Unified Locomotion Transformer with Simultaneous Sim-to-Real Transfer for Quadrupeds

Dikai Liu[1,2] Tianwei Zhang[2] Jianxiong Yin[1] Simon See[1,3]

***Abstract*— Quadrupeds have gained rapid advancement in their capability of traversing across complex terrains. The adoption of deep Reinforcement Learning (RL), transformers and various knowledge transfer techniques can greatly reduce the sim-to-real gap. However, the classical teacher-student framework commonly used in existing locomotion policies requires a pre-trained teacher and leverages the privilege information to guide the student policy. With the implementation of large-scale models in robotics controllers, especially transformers-based ones, this knowledge distillation technique starts to show its weakness in efficiency, due to the requirement of multiple supervised stages. In this paper, we propose Unified Locomotion Transformer (`ULT`), a new transformer-based framework to unify the processes of knowledge transfer and policy optimization in a single network while still taking advantage of privilege information. The policies are optimized with reinforcement learning, next state-action prediction, and action imitation, all in just one training stage, to achieve zero-shot deployment. Evaluation results demonstrate that with `ULT`, optimal teacher and student policies can be obtained at the same time, greatly easing the difficulty in knowledge transfer, even with complex transformer-based models.**

I. Introduction

Driven by the advancement of deep reinforcement learning (RL), quadruped robots have drawn great attention, due to their capability of traversing across complex terrains [1]–[4]. Traditional robotics controllers rely on dedicated models and heuristics, which require extensive prior knowledge and can struggle to adapt to dynamic environments and unpredictable situations. Recently, a new paradigm is introduced to push quadrupeds' limit to handle complex tasks in challenging environments [3]–[8]: a controller is learned from simulations that contain various environmental and physical factors, and then transferred to the physical robot through RL and various knowledge transfer techniques [3], [4], [9]–[11] to overcome the sim-to-real gap using privilege information.

One of the commonly used knowledge distillation methods is the teacher-student framework [3]. A teacher policy is first trained through RL, and privilege information about the environment is provided for learning the optimal locomotion strategies efficiently. As additional information, including ground-truth dynamics and terrain profiles, is often inaccessible in the real world, a student policy is subsequently trained to make the model deployable. This is typically achieved through supervised knowledge distillation, either offline [3] or online [4] fashion by creating a data set with new trajectories and associated teacher action labels through algorithms such as the Data Aggregator (DAgger) [12].

Sequential training with policy imitation for locomotion control is generally not data-efficient [13]. The performance of the student is limited by the teacher policy and the robustness depends on the diversity of the data set. Generally, they will deteriorate if the situation deviates from the trajectories in real-world deployment. To address this, DreamWaQ [14] was introduced with the asymmetric actor-critic architecture [15] and context estimation, including next observation, base velocity and latent space. By concurrently training with proprioceptive observation for exploration and privileged information as the critic, it allows the agent to explore with indirect guidance.

Recently, self-attention-based transformers [16] have been widely introduced into robotics in both lower control (e.g., legged locomotion directly [10], [11], [17], [18] or with a command interface [19]) and high-level decision making with multimodal processes [20]–[23] with their capability to handle variable context lengths, sensor combinations [18] and even robot embodiments [24]. In direct locomotion controllers, transformers have demonstrated superior capability in temporal information capture compared to recurring neural networks (RNN) and temporal convolutional networks (TCN). However, training transformers is data-hungry in general. For example, when combing the vanilla transformer with legged locomotion, Lai et al. [10] generated an additional 40M timesteps for 400K updates using a two-stage supervised training approach. Similarly, Radosavovic et al. [11] doubled the number of timesteps for joint supervised transfer compared to the teacher policy training stage. These approaches are not only time-consuming, but also necessitating additional setup to handle multiple models and the large volumes of generated data.

Motivated by the above limitation, the objective of this paper is to simplify the knowledge distillation process with a unified architecture to keep teacher and student policies in a single network for simultaneous optimization. Our inspiration comes from the exceptional capabilities of transformers in multimodal modeling of temporal and sensory information and context understanding [17], which can well support policy optimization [14]. Thus, we can introduce the privilege information into the observation as another modality to form a unified framework for single-phase optimization of teacher and student policies, achieving zero-shot sim-real transfer.

To this end, we propose Unified Locomotion Transformer (`ULT`), a new unified framework for end-to-end quadruped

[1] NVIDIA AI Technology Centre (NVAITC); e-mail: {dikail,jianxiongy,ssee}@nvidia.com

[2] College of Computing and Data Science, Nanyang Technological University, Singapore; e-mail: dikai001@e.ntu.edu.sg, tianwei.zhang@ntu.edu.sg

[3] also with Nanyang Technological University and Coventry University

locomotion. It is based on the standard transformer architecture with casual masking to pack teacher and student policies in a single network. These policies are optimized jointly with reinforcement learning, next state-action prediction, and action imitation, all in just one training phase, to overcome the sim-to-real gap and achieve zero-shot deployment. In this way, we eliminate the need for dedicated design and training of a teacher network, while the privilege information can still efficiently guide the student policy during the exploration to generate more diverse trajectories to improve the overall generalization and robustness.

We extensively evaluate our framework in simulation and compare it with state-of-the-art knowledge transfer baselines. We also deploy it directly in the real world for practicality validation. Evaluation results demonstrate that ULT exhibits better performance with less trajectory information needed, indicating its higher efficiency with the help from next state-action predication, action imitation and mixed exploration. With unified training in one single phase for simultaneous teacher and student policy optimization, we ease the pipeline of knowledge distillation for zero-shot sim-to-real transfer.

