Learn to Teach: Sample-Efficient Privileged Learning for Humanoid Locomotion over Diverse Terrains

Feiyang Wu[1], Xavier Nal[2], Jaehwi Jang[1], Wei Zhu[1], Zhaoyuan Gu[1], Anqi Wu[1], Ye Zhao[1]

***Abstract*— Humanoid robots promise transformative capabilities for industrial and service applications. While recent advances in Reinforcement Learning (RL) yield impressive results in locomotion, manipulation, and navigation, the proposed methods typically require enormous simulation samples to account for real-world variability. This work proposes a novel one-stage training framework—Learn to Teach (L2T)—which unifies teacher and student policy learning. Our approach recycles simulator samples and synchronizes the learning trajectories through shared dynamics, significantly reducing sample complexities and training time while achieving state-of-the-art performance. Furthermore, we validate the RL variant (L2T-RL) through extensive simulations and hardware tests on the Digit robot, demonstrating zero-shot sim-to-real transfer and robust performance over 12+ challenging terrains without depth estimation modules. Experimental videos are available online [1].**

I. Introduction

Reinforcement Learning (RL) has revolutionized robotic control by tackling complex tasks such as dynamic locomotion [1]–[3]. Despite these achievements, policies trained in simulators often falter when deployed into the real world due to the inevitable simulation-to-reality gap [4]. Although domain randomization [5] is widely used to mitigate these discrepancies, it incurs significantly higher sample complexity as agents must explore extensive environmental variations.

Recently, teacher-student learning methods have demonstrated promising results by leveraging an expert teacher to guide students with restricted observation spaces [6]–[8]. However, the conventional two-stage training discards valuable teacher interactions with the environment and often suffers from mismatches between independently trained teachers and students. To address these issues, we propose Learn-to-Teach (L2T): a unified training framework that co-trains teacher and student agents in a single, interactive stage, where the student fully utilizes the collected samples.

To quantify L2T's advantages, we implement L2T-RL, an RL variant, and benchmark its performance on humanoid locomotion tasks using the Digit robot in Isaac Lab—a state-of-the-art GPU-accelerated simulator [9]. Our results show that L2T-RL can achieve stable and superior performance compared to the conventional teacher-student learning paradigm, requiring 50% fewer samples. Consequently, we deploy our trained policy on the robot Digit and conduct extensive hardware experiments in indoor and outdoor environments. Strikingly, the resulting student agent, a lightweight LSTM-based policy, exhibits zero-shot sim2real transfer on the physical Digit robot across a wide range of terrains, including gravel, sand, grass, and slopes (Fig. 1). We also test our control policy on various perturbations such as push recovery, walking under payload, and walking on slippery or wet terrains or with the wind blowing (see Fig. 6 and the supplementary video). Our contributions are as follows:

Efficient training framework: We propose a joint teacher-student training paradigm that optimizes both policies simultaneously. Unlike prior decoupled approaches, our framework enables cross-agent knowledge transfer to the student policy by dynamically utilizing the teacher's training samples directly within a single training stage, avoiding the need for training from scratch in a separate stage.

Mitigation of teacher-student policy divergence: We propose a sample mixing strategy to alleviate the imitation gap between the teacher and student, which traditional privileged learning is unable to address [10]. Both agents will contribute to the replay buffer following a predefined schedule when collecting samples. Mixing samples enables a joint optimization process that mitigates policy divergence while promoting sample efficiency by letting both agents explore Out-of-Distribution (OOD) data.

Humanoid RL agent deployment: We demonstrate real-world locomotion agility through hardware experiments. Our policy, trained entirely in simulation, enables a physical humanoid robot to reliably traverse $12+$ real-world terrains (concrete, gravel, slopes, stairs, etc.) and withstand dynamic perturbations (pushes and payloads) without offline fine-tuning. The policy achieves high success in unstructured environments, matching the teacher's robustness despite using only proprioceptive inputs without depth estimation modules.

II. Related work

Teacher-student learning: In the robotics learning community, teacher-student learning [6]–[8] has gained significant attention due to its applicability and effectiveness in addressing sim2real challenges. In this framework, the teacher agent is trained with complete knowledge of the state space. After obtaining an expert-level teacher, a student agent is trained in an observation space that follows the available sensor configurations on hardware, where the goal is to imitate the teacher's action [8]. In this work, we extend this learning framework by training the teacher and the student simultaneously in a single stage. Prior work [11] proposed a

[1]Feiyang Wu, Jaehwi Jang, Wei Zhu, Zhaoyuan Gu, Anqi Wu, and Ye Zhao are with Georgia Institute of Technology, GA 30332, USA. {feiyangwu, jjang318, wzhu328, zgu78, anqiwu, yezhao}@gatech.edu

[2]Xavier Nal is with the School of Engineering, Ëcole Polytechnique Fédérale de Lausanne, 1015 Lausanne, Switzerland. xavier.nal@alumni.epfl.ch.

[1]https://lidar-learn-to-teach.github.io

Fig. 1: We implement our L2T-RL algorithm on our bipedal walking robot Digit and deploy it on diverse terrain with various environmental conditions such as wet grass, gravel, sandy terrain, and slippery surfaces.

method termed Concurrent Teacher Student (CTS) learning, which also explored the idea of co-training both agents. However, CTS trains a shared policy across the teacher and the student, only differentiating the observation encoder and the critic. This potentially disrupts the training process as the privileged critic naturally rewards actions that might not seem valuable to the student. In comparison, L2T trains two separate agents, with the option of sharing an encoder network or using an asymmetric learning style critic.

