ULC: A Unified and Fine-Grained Controller for Humanoid Loco-Manipulation

RSS Submit 355 Authors

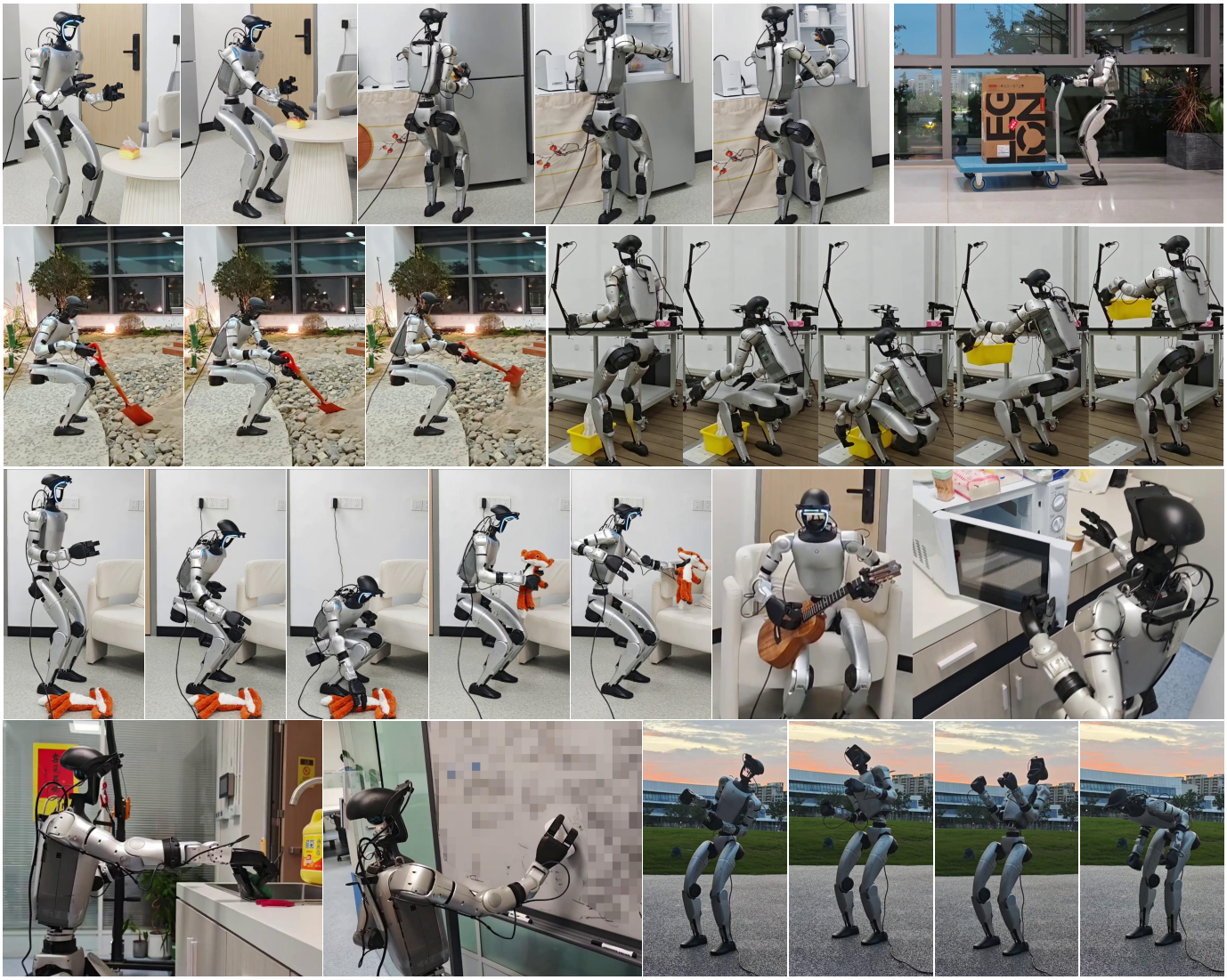

Fig. 1: Diverse loco-manipulation capabilities enabled by ULC. The humanoid robot demonstrates various coordinated whole-body actions, including picking up bread from a table and placing it in a refrigerator, pushing a cart with coordinated locomotion, squatting to shovel sand from the ground, lifting boxes from the floor to table height with dual-arm coordination, picking up dolls from the ground with hand switching and placing them on a sofa, sitting and playing ukulele with fine motor control, placing items in a microwave with precise manipulation, cleaning kitchen surfaces with wiping motions, erasing blackboards with arm coordination, and performing torso rotation in outdoor environments.