II. Related Work

A. Knowledge Transfer in RL-based Legged Locomotion

With the increased attention in Reinforcement Learning (RL) and the advance of robotics simulation, the RL-based simulation-first controller has been dominating legged locomotion [1]–[4], [10], eliminating the need for extensive prior knowledge and reducing the cost of training by massive parallelism [25]. To transfer the policy from simulation to the physical world, Yu et al. [9] used online system identification to infer physics parameters in the real world. To further advance the locomotion policy for optimized control and close the sim-to-real gap, Lee et al. [3] introduced a teacher-student framework for action cloning, which uses historical proprioceptive data to infer teacher behaviors leveraging the privilege information. This framework has been widely adopted in subsequent works. To improve the trajectory generation during transfer, Kumar et al. [4] used a randomly initialized student policy for online adaptation with better exploration to improve the robustness. The introduction of Transformer-based controllers in legged locomotion makes the optimization more challenging. Lai et al. [10] introduced a two-stage transfer to ensure that the student policy can gather useful trajectories during online correlation for fast convergence. Radosavovic et al. [11] combined RL exploration and teacher supervision to jointly optimize the student.

B. Transformer in Legged Locomotion

Transformer-based models have recently made significant inroads into robotics with their exceptional capabilities for in-context understanding and human-robot interaction. They are equipped with Large Vision Language Models (VLM) [21] and Vision Language Action Models (VLA) [22], [23], [26], to handle the multimodal input from humans and physical world.

Constrained by the computational capability and power consumption of real-time inference on the edge devices, the transformer used for legged locomotion is often much smaller and sometimes a high-level command is adopted as the control interface. Yang et al. [27] developed a transformer model for vision-based locomotion, which outputs desired velocity commands for a dedicated low-level controller to conduct motor output. Similarly, Tang et al. [19] used the gait pattern as the command interface for quadruped locomotion. With knowledge transfer techniques, Lai et al. [10] and Radosavovic et al. [11] utilized historical observation-action information for direct motor command in quadruped and bipedal locomotion, respectively. Radosavovic et al. [17] further expanded the framework with next token prediction to utilize different data source including captured real-world locomotion trajectories.

To improve the generalization of transformer-based locomotion controllers, different masking strategies are used in recent works to partially remove information during training and testing so the controller can learn to focus on the most important information. Sferrazza et al. [24] used body-induced bias based on the embodiment graph for an embodiment-aware transformer. Liu et al. [18] introduced masking directly at the sensor level for generalization with different sensor combinations as input. Transformer-based controllers can also achieve cross-embodiment policy to conduct different types of tasks on various robots with one single network [28], [29]

III. Simulation Environment

In this paper, we implement the simulation environment in Isaac Gym [30] and IsaacGymEnvs [30] to train locomotion agents in large scale parallelism. All baselines and variants of our method are trained with the exact same simulation setups to ensure fair comparison.

A. Terrain and Curriculum

To ensure that the policy is robust against different indoor and outdoor environments, we adopt the terrain curriculum from [25] with smooth slope, rough slope, stairs up, stairs down and discrete obstacle terrains. Each type of terrain has 10 levels with incremental difficulty and an overall proportion of $[0.1, 0.1, 0.35, 0.25, 0.2]$, respectively. The linear velocity return is tracked across each trajectory's life cycle. The level is considered solved when an agent reaches 80% of the maximum achievable tracking reward and progresses to the next level. If any agent fails to reach 25% of the maximum reward, it will regress to a lower level.

B. Domain Randomization

Following [14], [25], we apply Domain Randomization (DR) on key dynamics parameters to enhance the robustness of the policy. To simulate the actual user commands, we sample the linear commands in longitudinal and lateral direction separately with a uniform distribution in $[-0.5, 0.5]$ m/s. For the angular command, we first sample the desired heading of the robot and cap the resulted angular velocity

TABLE I

SIMULATION PARAMETERS OF DOMAIN RANDOMIZATION.

Parameters	Range	Unit
Linear Command	[-0.5, 0.5]	m/s
Angular Heading	[-3.14, 3.14]	rad
K_p Scale	[0.9, 1.1]	-
K_d Scale	[0.9, 1.1]	-
Friction Scale	[0.7, 1.3]	-
Motor Strength Scale	[0.9. 1.1]	-
Payload	[0, 5]	kg
Payload CoM Offset	[-0.1, 0.1]	m
External Push	[-1, 1]	m/s
Gravity	[9,41, 10.21]	m/s^2
System Delay	[0, 0.015]	s

command at 0.5 rad/s. The commands are resampled every 10 seconds. Table I lists the key parameters of DR used in the simulation environment.

C. Observations and Actions

Privilege Information. To achieve an optimal locomotion policy with the hidden information about the environment, related privilege data is extracted from simulation to form the privilege observation e_t for the teacher policy to utilize. The privilege information contains randomized dynamics parameters d_t sampled from Sec. III-B, ground truth robot states s_t including base velocity, orientation, and precise surrounding height map m_t. Although such information is often inaccessible in the real world, it helps the teacher policy reconstruct states and improve learning efficiency [3], [4].

Proprioceptive Observation. In order to conduct knowledge transfer for a deployable agent, the student policy only relies on onboard sensors to provide observations. Typically, quadrupedal robots are equipped with multiple sensors, including joint encoders, IMU, and foot contact sensors, which can provide information on joint position $q \in \mathbb{R}^{12}$, joint velocity $\dot{q} \in \mathbb{R}^{12}$, angular velocity $\omega \in \mathbb{R}^3$, gravity vector $g \in \mathbb{R}^3$ and binary foot contact $c \in \mathbb{R}^4$. In order to follow the user command, the agent also needs access to the randomly sampled $\mathrm{cmd} = [v_x, v_y, \omega_z] \in \mathbb{R}^3$ to form the observation of each step $o_t = [q, \dot{q}, \omega, g, c, \mathrm{cmd}] \in \mathbb{R}^{37}$. To provide the state transition and temporal information, the actions of the previous step $a_{t-1} \in \mathbb{R}^{12}$ are added with a list of historical information $\mathcal{T} = [a_0, o_1, a_1, o_2, \cdots, a_{t-1}, o_t]$. We use a rolling window of $t = 15$, resulting in a full observation in the space of $\mathbb{R}^{49\times15}$.