Learning with partial observation: Recent advancements in RL under partial observability have significantly improved the ability of robotic systems to operate in complex, uncertain environments [8]. Contemporary approaches often leverage deep recurrent architectures, such as [12], to infer latent state representations from sequential data, effectively bridging traditional POMDP solvers with modern deep RL frameworks. In robotics control, practitioners construct history-dependent policies from a sliding-window style observation or rely on the recurrent architecture of the policy network. At the same time, asymmetric learning has emerged as another effective strategy to bridge the gap between training and execution [13]. In these approaches, the critic network is provided access to privileged, full-state information during training. Recent works have demonstrated that such asymmetric actor-critic frameworks improve sample efficiency and enhance policy robustness [14]. In this work, we combine these learning techniques, utilizing a recurrent network and an asymmetric critic, to solve the underlying POMDP problem efficiently.

Learning from demonstrations: Learning from demonstrations (LfD) has attracted significant interest in the robot learning field due to the growing abundance of robot data and the popularity of simple yet effective imitation learning (IL) frameworks [15]. LfD has demonstrated impressive results in controlling robot manipulators for tabletop tasks [16]. A recent surge of LfD studies in humanoids and bipeds have shown the promising potential of whole-body control and loco-manipulation [16]–[19]. However, supervised learning demands high-quality behavior data, oftentimes through elaborate data collection pipelines [16], [19] and/or needs accurate re-targeting to robot states from datasets with different morphologies On the other hand, the prevalent IL loss is known to be suboptimal from a pure learning perspective in previous studies [20]. Thus, in this work, we focus on a generic algorithm framework to address the sim2real gap alone, without the interference of possible issues brought up by LfD methods. Furthermore, our proposed framework can be easily extended to the LfD setting, which we leave as a future direction.

Bipedal locomotion over complex terrain: Humanoid robots recently have gained increasing interest due to their applicability and versatility [17]–[19], [21]–[25], ranging from locomotion [26], to manipulation [27]. Prior bipedal locomotion works [26], [28] have explored the conventional teacher-student learning paradigm in locomotion tasks. However, the training process can take significant samples even with an elaborate training environment design. Concurrent works also incorporate memory structure into the policy architecture [11], [26], [29], or learning from demonstrations collected from various data sources such as human motion data [30] or data generation using model-based methods [27]. In comparison, we design straightforward and intuitive reward functions for bipeds in general terrain settings, offering a simple yet effective solution.

III. Methods

This section introduces our problem setup and notations and presents our learning framework. A Markov Decision

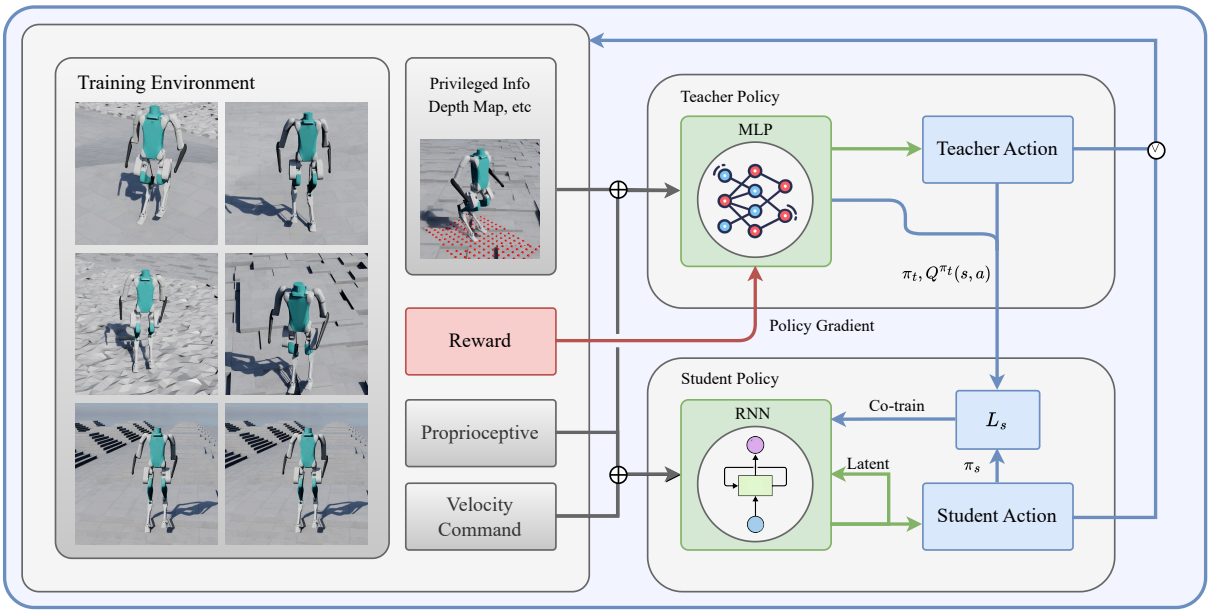

Fig. 2: Learn to Teach (L2T) training pipeline. The teacher agent utilizes a neural network for the policy, which comprises three fully connected layers with sizes $[512, 256, 128]$. The student agent's policy network deployed on the robot is an LSTM network with a hidden layer of 128 units, followed by fully connected layers with shape $[512, 256, 128]$. The teacher learns via conventional RL methods, while the student updates its policy by imitating the teacher.