***Abstract*—Loco-manipulation for humanoid robots aims to enable robots to integrate mobility with upper-body tracking capabilities. Most existing approaches adopt hierarchical architectures that decompose control into isolated upper-body (manipulation) and lower-body (locomotion) policies. While it reduces training complexity, this decomposition inherently limits coordination between subsystems and contradicts the unified whole-body control exhibited by humans. We demonstrate that a single unified policy can achieve a combination of tracking accuracy, large workspace, and robustness for humanoid loco-manipulation. We propose a Unified Loco-Manipulation Controller (ULC), a single-policy framework that simultaneously tracks root velocity, root height, torso rotation, and dual-arm joint positions in an end-to-end manner, demonstrating the feasibility of unified control without sacrificing the performance. We achieve this unified control through integrating a set of key technologies, including sequential skill acquisition for progressive learning complexity, residual action modeling for fine-grained control adjustments, command**

Method	Architecture	Legs	Torso Yaw	Torso Pitch	Torso Roll	Dual Arms	Workspace	Precision
HOMIE [1]	Decoupled	RL-1	PD	-	-	PD	Medium	Medium
FALCON [2]	Decoupled	RL-1	RL-1	RL-1	RL-1	RL-2	Medium	High
JAEGER [3]	Decoupled	RL-1	RL-1	-	-	RL-2	Medium	High
AMO [4]	Decoupled	RL	RL	RL	RL	PD	Large	Medium
SoFTA [5]	Decoupled	RL-1	RL-1	-	-	RL-2	Medium	Medium
R^2S^2 [6]	Unified	RL	RL	RL	RL	RL	Medium	Medium
ULC (Ours)	Unified	RL	RL	RL	RL	RL	Large	High

TABLE I: Comparison of humanoid loco-manipulation controllers. Colors indicate control types: Blue/Red: RL, Orange: PD, Purple: Unified RL, Gray: Not controlled.

polynomial interpolation for smooth motion transitions, random delay release for robustness to deploy variations, load randomization for generalization to external disturbances, and center of mass tracking for providing explicit policy gradients to maintain stability. We validate our method on the Unitree G1 humanoid robot with a three-degrees-of-freedom waist. Compared with the state-of-the-art, ULC shows better tracking performance than disentangled methods and demonstrates larger workspace coverage. The unified dual-arm tracking enables precise manipulation under external loads while maintaining coordinated whole-body control for complex loco-manipulation tasks.

I. INTRODUCTION

Humanoid robots, with their human-like morphology, represent a promising paradigm for versatile systems that can operate in human-centric environments. Recent advances have significantly improved locomotion [7, 8, 9, 10, 11, 12, 13, 14] and manipulation [15, 16, 17, 18, 19]. These results are often achieved by coupling high-level decision-making—via Imitation Learning (IL) [20, 21] or Vision-Language-Action (VLA) models [22, 18, 19]—with Loco-Manipulation Controllers (LMCs) [1, 4] that map commands to whole-body motions.

An effective LMC should translate motion commands into joint-level actions with minimal tracking error while maintaining dynamic stability. However, LMC design involves key trade-offs. The command space (e.g., joint positions, Cartesian poses, root velocity, root height) affects feasibility, conflicts across objectives, and how fully the robot can exploit its kinematic and dynamic range [23]. The control architecture also matters: whole-body controllers [24, 23, 25, 26, 27, 8, 28] offer strong coordination but are harder to train, whereas decoupled upper/lower-body designs [1, 3, 2] simplify learning but can weaken coordination. Finally, training data sources impose different biases: motion capture provides realistic motion but can be noisy, kinematically infeasible, and biased [24, 23, 25, 29, 30, 31, 32]; procedurally sampled commands improve coverage but are often limited (especially for legs) due to humanoid stability constraints [33, 34, 2, 1].

These choices ultimately determine deployability: command design must balance expressiveness and feasibility, architectures must balance coordination and training cost, and data generation must balance realism and coverage.

To address these issues, we propose a Unified Loco-Manipulation Controller (ULC) trained with massively parallel reinforcement learning to track procedurally sampled commands including root velocity, root height, torso orientation, and arm joint positions. By simplifying leg commands compared to motion-capture-driven approaches, we improve coverage of the feasible command space while preserving whole-body coordination. We enable single-model multi-task tracking via (i) feasibility-aware command space design, (ii) progressive curriculum learning, (iii) residual action modeling [35, 36, 8] to improve tracking precision, and (iv) sequential skill acquisition [37] to reduce catastrophic forgetting. For deployment-realistic command generation, we combine fixed-interval random sampling with fifth-degree polynomial interpolation [38, 39] and introduce stochastic command release that probabilistically buffers/releases commands to emulate deployment variations while keeping commands feasible. To improve robustness under varying payloads, we add center of mass tracking rewards [27] encouraging the center of mass projection to remain within the support polygon. Experiments in simulation and on hardware show state-of-the-art tracking performance, workspace coverage, and robustness; ablations verify the role of each component. Our contributions are:

- A unified framework with feasibility-aware command space design and progressive curriculum learning for multi-task loco-manipulation.
- Deployment-realistic training with stochastic command release and explicit balance optimization for sim-to-real transfer and payload robustness.
- Extensive experiments demonstrating improved tracking performance and generalization across diverse tasks.