Actions. As typical RL locomotion policies perform inference at the frequency of 50-100 Hz, both the teacher and student policies output the desired joint position $q = a_t$, which is passed to a PD controller running at a much higher frequency for a smooth torque output:

$$\tau = K_p(\hat{q} - q) + K_d(\hat{\dot{q}} - \dot{q}) \tag{1}$$

with base stiffness K_p and damping K_d set to 30 and 0.7, respectively, and additional DR is added on. The target joint velocity $\hat{\dot{q}}$ is set to 0.

TABLE II

REWARD TERMS FOR REINFORCEMENT LEARNING

Reward	Definition	Scale
Linear Velocity Tracking	$\exp\left(-5\|v_{xy}^{\mathrm{cmd}} - v_{xy}\|^2\right)$	1.0
Angular Velocity Tracking	$\exp\left(-5(\omega_z^{\mathrm{cmd}} - \omega_z)^2\right)$	0.5
Body Z Velocity	$\|v_z\|^2$	-2.0
Body Rotation	$\|\omega_z\|^2$	-0.05
Joint Acceleration	$\|\ddot{\theta}\|^2$	-2.5e-7
Output Work	$\int \|\tau \cdot \dot{\theta}\|$	-2.e-5
Action Rate	$(a_t - a_{t-1})^2$	-0.05
Feet Slip	$\|g_t \cdot v_{xy}^{feet}\|$	-0.1
Collision	$\mathbb{1}_{collision}$	-1

D. Reward Function

We follow the classic reward function design for omni-direction locomotion [4], [14], [25] to encourage the agent to follow the commanded velocity and primarily penalize the linear and angular movement along other axes, large joint acceleration and excessive power consumption. The complete reward structure is detailed in Table II.

IV. METHODOLOGY

We present ULT, a transformer-based framework to unify the process of knowledge transfer and policy optimization in a single network leveraging the privilege information. Compared to the classic teacher-student transfer solution that requires a pre-trained teacher policy [3], [4], [10], ULT optimizes both policies jointly in a single phase to simplify the sim-to-real pipeline and reduce the total number of generated trajectories. Figure 1 shows an overview of ULT.

RL-based quadrupedal locomotion is often formulated as a Partially Observable Markov Decision Process (POMDP), defined by a tuple $\mathcal{M} = (\mathcal{S}, \mathcal{A}, T, \mathcal{R}, \Omega, \mathcal{O}, \gamma)$, where $\mathcal{S}, \mathcal{A}, \mathcal{R}, \Omega$ are the state, action, reward and observation spaces, respectively. The ground truth states $s_t \in \mathcal{S}$ give the most important information about the environment and can be accessed by the teacher policy in the simulation to search for an optimal control policy $\pi^*(a_{t+1}|s_t)$ by maximizing the sum of discounted future rewards:

$$\pi^*(s, a) := \arg\max_{\pi} \mathbb{E}_{s_{t+1}\sim T(\cdot|s_t, a_t)}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)\right] \tag{2}$$

However, such information is often inaccessible in the real world, and the student policy can only rely on noisy proprioceptive observation $o_t \in \Omega$ during deployment. Essentially, knowledge transfer aims to find the observation probabilities $\mathcal{O} : \mathcal{S} \to \Omega$ such that the student policy can estimate its current status for decision making, either with latent space estimation [4] or direct action imitation [3], [10], [11].

Although many existing quadruped locomotion solutions have used the historical information $\mathcal{T}$ to capture the temporal information [3], [4], [10], they only treat it as a whole when predicting the next action $a_{t+1} = \pi(\mathcal{T})$ while missing the transition information $T(s_{t+1}|s_t, a_t)$ hidden in the sequence itself. With the transformer architecture and its next token prediction capability [17], we can extract and utilize information about the state-action transition from

