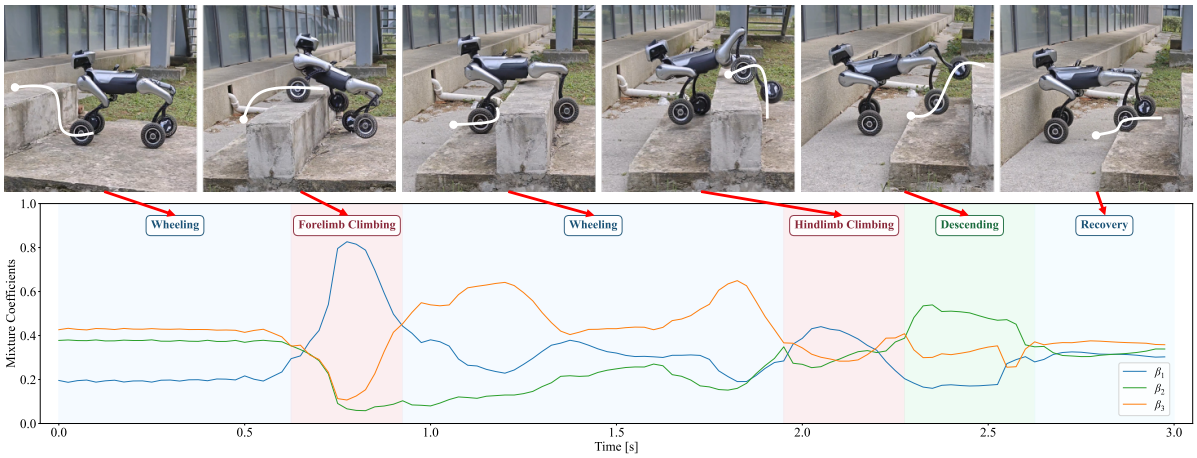

Fig. 4. Temporal evolution of mixture coefficients β_k as the wheeled-legged robot MagicDog-W traverses a 30 cm high, 25 cm wide obstacle. The white line in the figure represents the wheel-end trajectory of the wheeled-legged robot dog. The HWC-MDN decomposes actions into three motion primitives: β_1 (blue) for high-torque climbing, β_2 (orange) for landing and recovery, and β_3 (green) for wheel-driven gait.

the student struggles to follow the teacher guidance. These behaviors can destabilize the robot locomotion and are energy-inefficient. Ultimately, the IAR helps bridge this gap by aligning the teacher actions with the student ability to imitate. This ensures that the student remains in states that are well-supervised and familiar to the teacher, leading to more stable and efficient behavior.

Replacing HWC-MDN (TIN w/o HWC-MDN) with a single Gaussian model resulted in a substantial decline in performance. The frequency of wheel free-spinning jumped by 17.6%, while stability errors in angular velocities along the x and y axes increased by 8.6% and 7.6%, respectively. From the student perspective, the action distribution of the teacher policy is inherently multimodal, assuming a unimodal Gaussian is inadequate. Such a unimodal model cannot capture the full complexity of the teacher decisions. In contrast, our HWC-MDN explicitly models multiple action modes, providing the student with larger representational capacity and greater decision-making flexibility. This enhanced capability directly translates to more accurate and stable behaviors.

Additionally, removing other modules likewise leads to noticeable performance drops, reinforcing the effectiveness of the overall system design. Although removing the wheel free-spinning reward (TIN w/o Spin Rew.) improves the stability of the z-axis velocity by 1.7%. It causes substantial degradation in other metrics, especially a 4.8 percentage point drop in the success rate. This suggests that the free-spinning reward plays a critical role in facilitating stable and transferable behaviors, and its removal adversely affects the policy sim-to-real transferability.

C. Real-World Experiments

As shown in Fig. 4, we analyzed the temporal evolution of the mixture coefficients β_k to determine weather HWC-MDN can autonomously recognize and switch between locomotion modes. For this experiment, the wheeled-legged MagicDog-W traversed a 30 cm high and 25 cm wide obstacle. The results confirm that our HWC-MDN effectively breaks down complex continuous actions into several discrete motion primitives. These primitives possess clear physical interpretations, and the network can automatically switch between them according to the environmental context.

During ascent, when the robot limbs contact the obstacle and begin climbing, component associated with coefficient β_1 dominates. This component produces high-torque, large step-height actions that enable the legs to overcome gravity and lift the body onto the obstacle. In the descent phase, once the center of mass passes the obstacle peak, the component corresponding to β_2 rapidly becomes dominant. This corresponds to a landing and recovery mode, where the policy commands compliant leg extension to absorb impact forces and adjust body posture for a smooth transition. Before encountering the obstacle and after completing

TABLE V

COMPARISON OF REAL-WORLD ROBOT LOCOMOTION CAPABILITIES.

	Obs. H.	Slope	Stand. H.	Robot
Omni [25]	0.10 m	20°	≈ 0.36 m	OmniQuad
ANL [6]	0.40 m	31°	≈ 0.55 m	Swiss-Mile
TIN	**0.45** m	**45**°	≈ 0.37 m	MagicDog-W

traversal, component remains with β_3 dominant, representing a wheel-driven gait that provides stable and energy-efficient locomotion.

This experiment intuitively demonstrates the effectiveness of our approach. HWC-MDN does not simply blend multiple actions into a fuzzy mixture, but explicitly learns and allocates specialized sub-policies to distinct terrain phases. Such context-sensitive switching enables the robot to perform complex whole-body coordinated motions smoothly and stably, thereby addressing the multimodal action distribution issue caused by perceptual asymmetry.