II. RELATED WORK

A. Humanoid Loco-Manipulation Controller

Humanoid loco-manipulation control confronts fundamental challenges in coordinating locomotion with manipulation while maintaining tracking accuracy and system robustness [2, 1, 4].

Traditional approaches frequently employ decoupled control strategies that isolate leg and arm movements to reduce training complexity. Representative implementations integrate reinforcement learning for leg control with PD controllers for arms [1] while demonstrating suboptimal arm tracking under gravitational loads. Force adaptation challenges are tackled through joint upper-body policy training with force curriculum [2]. Dual-level control architectures implement separate upper/lower body controllers capable of root velocity

Parameter	Unit	Range
Linear Velocity X	m/s	[-0.45, 0.55]
Linear Velocity Y	m/s	[-0.45, 0.45]
Angular Velocity Z	rad/s	[-1.2, 1.2]
Root Height	m	[0.3, 0.75]
Torso Rotation Yaw	rad	[-2.62, 2.62]
Torso Rotation Roll	rad	[-0.52, 0.52]
Torso Rotation Pitch	rad	[-0.52, 1.57]
Arm Joint Positions	-	Robot Design Limits

TABLE II: Command space specifications.

tracking and fine-grained joint control [3], yet remain dependent on motion retargeting that may introduce artifacts. Hand stabilization during locomotion is achieved through multi-frequency frameworks using distinct upper/lower-body agents [5]. A new perspective switching dual-arm input source via binary flags [34] demonstrates conceptual advancement, but exhibits under-refined dual-arm tracking under PD control. Skill-space methodologies enable extended mobility through primitive ensembling [6]. However, none of the above methods completely frees up the working space of trunk rotation. Hierarchical designs combining trajectory optimization with reinforcement learning [4] show performance improvements at the cost of computational overhead, while accurate dual-arm tracking remains unintegrated.

B. Humanoid Whole-Body Tracking

Recent significant strides in humanoid whole-body motion tracking now enable robots to reproduce complex human motions using diverse datasets [26, 27, 8, 28, 29, 30, 31, 32], though fundamental challenges persist regarding morphological differences, noise handling, and sim-to-real transfer.

Recent innovations demonstrate diverse pathways for advancement. When training on large-scale motion capture datasets [24], whole-body dexterous manipulation policies emerge yet suffer from diluted learning of extreme motions. For data quality enhancement, teacher-student distillation and dataset selection techniques [40] mitigate issues but introduce performance gaps between teacher and student policies. Extreme motion reproduction is achievable through advanced processing pipelines [27], albeit with limited transferability across motion sequences. Meanwhile, visual imitation from video demonstrations [28] reduces motion capture dependency with video-to-motion techniques. General motion tracking frameworks [29] and robust tracking methods [30] advance whole-body control capabilities. Residual learning approaches [31] and diffusion-based methods [32] bridge the gap from motion tracking to versatile control.

III. PROBLEM FORMULATION

We formulate the humanoid loco-manipulation task as a goal-conditioned Markov Decision Process (MDP) $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, \mathcal{G}, P, R, \gamma \rangle$. The policy $\pi_{\boldsymbol{\theta}} : \mathcal{S} \times \mathcal{G} \to \Delta(\mathcal{A})$ maps state-command observations to a Gaussian distribution over actions:

$$\pi_{\boldsymbol{\theta}}(\boldsymbol{a}_t|\boldsymbol{s}_t, \boldsymbol{g}_t) = \mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{\theta}}(\boldsymbol{s}_t, \boldsymbol{g}_t), \boldsymbol{\Sigma}_{\boldsymbol{\theta}}(\boldsymbol{s}_t, \boldsymbol{g}_t)) \tag{1}$$

The observation

$$\boldsymbol{o}_{prop}^{(t)} = [\boldsymbol{q}_{joint}^{(t)}, \dot{\boldsymbol{q}}_{joint}^{(t)}, \boldsymbol{\omega}_{base}^{(t)}, \boldsymbol{g}_{proj}^{(t)}, \boldsymbol{a}_{t-1}, \boldsymbol{g}_t] \tag{2}$$

comprises joint positions/velocities, base angular velocity, gravity projection, previous actions, and current commands. We design a factorized command space $\boldsymbol{g} = [\boldsymbol{g}_{loco}, \boldsymbol{g}_{torso}, \boldsymbol{g}_{arms}]^T$ enabling independent control of locomotion ($\boldsymbol{v}_{xy}, \omega_z, h_{pelvis}$), torso orientation (ZXY Euler angles), and arm joint targets (Table II). Actions are target joint positions executed via PD control:

$$\boldsymbol{a}_t = [\boldsymbol{q}_{legs}^{target}, \boldsymbol{q}_{torso}^{target}, \boldsymbol{q}_{arms}^{target}]^T \cdot \alpha_{scale} + \boldsymbol{q}_{default} \tag{3}$$

where $\alpha_{scale} = 0.25$, and arm actions use residual modeling: $\boldsymbol{q}_{arms}^{final} = \boldsymbol{a}_{arms} + \boldsymbol{q}_{arms}^{desired}$ (Section IV-D).