Process (MDP), denoted as $\mathcal{M}$, is described by a tuple: $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, R, \mathcal{P}, \Pi, \gamma \rangle$, where an agent starts with a given state s_0 following the initial state distribution $p(s_0)$. At any time step t, the agent at the current state $s_t \in \mathcal{S}$ takes an action $a_t \in \mathcal{A}$ following the agent's policy $\pi \in \Pi$, which defines a probability distribution over action space for each state. While receiving an instantaneous scalar reward $r(s_t, a_t) \in \mathbb{R}$, the state of the agent transitions to a new state $s_{t+1} \in \mathcal{S}$ following a transition model $\mathcal{P}(\cdot|s_t, a_t)$. γ specifies the discount factor. The goal of the agent is to maximize the expected discounted sum of rewards the agent receives over time $\max_\pi \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)\right]$, where the expectation is taken over actions $a_t \sim \pi(\cdot|s_t)$, and transition probabilities $s_{t+1} \sim \mathcal{P}(\cdot|s_t, a_t)$ and initial distribution $s_0 \sim p(s_0)$. A Partially Observable Markov Decision Process (POMDP) is further coupled with an observation model $O(\cdot|s_t)$, which is generally hidden from the agent. At each time step, the agent only observes $o_t \sim O(\cdot|s_t)$ sampled from the observation model. Then, the agent takes an action based on o_t, following its policy $\pi(\cdot|o_t)$, and subsequently receives a reward from the environment $r(s_t, a_t)$. To train a policy robust to various observation models, domain randomization is often applied. For example, by adding noise to the state s_t, the trained agent's policy can handle observations $o_t = s_t + \epsilon$, where ϵ can be any noise distribution.

We train the teacher with a generic actor-critic method. During training, the teacher interacts with the environment, generates samples, and stores them in a replay buffer [31]. In standard teacher-student frameworks, the teacher's collected samples are used solely for training the teacher policy π_t and then discarded. In contrast, our L2T framework co-trains the student with the teacher, reusing the teacher's samples across all iterations. Fig. 2 illustrates our learning framework, and Algorithm 1 presents the pseudo-code. Specifically, as the teacher interacts with the environment, we record samples (s, a, r, s'), where s' denotes the next observation, the corresponding noisy observations o generated by domain randomization, and o', the next observation. In other words, we store (s, o, a, r, s', o') as training data in the replay buffer. The student updates its policy at each iteration by sampling mini-batches from the replay buffer, but its policy relies solely on the collected noisy data. This joint training procedure greatly improves sample efficiency as both agents learn together, without the need for a separate student training stage as that used in the traditional set-up.

Another key challenge in the teacher-student framework is the discrepancy between the teacher's and the student's observation spaces. Traditional teacher-student learning methods fall short because the teacher does not account for the limitations of the student's observations, leading to suboptimal guidance [10]. To bridge this gap, we introduce a sample-mixing mechanism in which the student collects its own samples directly from the environment. These student-generated samples—including actions and the resulting observations—are incorporated into the replay buffer as if they were produced by the teacher. This injection of OOD data helps reduce the imitation gap between the two agents.

To systematically blend teacher and student experiences, we define a mixture coefficient α_{mix}. At each time step, the action a is determined by a probabilistic mixture of the

teacher's policy π_t and the student's policy π_s. Specifically, the action selection is defined as follows:

$$a = \begin{cases} \text{sample } \pi_s(\cdot \mid o), & \text{with probability } \alpha_{\text{mix}}, \\ \text{sample } \pi_t(\cdot \mid s), & \text{with probability } 1 - \alpha_{\text{mix}}. \end{cases} \quad (1)$$

Additionally, α_{mix} is scheduled linearly from 0 to a predefined constant over the course of training, which is 0.2 in our implementation. This formulation ensures that, initially, the teacher's guidance dominates the action selection, but as training progresses, the student's policy increasingly influences the learning. The scheduling of α_{mix} helps balance the contributions of both policies and guarantees the stability of the training process. By combining these strategies, our framework leverages both the teacher's guidance and the student's explorative capabilities to achieve more robust learning outcomes.

We implement a variant of our framework, L2T-RL. We apply policy gradient methods to update the teacher policy π_t. At each iteration k, the critic is updated by estimating the value functions: $V^{\pi}(s) := \mathbb{E}_{\pi}\left[\sum_{t=0}^{\infty} \gamma^t\, r(s_t, a_t) \,|\, s_0 = s\right]$, $Q^{\pi}(s,a) := \mathbb{E}_{\pi}\left[\sum_{t=0}^{\infty} \gamma^t\, r(s_t, a_t) \,|\, s_0 = s, a_0 = a\right]$, where $V^{\pi}(s)$ is the value function, and $Q^{\pi}(s,a)$ is the discounted action-value function. For brevity, we will only use the subscript t to denote the teacher from now on and use subscript s to denote the student. Subsequently, the teacher's policy is updated via a Policy Mirror Descent [32] step with a step size β:

$$\min_{p_t} \; -\beta \langle Q^{\pi_t}(s,\cdot), p_t(\cdot|s)\rangle + \text{KL}(\pi_t \| p_t) \quad \forall s \in \mathcal{S}, \quad (2)$$

where the optimal p_t represents the teacher policy in the next iteration. Any policy improvement scheme can be fit into the L2T framework. This formulation encompasses a range of policy gradient methods [32], such as Proximal Policy Optimization (PPO) [33] and Soft Actor-Critic (SAC) [34]. In our experiments on the Digit robot, we employ a PPO-style update. However, our framework can be easily extended to various learning methods, including imitation learning methods, or Inverse Reinforcement Learning (IRL) methods such as in [20].