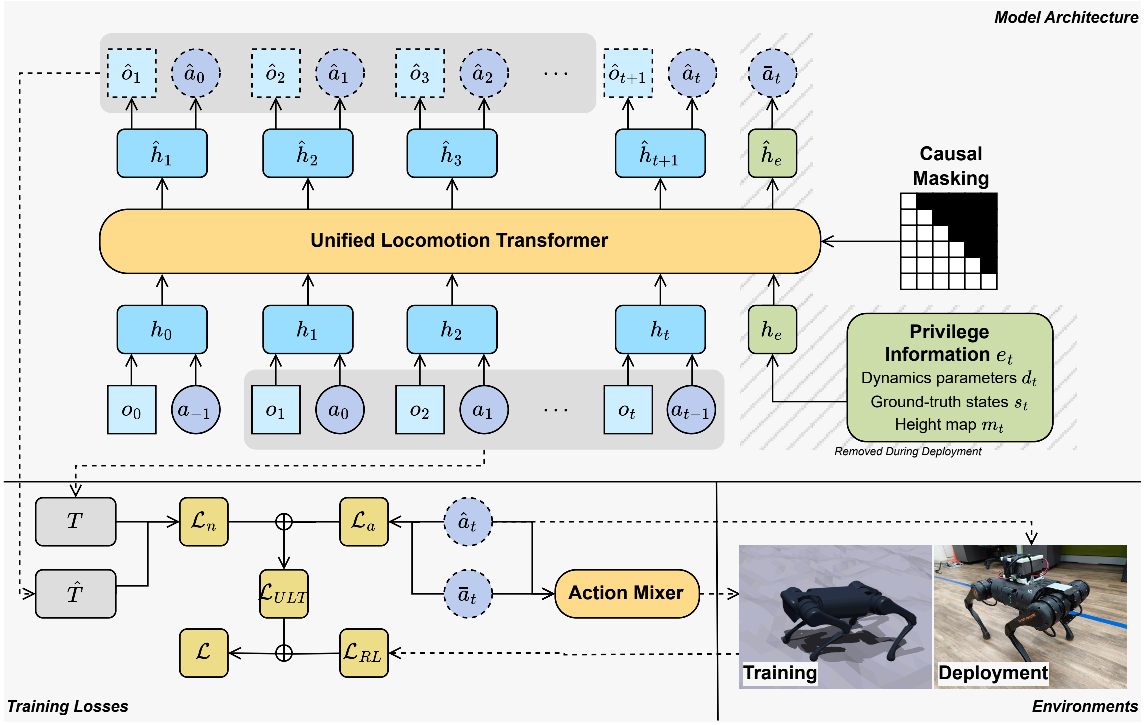
Model Architecture
$\hat{o}_1$ $\hat{a}_0$ $\hat{o}_2$ $\hat{a}_1$ $\hat{o}_3$ $\hat{a}_2$ $\cdots$ $\hat{o}_{t+1}$ $\hat{a}_t$ $\bar{a}_t$
$\hat{h}_1$ $\hat{h}_2$ $\hat{h}_3$ $\hat{h}_{t+1}$ $\hat{h}_e$
Causal Masking
Unified Locomotion Transformer
h_0 h_1 h_2 h_t h_e
o_0 a_{-1} o_1 a_0 o_2 a_1 $\cdots$ o_t a_{t-1}
Privilege Information e_t
Dynamics parameters d_t
Ground-truth states s_t
Height map m_t
Removed During Deployment
T $\hat{T}$ $\mathcal{L}_n$ $\mathcal{L}_a$ $\mathcal{L}_{ULT}$ $\mathcal{L}$ $\mathcal{L}_{RL}$ $\hat{a}_t$ $\bar{a}_t$
Action Mixer
Training
Deployment
Training Losses
Environments

Fig. 1. Illustration of the Unified Locomotion Transformer (ULT) framework. ULT is a vanilla transformer-based architecture to unify the optimization of locomotion policy and knowledge transfer. With state-action trajectories and privilege information in a single framework, both teacher and student actions can be generated simultaneously. The optimization is conducted jointly through PPO by combining the RL loss and transformer loss, which contains the next state-action prediction for future trajectories, and action imitation between student and teacher policies. During training in simulation, an action mixer is used to ensure both policies are played to enhance exploration. During the physical deployment, only proprioceptive observation is used for student actions to achieve zero-shot sim-to-real transfer.

the history $\mathcal{T}$. By pairing it with policy imitation from the privilege information and mixed exploration, we can greatly improve the sample efficiency to discovery the state transition and observation probabilities, thus achieving the optimal teacher and student policies simultaneously in just one training phase.

A. Model Architecture

The foundational part of ULT is a vanilla transformer [16]. It contains multiple stacked multihead attention blocks with causal masking, so the tokens can only attend to themselves and the past tokens while the proprioceptive tokens will not access the information from the privilege tokens.

Similarly to [17], we first tokenize the input trajectory $\mathcal{T}$ with a concatenated states action pair by a shared linear projection layer $W \in \mathbb{R}^{d\times(m+n)}$, where $m = 37$ and $n = 12$ are the dimensions of observation o and a at each step t. We choose $d = 128$ as the size of the token embedding:

$$z_t = \text{concat}(o_t, a_{t-1}) \tag{3}$$

$$h_t = W z_t \tag{4}$$

For privilege information $e_t = [d_t, s_t, m_t]$, we use an environmental factor encoder μ with a three-layer MLP similar to [4] to project it into the same embedding space:

$$h_e = \mu(e) \tag{5}$$

The transformer module takes the entire sequence of $H = [h_0, h_1, \cdots, h_t, h_e]$ with privilege information at the end to ensure that the information will not be leaked to the proprioceptive observation. The sequence is then processed through all the attention layers:

$$\begin{aligned}\hat{H} &= \texttt{ULT}(H) \\ &= [\hat{h}_0, \hat{h}_1, \cdots, \hat{h}_t, \hat{h}_e]\end{aligned} \tag{6}$$

Next State-Action Prediction. To perform an aligned prediction, we extract the first $(t-1)$ tokens from the output $\hat{H}_{0:(t-1)} = [\hat{h}_0, \hat{h}_1, \cdots, \hat{h}_{t-1}]$, and decode them through another shared linear project $\hat{W} \in \mathbb{R}^{(m+n)\times d}$ to predict the future state action trajectory for each step:

$$\hat{z}_{t+1} = \hat{W}\hat{h}_t \tag{7}$$

$$\hat{T} = [\hat{z}_1, \hat{z}_2, \cdots, \hat{z}_t] \tag{8}$$

We can compare the next state-action pairs between the predicted trajectory $\hat{T}$ and actual trajectory $T = [z_1, z_2, \cdots, z_t]$:

$$\mathcal{L}_n = \frac{1}{t}\sum_{1}^{t} \|z_t - \hat{z}_t\|^2 \tag{9}$$

By optimizing $\mathcal{L}_n$, the transformer can always learn the transition relation of the robot state and action, regardless of the actions taken or the quality of the trajectories.