IV. UNIFIED LOCO-MANIPULATION CONTROL

We present ULC, a *unified* and *fine-grained* controller for humanoid loco-manipulation that leverages massive parallel reinforcement learning to train a single policy from scratch. Our framework systematically addresses the fundamental challenges of high-dimensional exploration and skill coordination through four key technical innovations:

1) **Sequential skill acquisition** with adaptive curriculum;
2) **Command interpolation** with stochastic delay modeling;
3) **Load generalization** through dynamic mass distribution and center of mass tracking;
4) **Residual action modeling** for stable training and precise upper body tracking.

A. Sequential Skill Acquisition and Adaptive Curriculum Learning

To address the fundamental challenge of inefficient exploration in high-dimensional command spaces, ULC employs a *sequential skill acquisition strategy* with adaptive command curriculum. The policy progressively masters skills following a carefully designed hierarchical sequence. This sequential approach prevents catastrophic forgetting and ensures robust acquisition of fundamental capabilities before advancing to more complex behaviors.

1) Mathematical Framework for Curriculum Progression: We formalize the curriculum learning process through a structured progression system with rigorous mathematical foundations. Let $\mathcal{T} = \{T_1, T_2, T_3\}$ represent the ordered set of skills to be learned sequentially, where

$$T_1: \text{ Base velocity tracking} \quad (\boldsymbol{v}_{xy}, \omega_z) \tag{4}$$

$$T_2: \text{ Base height tracking} \quad (h_{\text{pelvis}}) \tag{5}$$

$$T_3: \text{ Torso and arm tracking} \quad (\boldsymbol{g}_{\text{torso}}, \boldsymbol{g}_{\text{arms}}) \tag{6}$$

For each skill T_i, we define a *curriculum parameter* $\alpha_i(t) \in [0, 1]$ that controls the difficulty progression over training time t. The curriculum advancement follows a reward-based gating mechanism that evaluates multiple performance metrics simultaneously.

$$\alpha_i(t+1) = \begin{cases} \min\{1, \alpha_i(t) + \Delta\alpha\} & \text{if } \mathcal{C}_i(t) = \texttt{True} \\ \alpha_i(t) & \text{otherwise} \end{cases} \tag{7}$$

Algorithm 1 ULC Sequential Skill Acquisition with Adaptive Curriculum

Input: Skills $\mathcal{T} = \{T_1, T_2, T_3\}$, reward weights $\{w_{vel}, w_{height}, w_{upper}, w_{torso}, w_{hip}\}$
Initialize: $\alpha_2 \leftarrow 0.0$, $\alpha_3 \leftarrow 0.0$, $t \leftarrow 0$
Initialize: Active skills $\mathcal{A} \leftarrow \{T_1\}$, curriculum update interval $I \leftarrow 1000$ steps
while training not converged **do**
  $t \leftarrow t + 1$
  // Sample commands based on current curriculum
  for each skill $T_i \in \mathcal{A}$ **do**
    Sample commands $\boldsymbol{g}_i$ using curriculum parameter α_i
  end for
  // Execute training step
  $\boldsymbol{g} \leftarrow$ Concatenate sampled commands from active skills

  Execute policy $\pi_{\boldsymbol{\theta}}(\boldsymbol{a}_t|\boldsymbol{s}_t, \boldsymbol{g})$
  Compute episode rewards and track running averages
  Update policy parameters $\boldsymbol{\theta}$ using PPO
  // Evaluate curriculum advancement every I steps
  if $t \bmod I = 0$ **then**
    // Height curriculum advancement
    if $\mathcal{C}_2(t)$ and $\alpha_2 < 0.98$ **then**
      $\alpha_2 \leftarrow \min(0.98, \alpha_2 + 0.05)$
      Reset tracked rewards for next evaluation
    end if
    // Upper body curriculum advancement
    if $\mathcal{C}_3(t)$ and $\alpha_3 < 0.98$ **then**
      $\alpha_3 \leftarrow \min(0.98, \alpha_3 + 0.05)$
      Reset tracked rewards for next evaluation
    end if
    // Activate terrain curriculum when both skills mastered
    if $\alpha_2 > 0.98$ and $\alpha_3 > 0.98$ **then**
      Enable terrain level progression
    end if
  end if
end while

where $\Delta\alpha = 0.05$ represents the curriculum increment. The advancement conditions $\mathcal{C}_i(t)$ are specifically designed based on empirical validation:

a) Height Curriculum Advancement ($\mathcal{C}_2$): The height curriculum advancement condition implements a multi-criteria evaluation that ensures the robot has mastered fundamental locomotion skills before introducing height variation challenges. The condition is mathematically defined as:

$$\mathcal{C}_2(t) = \mathcal{C}_{\text{height}}(t) \wedge \mathcal{C}_{\text{velocity}}(t) \wedge \mathcal{C}_{\text{hip}}(t), \tag{8}$$

where each component evaluates specific performance metrics with carefully tuned thresholds:

$$\mathcal{C}_{\text{height}}(t) = R^{\text{avg}}_{\text{height}}(t) \geq \tau_{\text{height}} \tag{9}$$

$$\mathcal{C}_{\text{velocity}}(t) = R^{\text{avg}}_{\text{vel}}(t) \geq \tau_{\text{vel}} \tag{10}$$

$$\mathcal{C}_{\text{hip}}(t) = R^{\text{avg}}_{\text{hip}}(t) \geq \tau_{\text{hip}} \tag{11}$$