For the student policy, we consider two choices for loss functions. First, we can minimize an imitation loss between the teacher's and the student's policies:

$$\min_{p_s} \; L_{IL} = \mathbb{E}_{s,\, o\sim D} \left\| p_s(\cdot|o) - \pi_t(\cdot|s) \right\|_2, \quad (3)$$

where D denotes the replay buffer and the optimal p_s represents the student policy in the next iteration. Alternatively, one may minimize the KL divergence between the two:

$$\min_{p_s} \; L_{KL} = \mathbb{E}_{s,\, \sim D} \, \text{KL}\left(p_s(\cdot|o) \parallel \pi_t(\cdot|s)\right), \quad (4)$$

or any statistical distance metric that fits the action space.

Besides the imitation loss, the student can be updated using an asymmetric learning approach [13] that leverages the teacher's critic, i.e., the value functions:

$$\min_{p_s} L_{\text{Asym}} = -\beta \langle Q^{\pi_t}(s,\cdot), p_s(\cdot|o)\rangle + \text{KL}(p_s \| \pi_t). \quad (5)$$

We denote the general loss function for student agents as L_s. In our application on the Digit robot, we observed that using the L_2 imitation loss yields the best performance, while the addition of L_{Asym} does not affect the overall performance by a large margin. We conjecture that the L_2 loss allows the student policy to have a slightly higher exploration capability as we observe that $L_{\text{Asym}} + L_2$ will reach a training plateau that is inferior in performance than using L_2 alone.

Algorithm 1 Learn to Teach - RL (L2T-RL)

1: **Input:** initial teacher policy π_t^0, student policy π_s^0, and step size sequences $\{\beta_t^k\}$ and $\{\beta_s^k\}$.
2: **for** $k = 0$ **to** K **do**
3: Sample a mini-batch D_k from replay buffer D.
4: Update the teacher critic by estimating:

$$Q^{\pi_t^{k+1}}(s,a), V^{\pi_t^{k+1}}(s).$$

5: Update the teacher policy:

$$\pi_t^{k+1} = \arg\min_{p_t} -\beta_t^k \langle Q^{\pi_t^k}(s,\cdot), p_t(\cdot|s)\rangle + \text{KL}(\pi_t^k \| p_t).$$

6: Update the student policy:

$$\pi_s^{k+1} = \arg\min_{\pi_s} L_s(\pi_s^k).$$

7: Roll out to collect new samples D' according to the scheduling in Eq. 1.
8: Update the replay buffer: $D \leftarrow D \cup D'$.
9: **end for**

IV. Environment Design

The Digit robot is a bipedal walking robot with 30 degrees of freedom, which includes 20 actuated joints with 4 per arm and 6 per leg. All joints are revolute joints except for the shin and heel joints, which are spring-based. Notably, the Digit robot features three closed kinematic chains per leg. Two of these chains involve motors controlling the foot, assisted by additional rods, while the third chain is responsible for controlling the heel via a rod extending from the hip. This leg design makes it a significantly challenging task for RL algorithms due to the high-dimensional action space and the complex dynamics of the robot.

We highlight a significant portion of our work is to reconstruct a faithful Universal Scene Description (USD) model of the robot in IsaacLab [9], although this is not claimed as an algorithmic contribution. As a result, we build a velocity-tracking RL task with accurate dynamics w.r.t the robot hardware, facilitating the sim2real transfer.

A. Observation space

The observation space (see table I) is constructed using data provided by the robot's sensors, including base linear velocity, base angular velocity, joint positions, and joint velocities. Additionally, we include the commanded velocity that the robot will receive from an external controller during execution and the computed projected gravity and gait phase

based on the IMU data and the robot execution time. Finally, the actions in a previous time step are also incorporated into the observation space, which allows us to learn a history-dependent policy using recurrent neural nets. We model measurement noise as $o_t = s_t + \alpha\epsilon$ where α is the scale, and ϵ is either Gaussian or uniform noise. We add this noise to the student's observation space to mimic the hardware sensors while keeping the teacher's observation noise-free, except for the ones that incorporate the student's observations in order to alleviate the imitation gap, which is considered a common practice. Additionally, privileged information (lower half of Table I) is provided for the teacher for easier training.

TABLE I: Observation Terms for Teacher and Student Agents

Observation Terms	Dim	Noise	Student π_s	Teacher π_t
Clock	2		✓	✓
Base lin. vel.	3	✓	✓	✓
Base ang. vel.	3	✓	✓	✓
Projected gravity	3	✓	✓	✓
Velocity command	3		✓	✓
Joint pos.	30	✓	✓	✓
Joint vel.	30	✓	✓	✓
Last action	20		✓	✓
Base lin. vel.	3			✓
Base ang. vel.	3			✓
Joint position	48			✓
Joint velocity	48			✓
Root state (w)	7			✓
Base lin. vel. (w)	3			✓
Base ang. vel. (w)	3			✓
Base pos. (w)	3			✓
Base quant. (w)	4			✓
Env params	316			✓
Height scan	187			✓

B. Action space

The action space is designed as the target full-body joint positions q_{target}, which a Proportional Derivative (PD) controller will aim to track during execution. At a frequency of 50 Hz, the policy predicts the current targeted joint based on the current observation, and then at a higher frequency (1 kHz), the PD controller computes the torque τ as inputs to the motors to control the robot's joints. The target velocity is set to zero, which is commonly employed in legged robot research. The PD gains are determined through empirical tuning to ensure stable joint control. We use a standard PD control law for computing the torque, i.e, $\tau = \mathrm{K}_p(q_{\text{target}} - q) + \mathrm{K}_d(\dot{q}_{\text{target}} - \dot{q})$, where q represents the measured joint positions and $\dot{q}$ represents the measured joint velocities.

C. Reward functions

Our reward function design is summarized in Table II. We adopt some of the existing reward functions in IsaacLab across other velocity command tasks for bipeds and quadrupeds, including termination penalty, action rate, joint deviation, etc. In addition, we design specific reward functions for training the Digit robot, for which we highlight two of them. For implementation details, please refer to our code.