Action Output. ULT can simultaneously output actions from the teacher and student. For the student, with the next state-action prediction, we already implement a decoder layer to extract the predicted next state-action trajectory and it can already output the next action at each time step. Thus,

TABLE III
HYPERPARAMETERS FOR PPO

Parameters	Value
Number of GPUs	2
Actors per GPU	4096
Episode Length	20s
Horizon Length	24
Mini Epochs	5
Minibatch Size	16384
Learning Rate	3e-3
Scheduler	cosine
Optimizer	AdamW
Clip range	0.2
Entropy coefficient	0.005
Reward Discount	0.99
GAE Discount	0.95
Desired KL-divergence	0.008
Weight Decay	0.01

we directly reuse the information in Eq. 7 to form the last concatenated state-action input token:

$$\hat{a}_t = (\hat{z}_{t+1})_{m:(m+n)} \tag{10}$$

For the teacher, with the process through all the attention layers, the resulted $\hat{h}_e$ (Eq. 6) have already gathered all the information from the state-action trajectory due to casual masking. In order to generate actions from $\hat{h}_e$, we implement a policy π with an MLP network similar to [4]:

$$\bar{a}_t = \pi(\hat{h}_e) \tag{11}$$

Thus, we can combine the imitation loss of the action and the next state-action prediction to get the overall performance with a weighting factor β:

$$\mathcal{L}_a = \|\bar{a}_t - \hat{a}_t\|^2 \tag{12}$$

$$\mathcal{L}_{\texttt{ULT}} = \mathcal{L}_n + \beta \mathcal{L}_a \tag{13}$$

B. Action Mixer and Unified Training

In order to achieve optimized performance of $\bar{a}_t$ and $\hat{a}_t$ simultaneously without a pre-trained policy, online trajectories generated by both actions are needed. In a massive parallelism training environment with X agents, an agent mask M is created with a threshold α as the mix ratio such that:

$$a_i = \begin{cases} \bar{a}_i, \text{if } M_i < \alpha \\ \hat{a}_i, \text{otherwise} \end{cases} \quad \text{where } M_i \sim \mathcal{U}(0,1), i = 1, \cdots, X \tag{14}$$

Thus, a higher mix ratio α means more involvement of the teacher. The agent mask M is frequently resampled to ensure exploration for both policies. With the trajectories generated by the resulted a, we use PPO [31] to jointly optimize the policy and action imitation by appending the PPO RL loss and transformer loss with the hyperparameters in Table III:

$$\mathcal{L} = \mathcal{L}_{\text{RL}} + \lambda \mathcal{L}_{\texttt{ULT}} \tag{15}$$

Optimization of $\mathcal{L}$ can simultaneously leverage the state transition, action imitation, and guidance from the teacher policy based on the privilege information to achieve a unified training in a single phase.

C. Direct Sim-to-Real Deployment

The transformer architecture gives flexibility in variable input length. During training, causal masking ensures that future information will not be seen by previous tokens and the privilege information will never be leaked to proprioceptive observation. Thus, when deployed in the real world, we can directly remove the last privileged environmental token h_e safely from the input sequence to generate the student action $\hat{a}_i$ directly with only the historical information $\mathcal{T}$.

V. EXPERIMENTS AND RESULTS

We evaluate the effectiveness of ULT in simulated environments, mainly focusing on three metrics: the average linear and angular velocity tracking return per step for task-related performance and the final total episode reward return for the overall locomotion quality. All the reported results are averaged over 5000 trajectories collected across all terrain types and levels and normalized over the performance of a pretrained privilege Oracle policy adopted from [4] on respective terrain, which is also used as the common teacher for all baselines for a fair comparison.

A. Action Mixer Ratio

The first question we want to answer is: what is the optimal value for the Action Mixer introduced in Sec. IV-B? This mix ratio directly decides the proportion of the trajectories of the teacher and the student during the exploration and eventually affects the final performance due to the difference in their information density and the combined optimization objective with the next prediction and imitation of actin. To answer it, we train multiple ULT models with different values of α and keep all other configurations untouched. Fig. 2 shows the key metrics for different terrains during testing.

We observe that the teacher policy suffers when the mix ratio is too low, as it cannot gather enough trajectories to reach an optimal policy, even with the help from all the privilege information. When the mix ratio is too high, the performance of the student starts to drop, as the training process is overly dependent on the guidance from the teacher policy and does not have enough exploration experience to handle out-of-the-distribution situations in complex environments.

In most cases, the knowledge is transferred efficiently, and the student policy can achieve similar performance as its respective teacher. One special case is $\alpha = 1$, where only the teacher trajectory is generated and used during training while the student is trained in a purely supervised manner. Although it produces one of the best performing teacher policies, the student acts poorly due to the lack of exploration, resulting in low survival rate. Another special case of $\alpha = 0$ will be discussed in Sec. V-C as it has a fundamental difference due to the lack of teacher participation and optimization.

For the rest of this paper, we will use the policies trained with a ratio of $\alpha = 0.6$ unless specified otherwise.