The threshold parameters are determined through systematic hyperparameter tuning: $\tau_{\text{height}} = 0.85$ and $\tau_{\text{vel}} = 0.8$ ensure the policy achieves near-convergent tracking performance before curriculum advancement, while $\tau_{\text{hip}} = 0.2$ prevents excessive hip deviation that would compromise subsequent skill learning. These values were selected based on grid search to balance training stability with curriculum progression speed.

b) Upper Body Curriculum Advancement ($\mathcal{C}_3$): The upper body curriculum advancement implements a comprehensive condition that requires mastery of both arm tracking and torso control capabilities, while simultaneously maintaining all previously acquired skills. The advancement criterion is

$$\mathcal{C}_3(t) = \mathcal{C}_{\text{upper}}(t) \wedge \mathcal{C}_{\text{torso}}(t) \wedge \mathcal{C}_{\text{prev}}(t) \wedge \mathcal{C}_{\text{complete}}(t) \tag{12}$$

The individual components are rigorously defined as:

$$\mathcal{C}_{\text{upper}}(t) = R^{\text{avg}}_{\text{upper}}(t) \geq \tau_{\text{upper}} \tag{13}$$

$$\mathcal{C}_{\text{torso}}(t) = R^{\text{avg}}_{\text{torso}}(t) \geq \tau_{\text{torso}} \tag{14}$$

$$\mathcal{C}_{\text{prev}}(t) = \mathcal{C}_{\text{height}}(t) \wedge \mathcal{C}_{\text{velocity}}(t) \wedge \mathcal{C}_{\text{hip}}(t) \tag{15}$$

$$\mathcal{C}_{\text{complete}}(t) = \alpha_2 \geq 0.98 \tag{16}$$

where $R^{\text{avg}}_{\text{upper}}(t)$ denotes the upper body joint tracking reward, and $R^{\text{avg}}_{\text{torso}}(t)$ represents the torso orientation tracking reward, which are detailed in Appendix D. The threshold coefficients $\tau_{\text{upper}} = 0.8$ and $\tau_{\text{torso}} = 0.8$ are chosen to ensure adequate skill mastery; lower values led to premature advancement and skill degradation, while higher values unnecessarily prolonged training.

This multi-criteria gating mechanism ensures that curriculum progression occurs only when all prerequisite skills are sufficiently mastered, thereby preventing catastrophic forgetting and maintaining stable performance across all learned capabilities.

B. Stochastic Delay Mechanism and Command Interpolation

To ensure stable arm movements and enhance training robustness, we implement sophisticated command processing mechanisms such as quintic polynomial interpolation and stochastic delay modeling to accurately reflect real-world communication delays.

a) Quintic Polynomial Interpolation: The upper body commands are smoothly interpolated using quintic polynomial transitions between randomly sampled target positions. This design separates trajectory smoothness from deployment realism: the quintic interpolation generates physically plausible reference trajectories, while the subsequent stochastic delay mechanism introduces realistic command discontinuities. The interpolation is executed over a fixed interval $T_{\text{interval}} = 1.0$ s, with the instantaneous target position determined by

$$\boldsymbol{q}_{\text{target}}(t) = \boldsymbol{q}_{\text{start}} + (\boldsymbol{q}_{\text{goal}} - \boldsymbol{q}_{\text{start}}) \cdot s(t), \tag{17}$$

where $s(t)$ is the quintic smoothing factor

$$s(t) = 10t^3 - 15t^4 + 6t^5, \quad t \in [0, 1]. \tag{18}$$

The movement step counter t_{step} is normalized as $t = \min(t_{\text{step}}/T_{\text{interval}}, 1.0)$ to ensure smooth transitions. This quintic polynomial ensures C^2 continuity with zero velocity and acceleration at the endpoints, providing natural arm movement characteristics.