Track Foot Height: We reward the agent for following a desired foot height trajectory, which is precomputed as a quintic polynomial.

TABLE II: Reward Functions and Their Weights

Reward	Weight	Reward	Weight
Termination penalty	-200.0	Foot contact	2.0
Being alive	0.01	Track foot height	0.5
Action rate	-0.015	Foot clearance	0.5
DOF velocity	-5e-4	Track lin vel XY	0.5
Undesired contacts	-1.0	Track ang vel Z	1.0
Flat orientation	-10.0	Lin vel XY	-2.0
Feet air time	0.25	Ang vel Z	-0.1
Feet sliding	-1.0	DOF torques	-1.0e-5
DOF pos limits	-0.5	DOF acc	-2.5e-7
Joint deviation hip	-5.0	Feet air time	0.125
Joint deviation arms	-0.3	Joint deviation toes	-0.1

$$r_{\text{foot-track}} = \exp\{-\|h_{\text{foot\_traj}} - h_{\text{foot\_traj\_target}}\|_2\}, \quad (6)$$

$h_{\text{foot\_traj}}$ is the actual foot height and $h_{\text{foot\_traj\_target}}$ is the target foot height. We adjust the desired foot height based on the current CoM position to adapt to uneven terrains. Specifically, instead of tracking the absolute height of the foot, we track the relative distance of the foot and the CoM position to compensate for the terrain height, which is hard to obtain at run time. This reward seems to be the most important reward term, without which we are unable to train an agent robust to challenging terrains.

Foot Contact Matching:

$$r_{\text{contact}} = \begin{cases} c_1, & \text{sign}(\phi(t)) = \text{sign}(F_{\text{GRF}} > 0) \\ -c_2, & \text{otherwise.} \end{cases} \quad (7)$$

where $\phi(t) = \sin(2\pi t/h)$, $h = 0.68$ is the gait cycle duration, $c_1, c_2 \in \mathbb{R}^+$ are constants. We reward the agent when the foot contact matches the desired gait cycle. For example, if $\phi(t) > 0$ (indicating the foot should be in contact with the ground) and $F_{\text{GRF}} > 0$ (indicating the foot is actually in contact), we assign a positive reward c_1. Otherwise, a penalty $-c_2$ is applied.

TABLE III: Event Terms for Domain Randomization

Name	Name
Rand. friction coeff	Add base mass
Rand. gravity	Add external force
Rand. base location	Push robot
Rand. robot joints	

D. Domain randomization and curriculum

We dynamically change the observation space and perturb the physical dynamics of the environment in the hope of capturing the randomness and variation of the real world. We list all the domain randomization schemes used in the training in Table. III. In practice, we find that randomizing friction coefficients and adding external pushes can greatly improve the robustness of the trained policy.

Additionally, we adopt the curriculum training setup implemented by IsaacLab. We utilize the existing terrain map, which includes seven different terrains: flat ground, slopes, stepping stones, and pyramid stairs up and down.

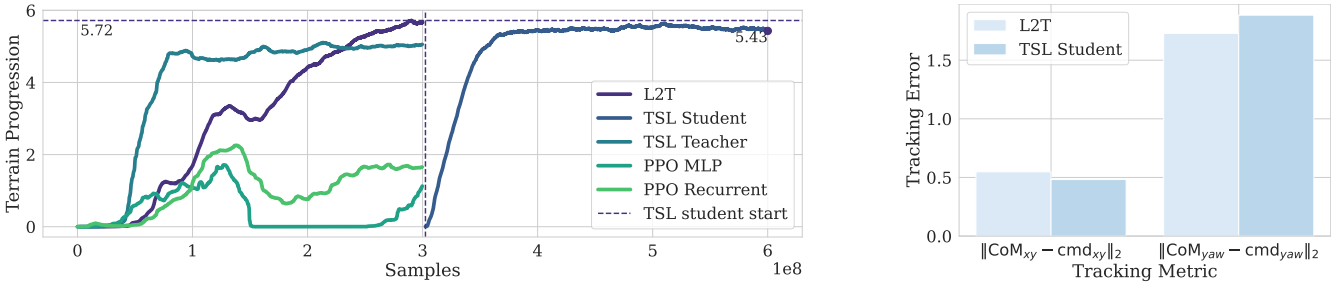

Fig. 3: Training curve against baselines. (a) We plot the difficulty of the terrain the agent can successfully traverse during training. (b) We train and evaluate each method on the flat ground and plot the velocity tracking of each agent.

V. Computational Results

A. Rough terrain difficulty progression

First, we show the progression of terrain difficulty (curriculum progression) throughout training, representing the overall task completion, i.e., the level of terrain difficulty the agent can walk over with the CoM velocity maintained within a specific range from the velocity command. This metric is calculated as the average difficulty level across the 6 terrain setups mentioned earlier. In essence, faster learning corresponds to a steeper curve in terrain difficulty progression. Fig. 3(a) illustrates the terrain progression during training. The purple curve illustrates the training curve of L2T, reaching a maximum of 5.72 level of difficulty. Compared to a two-stage traditional teacher-student learning (TSL) paradigm, which is represented by two curves: the teal curve represents the TSL teacher agent, and the blue curve represents the TSL student agent. Notice that in TSL, the student agent can only be trained after the teacher. The TSL teacher has an MLP-based policy, while the student is LSTM-based, sharing the same network architecture and hyperparameter as in L2T. It is noteworthy to mention that we can only obtain a reasonable training result by using data aggregation (DAGGER) [35], i.e., when training the student, periodically use the teacher to predict the action. Additionally, we include two RL methods for learning with student observation space using vanilla PPO, and recurrent PPO. It turns out that these two baselines perform significantly inferior. Since IsaacLab cannot be evaluated during training, the L2T curves represent the joint training progression between the L2T teacher and the student. One might think that the true performance of L2T students can be much worse than what the curve shows. However, as we observe that the L2T curve eventually goes higher than the TSL teacher, we argue that this curve is rather the performance lower bound for the L2T student.