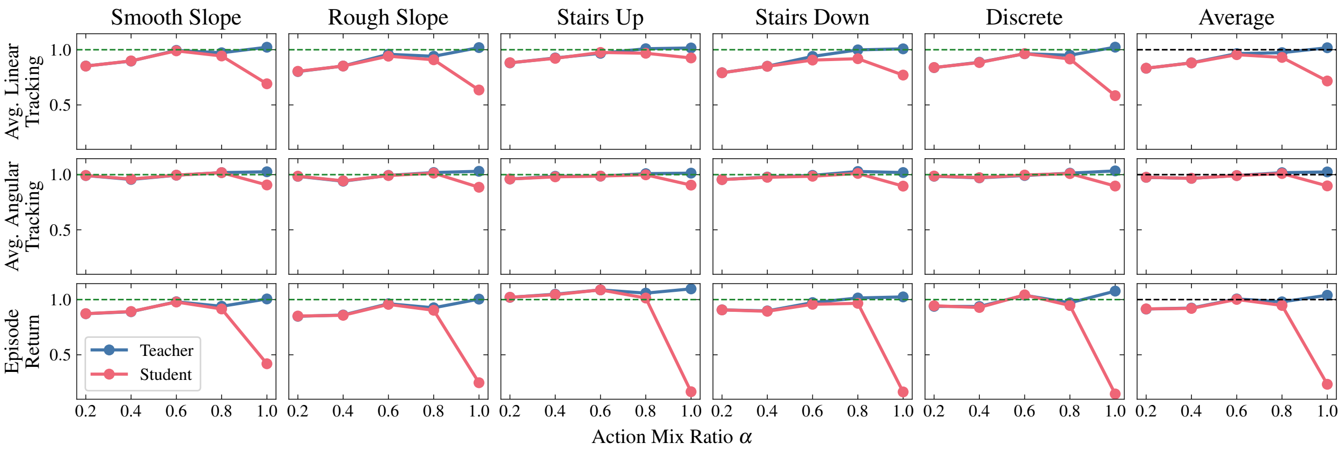

Fig. 2. Performance of ULT with different values of Action Mixer ratio α on five terrains and the overall performance across all trails.

TABLE IV

NORMALIZED PERFORMANCE IN KEY METRICS WITH DIFFERENT BASELINES ON FIVE TERRAIN TYPES AND AVERAGE ACROSS ALL TRAILS.

Terrain	Metric	ULT (Ours)		Supervised Transfer			Joint Transfer	CENet	PPO
		Teacher	Student	Offline-Only	Online-Only	Two-Stages			
Smooth Slope	Avg. Linear Tracking	0.994	0.990	0.691	0.994	1.003	0.907	1.005	0.956
	Avg. Angular Tracking	0.995	0.995	0.851	1.006	1.002	0.958	1.020	0.977
	Total Episode Return	0.980	0.978	0.412	0.999	0.993	0.880	1.001	0.916
Rough Slope	Avg. Linear Tracking	0.959	0.941	0.660	0.985	0.990	0.882	0.950	0.926
	Avg. Angular Tracking	0.993	0.992	0.844	1.007	0.997	0.957	1.015	0.987
	Total Episode Return	0.962	0.957	0.432	0.977	0.967	0.876	0.965	0.911
Stairs Up	Avg. Linear Tracking	0.967	0.965	0.837	0.962	0.995	0.943	0.966	0.933
	Avg. Angular Tracking	0.987	0.985	0.844	1.000	0.973	0.965	0.982	0.982
	Total Episode Return	1.088	1.086	0.138	0.829	0.824	0.890	1.074	0.933
Stairs Down	Avg. Linear Tracking	0.939	0.906	0.705	0.948	0.953	0.884	0.872	0.890
	Avg. Angular Tracking	0.993	0.986	0.837	0.993	0.979	0.956	0.961	0.967
	Total Episode Return	0.973	0.957	0.125	0.836	0.879	0.923	0.929	0.877
Discrete	Avg. Linear Tracking	0.964	0.964	0.613	1.000	0.996	0.848	0.915	0.857
	Avg. Angular Tracking	0.992	0.997	0.805	1.012	1.003	0.944	0.983	0.957
	Total Episode Return	1.039	1.043	0.107	1.009	0.990	0.889	0.926	0.905
Average	Avg. Linear Tracking	0.965	0.953	0.698	0.978	0.987	0.892	0.942	0.912
	Avg. Angular Tracking	0.992	0.991	0.836	1.004	0.991	0.956	0.992	0.974
	Total Episode Return	1.006	1.001	0.249	0.932	0.933	0.892	0.978	0.908

B. Comparison with Baselines

We compare ULT with several knowledge transfer solutions and their variants with a base teacher network similar to [10]:

- **Supervised Transfer.** Based on [3], [4], [10], we implemented different variants of supervised knowledge transfer with direct action imitation. **Offline-Only**: single stage with Oracle pre-trained policy used for trajectory generation [3]; **Online-Only**: single stage with the student generating online trajectories with [4]; **Two-Stages**: combining two stages with offline pre-training first, followed by online correction [10].
- **Joint Transfer.** Following [11], combining the RL exploration of the student with online supervised transfer with the joint ratio of policy imitation gradually annealed to zero by the mid-point of training.
- **CENet.** Implementation of an auto-encoder model from DreamWaQ [14] with VAE loss for the next observation, base velocity and latent space estimation with asymmetric actor-critic architecture.
- **PPO.** We use vanilla PPO [31] to train ULT solely on proprioceptive observation and RL loss, which is equivalent to an action mixer ratio of $\alpha = 0$, and with the teacher head and other loss modules disabled, resulting in a student-only training.

The comparison results are summarized in Table IV. Although ULT only uses the same number of trajectories as Oracle training, both the teacher and student policies achieve the similar performance level of Oracle performance and outperform other baseline models, which require many more trajectories to be generated for most cases. This shows the high efficiency of our proposed framework. With the increased difficulty of using omnidirectional control on all types of terrain, single-stage offline supervised transfer cannot efficiently capture the dynamics of the environment with just good trajectories, and the rough terrains make it hard for the student to survive. Although online supervised transfer and two-stage transfer significantly improve performance, ULT still outperforms supervised transfer with fewer trajectories used. Joint transfer require manual tuning of the joint ratio and struggles to handle the increased environmental challenges with similar performance to vanilla PPO with

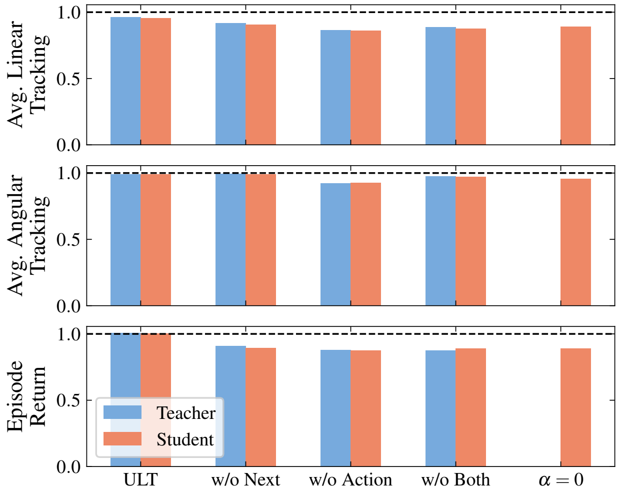

Fig. 3. Normalized metrics for ULT and its ablated variants. The return is averaged across trails on all five terrains.

direct student-only training.