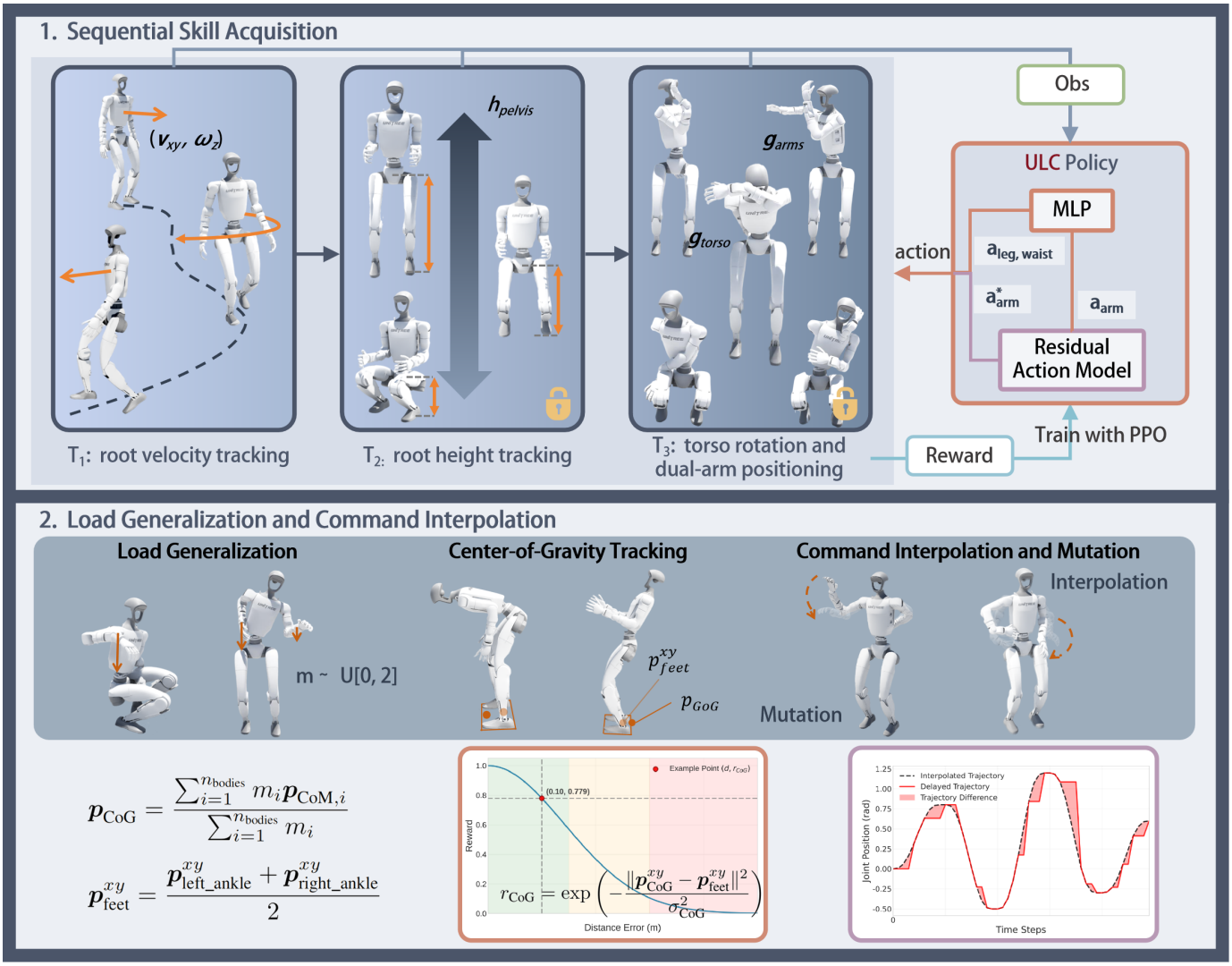

Fig. 2: Method overview of the Unified Loco-Manipulation Controller (ULC). Our approach employs massively parallel reinforcement learning to train a single unified policy that tracks procedurally sampled commands including root velocity, root height, torso orientation, and arm joint positions. The framework addresses multi-task learning challenges through sequential skill acquisition with adaptive curriculum, deployment-realistic command generation with interpolation and random delay, and loaded balance optimization with center of mass tracking.

b) Stochastic Delay Mechanism: The delay mechanism is implemented through a sophisticated accumulation and release system that operates on the incremental commands between consecutive timesteps. Let $\Delta \boldsymbol{q}^{(t)}$ represent the incremental change in target position at timestep t

$$\Delta \boldsymbol{q}^{(t)} = \boldsymbol{q}^{(t)}_{\text{target}} - \boldsymbol{q}^{(t-1)}_{\text{theoretical}}, \tag{19}$$

where $\boldsymbol{q}^{(t-1)}_{\text{theoretical}}$ is the theoretical position from the previous timestep's interpolation.

Delay Mask and Accumulation: At each timestep, a random delay mask $\boldsymbol{d}^{(t)} \in \{0,1\}^{n_j}$ is generated

$$d^{(t)}_j \sim \text{Bernoulli}(p_{delay}), \quad j = 1, \ldots, n_j, \tag{20}$$

where $p_{delay} = 0.5$ is the fixed delay probability.