B. Flat terrain velocity tracking error

Next, we investigate the agent velocity tracking performance. Unfortunately velocity tracking is not a valid metric on general terrain due to the variability of the terrain difficulty. For example, the same agent might exhibit significantly different velocity tracking abilities either on slopes

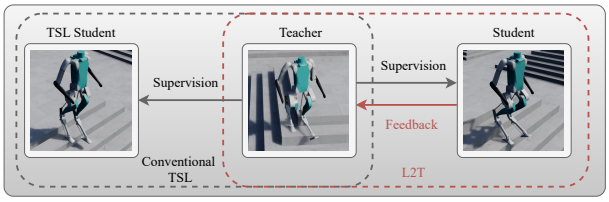

Fig. 4: **Left:** in conventional TSL, the teacher supervises student. **Right:** L2T co-trains both agents together, where the student provides feedback to the teacher through mixed samples.

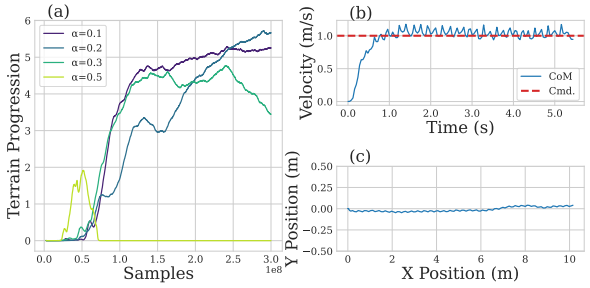

Fig. 5: (a) Ablation study on the mixture coefficient α_{mix}. (b) CoM velocity versus commanded velocity. (c) xy trajectory with x direction command.

or stairs. To make a fair comparison, we re-train separate agents and collect the results on flat terrain, as it presents the most straightforward task for locomotion. We compare the velocity tracking between the CoM velocity and the commanded velocity on the xy plane and the yaw angle. The results are plotted in Fig. 3(b). The L2T agent and the student from the traditional teacher-student learning (TSL) paradigm are comparable, with L2T better at yaw tracking and worse at tracking in the xy plane. This shows that on the flat surface, L2T performs similarly to traditional teacher-student training paradigms but saves 50% of the total samples. It is noteworthy to mention that we use the same network architecture and hyperparameters for both training frameworks. In Fig. 5(b)-(c), we also plot the recorded robot CoM sagittal velocity versus the commanded velocity and the robot's CoM trajectory in the horizontal plane.

C. Mitigation of policy divergence

We observe that mixing samples can alleviate the imitation gap, which is caused by the teacher having access to privileged information that is unavailable to the student. This privileged information is marginalized during imitation learning for the student agent, resulting in the student agent requiring more exploration and acting more conservatively. For example, since the student does not know if it is at the edge of the stairs, they will act less confidently when walking downstairs. However, a trained expert teacher agent might act more confidently as it knows the structure of the stairs due to the privileged information of a local depth map. This is generally true due to the existence of observational noise. In the presence of the imitation gap, the teacher agent might generate desirable demonstrations only for the teacher itself, but not necessarily desirable ones for the student agent [10].

In traditional TSL paradigms, this problem can be partially addressed by DAGGER or asymmetric learning. In L2T, all these techniques are naturally blended together as a single-stage co-training process, providing an efficient and elegant solution to address the imitation gap and policy divergence issue. We show that in Fig. 4, the walking gait of the student trained by DAGGER is significantly different from the teacher policy, while L2T can faithfully imitate the teacher. Notice the shape of the toe pad from the L2T student, which slightly tilts up from the horizontal plane, accurately mimicking the teacher's toe. In contrast, the DAGGER-trained student agent has a flat-ground walking gait, with the toe pad parallel to the horizontal plane.

D. Ablation study on the mixture coefficient

We conduct an ablation study on the critical components to further explore key design choices within our algorithm framework. Fig. 5(a) demonstrates the importance of the mixture coefficient α_{mix} in the training process. We observe that while the algorithm becomes unstable with a large α_{mix}, which is caused by letting a highly suboptimal student agent inject too many samples, an appropriate chosen α_{mix} can benefit the overall training process. We hypothesize that the discrepancies between the student and teacher promote exploration within the action space, enabling both agents to learn from a broader region around the teacher's actions.

VI. Hardware Experiments

A. Locomotion over challenging terrains

We report the results of deploying our policy in various outdoor environments, as illustrated in Fig. 1 and the supplementary video. These experiments were conducted around a university campus, including pedestrian walkways, wooden bridges, grass-covered hills, beach volleyball courts, and gravel paths. Extensive outdoor testing highlights the policy's capability to maintain a reliable and natural walking gait and withstand diverse environmental conditions, such as walking on wet surfaces and operating with strong wind.