Learning with VAE loss shows strong performance, with only slight disadvantages compared to ULT, making it one of the best performing baselines. These results strongly support the shared idea of DreamWaQ and ULT, that understanding the state-action transition can significantly enhance policy optimization and overall performance.

These results demonstrate that with our ULT framework, optimal teacher and student policies can be achieved at the same time with a more compact unified network, without the need for multiple stages of knowledge transfer with complex network and loss function design. This greatly eases the training setup and the difficulty of knowledge transfer for quadruped locomotion for sim-to-real deployment.

C. Ablation Studies

To better evaluate the main components of ULT, we create different variants by removing the next state-action predictions, imitation of actions, and both from the unified optimization pipeline. The results are shown in Fig. 3. With unified training, our core transformer benefits from both the next state-action prediction module and action imitation module, to increase its capability of understanding state-action transition with high-quality action guided by privilege information.

In addition, we explore the special case of the mix ratio $\alpha = 0$, which is equipment to remove the action mixer module. As the teacher head will not be optimized, there is also no need to imitate the action. It is clear that missing the privilege information makes it difficult to optimize the next state-action prediction resulted in a performance similar to vanilla PPO. ULT needs all modules to work together in order to achieve the sample and learning efficiency.

D. ULT with Supervised Transfer

Training with proper mixed actions ensures that we can achieve optimal teacher and student policy at the same time, but a teacher-only policy still has slightly better performance. Can ULT act as the teacher first and then as the classic supervised knowledge transfer in a single network? Tab. V shows the average performance of the original student agent and after an online supervised transfer phase. Although we can recover some of the performance with additional transfer stage, it is still not comparable to ULT as we can hardly update the transformer while fixing the output of the teacher head, demonstrating the importance and efficiency from the action mixer while training the framework as a whole.

TABLE V
PERFORMANCE OF $\alpha = 1$ STUDENT BEFORE AND AFTER SUPERVISED KNOWLEDGE TRANSFER.

	Avg. Linear Tracking	**Avg. Angular Tracking**	**Episode Return**
Original	0.717	0.898	0.233
After Online Transfer	0.760	0.893	0.865

E. Physical Deployment

Following Sector IV-C, we directly extract the ULT framework and use onboard sensor observations to achieve zero-shot sim-to-real transfer. The policy is exported as JIT for portability and edge inference on a Unitree A1 robot equipped with a Jetson AGX Orin Developer Kit. The policy can run at up to 300Hz and we set the control frequency to 50Hz, $K_p = 30$ and $K_d = 0.7$ while communicating with A1's onboard low-level controller. Fig. 4 shows some snapshots from the deployment test. Please refer to the supplementary video for more information.

VI. Conclusion

We introduce ULT, a unified framework based on transformers for simultaneous optimization of teacher and student policies for quadruped locomotion. With next state-action prediction and action imitation, ULT can efficiently extract valuable transition information and provide guidance with privileged information for mixed exploration to improve the training process. This greatly reduces the complexity and trajectory data needed for sim-to-real transfer, enabling the direct deployment of the agent on physical systems.

Although we have simplified the training process as one single phase without training a dedicated teacher policy, we still need to retrain the model when new task requirement is raised. It is appealing to explore the continual learning and generalization capability through the power of large language models (LLMs) in handling wide variety of information with multimodality and transfer to different tasks and embodiments as future work.

References

[1] J. Tan, T. Zhang, E. Coumans, A. Iscen, Y. Bai, D. Hafner, S. Bohez, and V. Vanhoucke, "Sim-to-real: Learning agile locomotion for quadruped robots," *arXiv preprint arXiv:1804.10332*, 2018.

[2] J. Hwangbo, J. Lee, A. Dosovitskiy, D. Bellicoso, V. Tsounis, V. Koltun, and M. Hutter, "Learning agile and dynamic motor skills for legged robots," *Science Robotics*, vol. 4, no. 26, p. eaau5872, 2019.

[3] 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.