Command Release: The effective command executed at timestep t releases both the current increment (if not delayed) and any previously accumulated commands (if released):

$$\Delta \boldsymbol{q}^{(t)}_{\text{effective}} = \Delta \boldsymbol{q}^{(t)} \odot (1 - \boldsymbol{d}^{(t)}) + \boldsymbol{A}^{(t-1)} \odot (1 - \boldsymbol{d}^{(t)}) \tag{21}$$

Buffer Update: After command release, the accumulation buffer is updated to retain delayed commands and accumulate new delayed increments:

$$\boldsymbol{A}^{(t)} = \boldsymbol{A}^{(t-1)} \odot \boldsymbol{d}^{(t)} + \Delta \boldsymbol{q}^{(t)} \odot \boldsymbol{d}^{(t)} \tag{22}$$

C. Load Generalization and Balance Control

a) Random Load Distribution: During training, we apply random masses to the robot's wrists to simulate diverse payload conditions. The mass randomization is applied to the robot's wrists masses during environment reset, with the total wrists mass distribution modified to simulate carrying loads.

b) Center of Mass Tracking: We implement a sophisticated center of mass tracking reward that maintains stability across all motion phases. The reward function is formulated as:

$$r_{\text{CoM}} = \exp\left(-\frac{\|\boldsymbol{p}^{xy}_{\text{CoM}} - \boldsymbol{p}^{xy}_{\text{feet}}\|^2}{\sigma^2_{\text{CoM}}}\right) \tag{23}$$

Method	Whole Command Space							Edge Command Space						
	$E_v \downarrow$	$E_\omega \downarrow$	$E_h \downarrow$	$E_y \downarrow$	$E_p \downarrow$	$E_r \downarrow$	$E_a \downarrow$	$E_v \downarrow$	$E_\omega \downarrow$	$E_h \downarrow$	$E_y \downarrow$	$E_p \downarrow$	$E_r \downarrow$	$E_a \downarrow$
HOMIE	0.15±.02	0.18±.02	0.04±.01	0.08±.01	/	/	0.12±.02	0.18±.03	0.22±.03	0.06±.01	0.12±.02	/	/	0.15±.02
HOMIE-3-DoF-Waist	0.14±.02	0.17±.02	0.04±.01	0.09±.01	0.15±.02	0.14±.02	0.12±.02	0.25±.04	0.28±.04	0.10±.02	0.18±.03	0.28±.04	0.26±.04	0.18±.03
FALCON	0.16±.02	0.19±.02	0.05±.01	0.10±.01	/	/	0.08±.01	0.24±.03	0.26±.03	0.09±.02	0.16±.02	/	/	0.14±.02
AMO	**0.08±.01**	**0.10±.01**	0.03±.01	0.07±.01	0.11±.02	0.12±.02	0.11±.02	0.12±.02	0.14±.02	0.05±.01	0.10±.02	0.16±.02	0.15±.02	0.14±.02
R^2S^2	0.13±.02	0.15±.02	0.04±.01	0.17±.02	0.13±.02	/	0.10±.01	0.17±.02	0.20±.03	0.07±.01	0.18±.02	0.19±.03	/	0.13±.02
ULC	0.10±.01	0.12±.01	**0.02±.00**	**0.05±.01**	**0.06±.01**	**0.05±.01**	**0.04±.01**	**0.11±.01**	**0.13±.02**	**0.03±.01**	**0.06±.01**	**0.08±.01**	**0.07±.01**	**0.05±.01**
Method	**Wrist Loaded (2kg)**							**Command Mutation**						
	$E_v \downarrow$	$E_\omega \downarrow$	$E_h \downarrow$	$E_y \downarrow$	$E_p \downarrow$	$E_r \downarrow$	$E_a \downarrow$	$E_v \downarrow$	$E_\omega \downarrow$	$E_h \downarrow$	$E_y \downarrow$	$E_p \downarrow$	$E_r \downarrow$	$E_a \downarrow$
HOMIE	0.18±.02	0.21±.03	0.05±.01	0.10±.01	/	/	0.18±.03	0.28±.04	0.32±.05	0.12±.02	0.20±.03	/	/	0.22±.03
HOMIE-3-DoF-Waist	0.17±.02	0.20±.03	0.05±.01	0.11±.02	0.18±.03	0.17±.02	0.17±.02	0.30±.05	0.35±.05	0.14±.03	0.24±.04	0.32±.05	0.30±.04	0.24±.04
FALCON	0.18±.02	0.21±.03	0.06±.01	0.12±.02	/	/	0.11±.02	0.26±.04	0.29±.04	0.10±.02	0.18±.03	/	/	0.16±.02
AMO	**0.09±.01**	**0.11±.01**	0.04±.01	0.08±.01	0.13±.02	0.14±.02	0.16±.02	0.14±.02	0.16±.02	0.06±.01	0.14±.02	0.22±.03	0.20±.03	0.18±.03
R^2S^2	0.15±.02	0.17±.02	0.05±.01	0.19±.02	0.15±.02	/	0.14±.02	0.20±.03	0.23±.03	0.08±.01	0.22±.04	0.20±.03	/	0.17±.02
ULC	0.10±.01	0.13±.02	**0.03±.01**	**0.06±.01**	**0.07±.01**	**0.06±.01**	**0.05±.01**	**0.12±.02**	**0.14±.02**	**0.03±.01**	**0.07±.01**	**0.09±.01**	**0.08±.01**	**0.06±.01**

TABLE III: **Tracking accuracy comparison across different scenarios.** We present a performance comparison between ULC and baselines for the proposed metrics. The means and standard deviation are reported across 5 evaluations, each with 1024 parallel environments for 50,000 steps. Best results are in **bold**, second best are underlined. "/" indicates the method lacks this capability.

where $\boldsymbol{p}^{xy}_{\text{CoM}}$ is the horizontal projection of the whole-body center of mass, and $\boldsymbol{p}^{xy}_{\text{feet}}$ represents the midpoint between the ankle positions.