Surprisingly, the policy shows generalization ability to scenarios not included in the training. For example, on grass hills, due to rain, the grass and the soil underneath exhibit a certain level of deformation upon impact, which increases the difficulty of state estimation and thus further increases the observation noise. Moreover, the policy can walk on the beach volleyball court with sandy terrain, as shown in Fig. 1. Despite the robot not being calibrated or trained for such conditions, the policy adapts to the environmental changes without additional training. This demonstrates the robustness and adaptability of our approach. Next, we examine the policy's performance on terrains with obstacles. The first test involves a crate of gravel shown in Fig. 6(b). The robot consistently performs stepping-in-place actions with a stable motion. Even when the robot occasionally strikes the crate's edge, it recovers and resumes normal a stepping gait. In the second test, we deploy the robot on a slippery terrain, where we distribute poppy seeds on a whiteboard shown in Fig. 6(c). Surprisingly, our robot does not exhibit any perceptible shaking motions. In contrast, the company controller provided by Agility Robotics fails to walk over. Furthermore, the policy enables omnidirectional control of the robot. Experiments with velocity commands demonstrate the robot's directional control, which is crucial for practical deployment. Readers can find these experiments in the video.

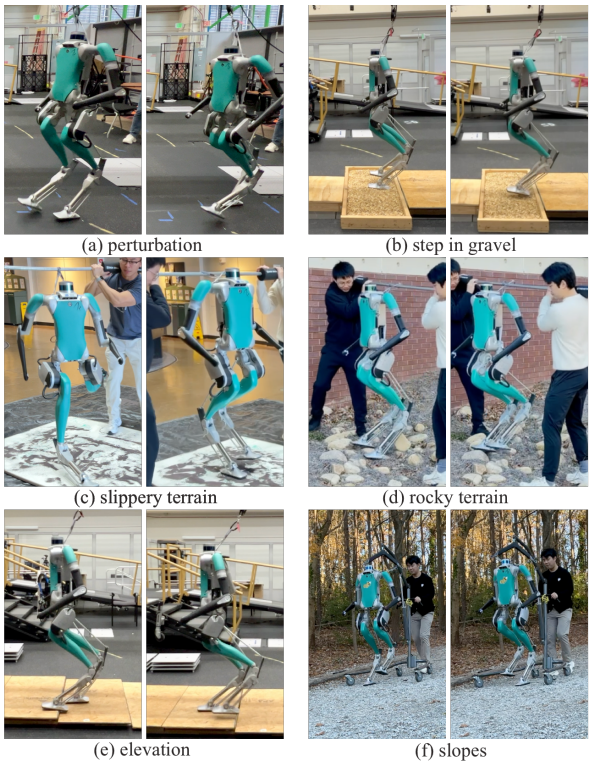

Fig. 6: (a) we perturb the robot with a harness. (b) we test the step-in-place motion on a crate of gravel. We let the robot walk over (c) a slippery terrain with (left) the Agility company controller and (right) our L2T controller. (d) rocky terrain with slopes. (e) elevated platforms. (f) hill with gravel.

B. Perturbation experiments

We evaluate the policy's response to external perturbations applied during locomotion. Two scenarios are considered: pushing the robot's center of mass (CoM) from the front,

and back using a stick. The policy demonstrates robustness to withstand frontal and rearward pushes while maintaining a stable walking gait. We could not push the robot to failure even when applying a large force. Additionally, we apply a more substantial perturbation by using a harness to pull the robot with an impulsive force (see Fig. 6(a)). The robot adapts dynamically, exhibiting agile adjustments to its gait to compensate for the external perturbations.

VII. Conclusion

We introduced L2T-RL, a novel single-stage learning framework that unifies teacher and student training to address sample inefficiency and enhance real-world performance. Our extensive simulation and hardware experiments demonstrate that L2T-RL achieves robust, agile, and precise locomotion while reducing sample complexity by 50% and dramatically saving training time. These results highlight our contributions to redefining teacher-student learning paradigms and paving the way for practical RL-based robotic systems.

References

[1] J. Siekmann, K. Green, J. Warila, A. Fern, and J. Hurst, "Blind bipedal stair traversal via sim-to-real reinforcement learning," in *Robotics: Science and Systems*, 2021.

[2] Z. Li, X. B. Peng, P. Abbeel, S. Levine, G. Berseth, and K. Sreenath, "Robust and versatile bipedal jumping control through reinforcement learning," 2023.

[3] T. Haarnoja, B. Moran, G. Lever, S. H. Huang, D. Tirumala, M. Wulfmeier, J. Humplik, S. Tunyasuvunakool, N. Y. Siegel, R. Hafner, *et al.*, "Learning agile soccer skills for a bipedal robot with deep reinforcement learning," *preprint arXiv:2304.13653*, 2023.

[4] J. Kober, J. A. Bagnell, and J. Peters, "Reinforcement learning in robotics: A survey," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1238–1274, 2013.

[5] X. B. Peng, M. Andrychowicz, W. Zaremba, and P. Abbeel, "Sim-to-real transfer of robotic control with dynamics randomization," in *2018 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2018, pp. 3803–3810.

[6] D. Chen, B. Zhou, V. Koltun, and P. Krähenbühl, "Learning by cheating," in *Conference on Robot Learning*. PMLR, 2020, pp. 66–75.

[7] 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.

[8] T. Miki, J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter, "Learning robust perceptive locomotion for quadrupedal robots in the wild," *Science Robotics*, vol. 7, no. 62, p. eabk2822, 2022.

[9] M. Mittal, C. Yu, Q. Yu, J. Liu, N. Rudin, D. Hoeller, J. L. Yuan, R. Singh, Y. Guo, H. Mazhar, A. Mandlekar, B. Babich, G. State, M. Hutter, and A. Garg, "Orbit: A unified simulation framework for interactive robot learning environments," *IEEE Robotics and Automation Letters*, vol. 8, no. 6, pp. 3740–3747, 2023.