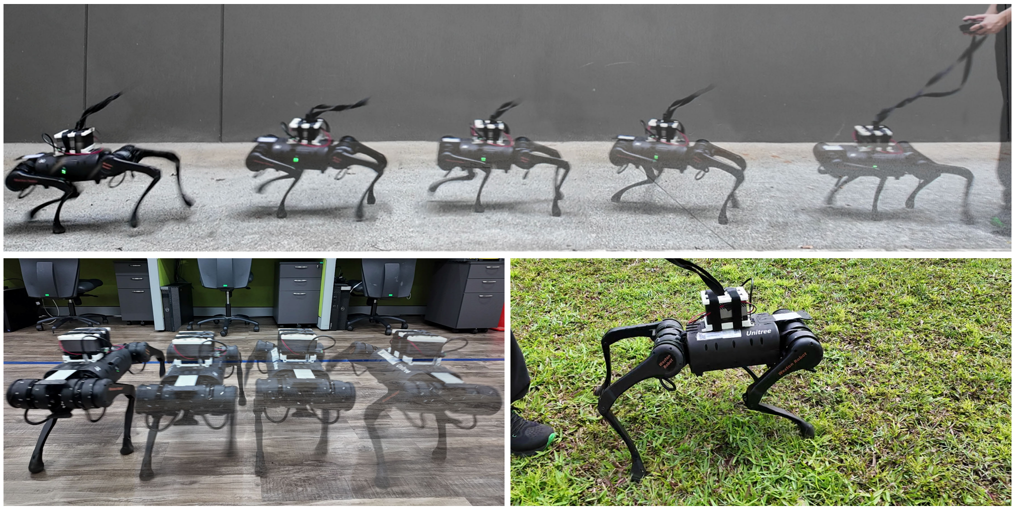

Fig. 4. Footage snapshots showing a single trained `ULT` policy deployed in the real world with zero-shot transfer on Unitree A1 equipped with Jetson Orin AGX for inference on different terrains with motion in omnidirections.

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

[5] E. Krotkov, D. Hackett, L. Jackel, M. Perschbacher, J. Pippine, J. Strauss, G. Pratt, and C. Orlowski, "The darpa robotics challenge finals: Results and perspectives," in *The DARPA Robotics Challenge Finals: Humanoid Robots To The Rescue*. Springer, 2018, pp. 1–26.

[6] C. D. Bellicoso, M. Bjelonic, L. Wellhausen, K. Holtmann, F. Günther, M. Tranzatto, P. Fankhauser, and M. Hutter, "Advances in real-world applications for legged robots," *Journal of Field Robotics*, vol. 35, no. 8, pp. 1311–1326, 2018.

[7] Z. Chen, T. Fan, X. Zhao, J. Liang, C. Shen, H. Chen, D. Manocha, J. Pan, and W. Zhang, "Autonomous social distancing in urban environments using a quadruped robot," *IEEE Access*, vol. 9, pp. 8392–8403, 2021.

[8] J. Hooks, M. S. Ahn, J. Yu, X. Zhang, T. Zhu, H. Chae, and D. Hong, "Alphred: A multi-modal operations quadruped robot for package delivery applications," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 5409–5416, 2020.

[9] W. Yu, J. Tan, C. K. Liu, and G. Turk, "Preparing for the unknown: Learning a universal policy with online system identification," *arXiv preprint arXiv:1702.02453*, 2017.

[10] 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)*. IEEE, 2023, pp. 5141–5147.

[11] I. Radosavovic, T. Xiao, B. Zhang, T. Darrell, J. Malik, and K. Sreenath, "Learning humanoid locomotion with transformers," *arXiv e-prints*, pp. arXiv–2303, 2023.

[12] S. Ross, G. Gordon, and D. Bagnell, "A reduction of imitation learning and structured prediction to no-regret online learning," in *Proceedings of the fourteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings, 2011, pp. 627–635.

[13] 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.

[14] I. M. A. Nahrendra, B. Yu, and H. Myung, "Dreamwaq: Learning robust quadrupedal locomotion with implicit terrain imagination via deep reinforcement learning," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 5078–5084.

[15] 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.

[16] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[17] 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.

[18] D. Liu, T. Zhang, J. Yin, and S. See, "Masked sensory-temporal attention for sensor generalization in quadruped locomotion," *arXiv preprint arXiv:2409.03332*, 2024.

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

[20] S. Reed, K. Zolna, E. Parisotto, S. G. Colmenarejo, A. Novikov, G. Barth-Maron, M. Gimenez, Y. Sulsky, J. Kay, J. T. Springenberg *et al.*, "A generalist agent," *arXiv preprint arXiv:2205.06175*, 2022.

[21] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, J. Dabis, C. Finn, K. Gopalakrishnan, K. Hausman, A. Herzog, J. Hsu *et al.*, "Rt-1: Robotics transformer for real-world control at scale," *arXiv preprint arXiv:2212.06817*, 2022.

[22] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, X. Chen, K. Choromanski, T. Ding, D. Driess, A. Dubey, C. Finn *et al.*, "Rt-2: Vision-language-action models transfer web knowledge to robotic control," *arXiv preprint arXiv:2307.15818*, 2023.

[23] A. Padalkar, A. Pooley, A. Jain, A. Bewley, A. Herzog, A. Irpan, A. Khazatsky, A. Rai, A. Singh, A. Brohan *et al.*, "Open x-embodiment: Robotic learning datasets and rt-x models," *arXiv preprint arXiv:2310.08864*, 2023.

[24] C. Sferrazza, D.-M. Huang, F. Liu, J. Lee, and P. Abbeel, "Body transformer: Leveraging robot embodiment for policy learning," *arXiv preprint arXiv:2408.06316*, 2024.

[25] 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.

[26] M. J. Kim, K. Pertsch, S. Karamcheti, T. Xiao, A. Balakrishna, S. Nair, R. Rafailov, E. Foster, G. Lam, P. Sanketi *et al.*, "Openvla: An open-source vision-language-action model," *arXiv preprint arXiv:2406.09246*, 2024.

[27] R. Yang, M. Zhang, N. Hansen, H. Xu, and X. Wang, "Learning vision-guided quadrupedal locomotion end-to-end with cross-modal transformers," *arXiv preprint arXiv:2107.03996*, 2021.

[28] N. Bohlinger, G. Czechmanowski, M. Krupka, P. Kicki, K. Walas, J. Peters, and D. Tateo, "One policy to run them all: an end-to-end learning approach to multi-embodiment locomotion," *arXiv preprint arXiv:2409.06366*, 2024.

[29] R. Doshi, H. Walke, O. Mees, S. Dasari, and S. Levine, "Scaling cross-embodied learning: One policy for manipulation, navigation, locomotion and aviation," *arXiv preprint arXiv:2408.11812*, 2024.

[30] 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.

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