We further benchmark our method against other wheeled-legged locomotion methods deployed in real-world settings, as summarized in Table V. For each study, we report the maximum obstacle height (Obs. H.) and slope angle (Slope) successfully traversed in practice, as well as the standing height (Stand. H., the robot nominal height when standing). The results show that our approach enables the wheeled-legged robot MagicDog-W to surmount obstacles up to 1.22 times its body height and to traverse irregular slopes of 45°, demonstrating clear performance advantages over competing methods.

VI. CONCLUSIONS

This work introduces TIN, a bidirectional TS learning for wheeled-legged locomotion. TIN integrates an HWC-MDN and an IAR to address multimodal confusion within the student policy and the low imitability of teacher actions. Experiments show that HWC-MDN effectively captures diverse action modes, while IAR feeds back the student's learned distributional features to the teacher, thereby enhancing the imitability of teacher actions. This leads to stable and energy-efficient locomotion behaviors.

In future work, TIN could be extended to exteroceptive settings. Even with sensors such as depth cameras or LiDAR, discrepancies between teacher and student capabilities can persist. For instance, these sensors often have limited fields of view. Depth cameras typically do not provide 360° coverage, resulting in occlusions and incomplete observations. Consequently, their perception cannot fully match the privileged information available in simulation. TIN offers a promising approach to mitigating this perception gap.

REFERENCES

[1] N. Rudin, D. Hoeller, P. Reist, and M. Hutter, "Learning to walk in minutes using massively parallel deep reinforcement learning," in *Conference on robot learning*. PMLR, 2022, pp. 91–100.

[2] A. Escontrela, X. B. Peng, W. Yu, T. Zhang, A. Iscen, K. Goldberg, and P. Abbeel, "Adversarial motion priors make good substitutes for complex reward functions," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 25–32.

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

[4] D. Hoeller, N. Rudin, D. Sako, and M. Hutter, "Anymal parkour: Learning agile navigation for quadrupedal robots," *Science Robotics*, vol. 9, no. 88, p. eadi7566, 2024.

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

[6] J. Lee, M. Bjelonic, A. Reske, L. Wellhausen, T. Miki, and M. Hutter, "Learning robust autonomous navigation and locomotion for wheeled-legged robots," *Science Robotics*, vol. 9, no. 89, p. eadi9641, 2024.

[7] C. Tang, B. Abbatematteo, J. Hu, R. Chandra, R. Martín-Martín, and P. Stone, "Deep reinforcement learning for robotics: A survey of real-world successes," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 39, no. 27, 2025, pp. 28 694–28 698.

[8] J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter, "Learning quadrupedal locomotion over challenging terrain," *Science robotics*, vol. 5, no. 47, p. eabc5986, 2020.

[9] Z. Zhuang, S. Yao, and H. Zhao, "Humanoid parkour learning," *arXiv preprint arXiv:2406.10759*, 2024.

[10] G. B. Margolis, G. Yang, K. Paigwar, T. Chen, and P. Agrawal, "Rapid locomotion via reinforcement learning," *The International Journal of Robotics Research*, vol. 43, no. 4, pp. 572–587, 2024.

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

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

[13] I. Shenfeld, Z.-W. Hong, A. Tamar, and P. Agrawal, "Tgrl: Teacher guided reinforcement learning algorithm for pomdps," in *Workshop on Reincarnating Reinforcement Learning at ICLR 2023*, 2023.

[14] H. Nguyen, A. Baisero, D. Wang, C. Amato, and R. Platt, "Leveraging fully observable policies for learning under partial observability," *arXiv preprint arXiv:2211.01991*, 2022.

[15] A. Warrington, J. W. Lavington, A. Scibior, M. Schmidt, and F. Wood, "Robust asymmetric learning in pomdps," in *International Conference on Machine Learning*. PMLR, 2021, pp. 11 013–11 023.

[16] C. M. Bishop, *Neural Networks for Pattern Recognition*. Oxford, U.K.: Oxford University Press, 1995.

[17] Z. Fu, X. Cheng, and D. Pathak, "Deep whole-body control: learning a unified policy for manipulation and locomotion," in *Conference on Robot Learning*. PMLR, 2023, pp. 138–149.

[18] H. Wang, H. Luo, W. Zhang, and H. Chen, "Cts: Concurrent teacher-student reinforcement learning for legged locomotion," *IEEE Robotics and Automation Letters*, 2024.

[19] F. Wu, X. Nal, J. Jang, W. Zhu, Z. Gu, A. Wu, and Y. Zhao, "Learn to teach: Sample-efficient privileged learning for humanoid locomotion over real-world uneven terrain," *IEEE Robotics and Automation Letters*, 2025.

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

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

[22] I. Nahrendra, B. Yu, and H. Myung, "Dreamwaq: Learning robust quadrupedal locomotion with implicit terrain imagination via deep reinforcement learning," *arXiv preprint arXiv:2301.10602*, 2023.

[23] J. Long, Z. Wang, Q. Li, J. Gao, L. Cao, and J. Pang, "Hybrid internal model: Learning agile legged locomotion with simulated robot response," *arXiv preprint arXiv:2312.11460*, 2023.

[24] V. Makoviychuk, L. Wawrzyniak, Y. Guo, M. Lu, K. Storey, M. Macklin, D. Hoeller, N. Rudin, A. Allshire, A. Handa, *et al.*, "Isaac gym: High performance gpu-based physics simulation for robot learning," *arXiv preprint arXiv:2108.10470*, 2021.

[25] F. Iotti, A. Ranjan, F. Angelini, and M. Garabini, "Omniquad: A wheeled-legged hybrid robot with omnidirectional wheels," *Mechanism and Machine Theory*, vol. 214, p. 106125, 2025.