[10] L. Weihs, U. Jain, I.-J. Liu, J. Salvador, S. Lazebnik, A. Kembhavi, and A. Schwing, "Bridging the imitation gap by adaptive insubordination," *Advances in Neural Information Processing Systems*, vol. 34, pp. 19 134–19 146, 2021.

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

[12] T. Ni, B. Eysenbach, and R. Salakhutdinov, "Recurrent model-free rl can be a strong baseline for many pomdps," *arXiv preprint arXiv:2110.05038*, 2021.

[13] L. Pinto, M. Andrychowicz, P. Welinder, W. Zaremba, and P. Abbeel, "Asymmetric actor critic for image-based robot learning," *arXiv preprint arXiv:1710.06542*, 2017.

[14] Y. Ma, F. Farshidian, and M. Hutter, "Learning arm-assisted fall damage reduction and recovery for legged mobile manipulators," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 12 149–12 155.

[15] A. O'Neill, A. Rehman, A. Maddukuri, A. Gupta, A. Padalkar, A. Lee, A. Pooley, A. Gupta, A. Mandlekar, A. Jain, *et al.*, "Open x-embodiment: Robotic learning datasets and rt-x models: Open x-embodiment collaboration 0," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 6892–6903.

[16] Y. Ze, Z. Chen, W. Wang, T. Chen, X. He, Y. Yuan, X. B. Peng, and J. Wu, "Generalizable humanoid manipulation with improved 3d diffusion policies," *arXiv preprint arXiv:2410.10803*, 2024.

[17] X. Cheng, Y. Ji, J. Chen, R. Yang, G. Yang, and X. Wang, "Expressive whole-body control for humanoid robots," *arXiv preprint arXiv:2402.16796*, 2024.

[18] Z. Gu, J. Li, W. Shen, W. Yu, Z. Xie, S. McCrory, X. Cheng, A. Shamsah, R. Griffin, C. K. Liu, *et al.*, "Humanoid locomotion and manipulation: Current progress and challenges in control, planning, and learning," *arXiv preprint arXiv:2501.02116*, 2025.

[19] Q. Ben, F. Jia, J. Zeng, J. Dong, D. Lin, and J. Pang, "Homie: Humanoid loco-manipulation with isomorphic exoskeleton cockpit," *arXiv preprint arXiv:2502.13013*, 2025.

[20] F. Wu, J. Ke, and A. Wu, "Inverse reinforcement learning with the average reward criterion," *arXiv preprint arXiv:2305.14608*, 2023.

[21] Z. Chen, X. He, Y.-J. Wang, Q. Liao, Y. Ze, Z. Li, S. S. Sastry, J. Wu, K. Sreenath, S. Gupta, *et al.*, "Learning smooth humanoid locomotion through lipschitz-constrained policies," *arXiv preprint arXiv:2410.11825*, 2024.

[22] P. Dugar, A. Shrestha, F. Yu, B. van Marum, and A. Fern, "Learning multi-modal whole-body control for real-world humanoid robots," *arXiv preprint arXiv:2408.07295*, 2024.

[23] X. Gu, Y.-J. Wang, and J. Chen, "Humanoid-gym: Reinforcement learning for humanoid robot with zero-shot sim2real transfer," *arXiv preprint arXiv:2404.05695*, 2024.

[24] J. Long, J. Ren, M. Shi, Z. Wang, T. Huang, P. Luo, and J. Pang, "Learning humanoid locomotion with perceptive internal model," *arXiv preprint arXiv:2411.14386*, 2024.

[25] F. Wu, Z. Gu, H. Wu, A. Wu, and Y. Zhao, "Infer and adapt: Bipedal locomotion reward learning from demonstrations via inverse reinforcement learning," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 16 243–16 250.

[26] I. Radosavovic, T. Xiao, B. Zhang, T. Darrell, J. Malik, and K. Sreenath, "Real-world humanoid locomotion with reinforcement learning," *Science Robotics*, vol. 9, no. 89, p. eadi9579, 2024.

[27] F. Liu, Z. Gu, Y. Cai, Z. Zhou, S. Zhao, H. Jung, S. Ha, Y. Chen, D. Xu, and Y. Zhao, "Opt2skill: Imitating dynamically-feasible whole-body trajectories for versatile humanoid loco-manipulation," *arXiv preprint arXiv:2409.20514*, 2024.

[28] I. Radosavovic, B. Zhang, B. Shi, J. Rajasegaran, S. Kamat, T. Darrell, K. Sreenath, and J. Malik, "Humanoid locomotion as next token prediction," *arXiv preprint arXiv:2402.19469*, 2024.

[29] J. Dao, H. Duan, and A. Fern, "Sim-to-real learning for humanoid box loco-manipulation," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 16 930–16 936.

[30] I. Radosavovic, S. Kamat, T. Darrell, and J. Malik, "Learning humanoid locomotion over challenging terrain," *arXiv preprint arXiv:2410.03654*, 2024.

[31] L.-J. Lin, "Self-improving reactive agents based on reinforcement learning, planning and teaching," *Machine learning*, vol. 8, pp. 293–321, 1992.

[32] G. Lan, "Policy mirror descent for reinforcement learning: Linear convergence, new sampling complexity, and generalized problem classes," *Mathematical programming*, vol. 198, no. 1, pp. 1059–1106, 2023.

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

[34] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *International conference on machine learning*. PMLR, 2018, pp. 1861–1870.

[35] S. Ross, G. Gordon, and D. Bagnell, "A reduction of imitation learning and structured prediction to no-regret online learning," in *Proceedings of the International Conference on Artificial Intelligence and Statistics*, 2011, pp. 627–635.