Feet Support Reference: The support reference is computed as the midpoint between the ankle positions

$$\boldsymbol{p}^{xy}_{\text{feet}} = \frac{\boldsymbol{p}^{xy}_{\text{left\_ankle}} + \boldsymbol{p}^{xy}_{\text{right\_ankle}}}{2}. \tag{24}$$

This provides a consistent reference point for balance control that accounts for the robot's current stance configuration.

D. Residual Action Modeling for Arm Control

We introduce residual action modeling for arm joints that enables precise tracking while maintaining training stability. The final control command combines policy output with residual correction:

$$\boldsymbol{q}_{\text{processed}} = \alpha_{scale} \cdot \pi_{\boldsymbol{\theta}}(\boldsymbol{s}, \boldsymbol{g}) + \boldsymbol{q}_{\text{default}} \tag{25}$$

$$\boldsymbol{q}_{\text{final}}[\mathcal{J}_{\text{upper}}] = \boldsymbol{q}_{\text{processed}}[\mathcal{J}_{\text{upper}}] + \boldsymbol{q}_{\text{desired}}[\mathcal{J}_{\text{upper}}], \tag{26}$$

where $\boldsymbol{q}_{\text{desired}}$ is generated through command interpolation and delay mechanism. The residual term acts as a feedforward component compensating for predictable dynamics, allowing the policy to focus on learning corrective adjustments rather than reconstructing the entire control signal.

V. EXPERIMENT

A. Experimental Setup

We compare ULC with state-of-the-art loco-manipulation controllers. Baseline results are obtained from official checkpoints or our faithful reimplementations. All methods use identical simulator settings, evaluation protocols, and low-level assumptions (e.g., PD gains, control frequency). Methods with different observation/action interfaces (e.g., MoCap-driven arms, skill primitives) are evaluated in their native interfaces within feasible command ranges. For robustness evaluation, identical payload and command mutation conditions are applied, while training-time randomization follows each method's original design.

Method	Height		Yaw		Pitch		Roll		Arm
	Min	Max	Min	Max	Min	Max	Min	Max	Ctrl
HOMIE	0.30	0.75	-2.62	2.62	/	/	/	/	PD
HOMIE-3-DoF-Waist	0.30	0.75	-2.62	2.62	-0.52	0.52	-0.52	0.52	PD
FALCON	0.50	0.75	-1.00	1.00	/	/	/	/	MoCap
AMO	0.35	0.75	-2.62	2.62	-0.52	1.57	-0.46	0.46	PD
R^2S^2	0.35	0.75	-1.00	1.00	0.00	0.50	/	/	Skill Lib
ULC	**0.30**	**0.75**	**-2.62**	**2.62**	**-0.52**	**1.57**	**-0.52**	**0.52**	**Proc.**

TABLE IV: Reachable command ranges for different methods.

- **HOMIE** [1]: A decoupled controller using RL for legs and PD control for waist yaw and arms.
- **HOMIE-3-DoF-Waist**: An extension of HOMIE with unlocked three waist DoF for PD control.
- **FALCON** [2]: A decoupled controller using dual policy for lower and upper body with adaptive force curriculum.
- **AMO** [4]: A hierarchical controller combining trajectory optimization with RL for leg and waist control, with PD-controlled arms.
- **R^2S^2** [6]: A skill-based whole-body controller using a pre-trained skill library for goal-reaching tasks.

Our metrics include:

- **Root Linear Velocity Tracking Error** E_v
- **Root Angular Velocity Tracking Error** E_ω
- **Root Height Tracking Error** E_h
- **Root Yaw Orientation Tracking Error** E_y
- **Root Pitch Orientation Tracking Error** E_p
- **Root Roll Orientation Tracking Error** E_r
- **Arm Joint Position Tracking Error** E_a

All metrics are computed by rolling out 1024 parallel environments in Isaaclab [41] for 50,000 steps, averaging tracking errors across all timesteps and environments.

B. Comparison of Reachable Workspace

Table IV compares the reachable command ranges across different methods.

HOMIE and **HOMIE-3-DoF-Waist** employ PD control for waist joints, creating a decoupling between torso rotation and leg control. Although this enables full yaw rotation and maximum root height range, the legs cannot actively participate in torso orientation control. **FALCON** prioritizes adaptive force

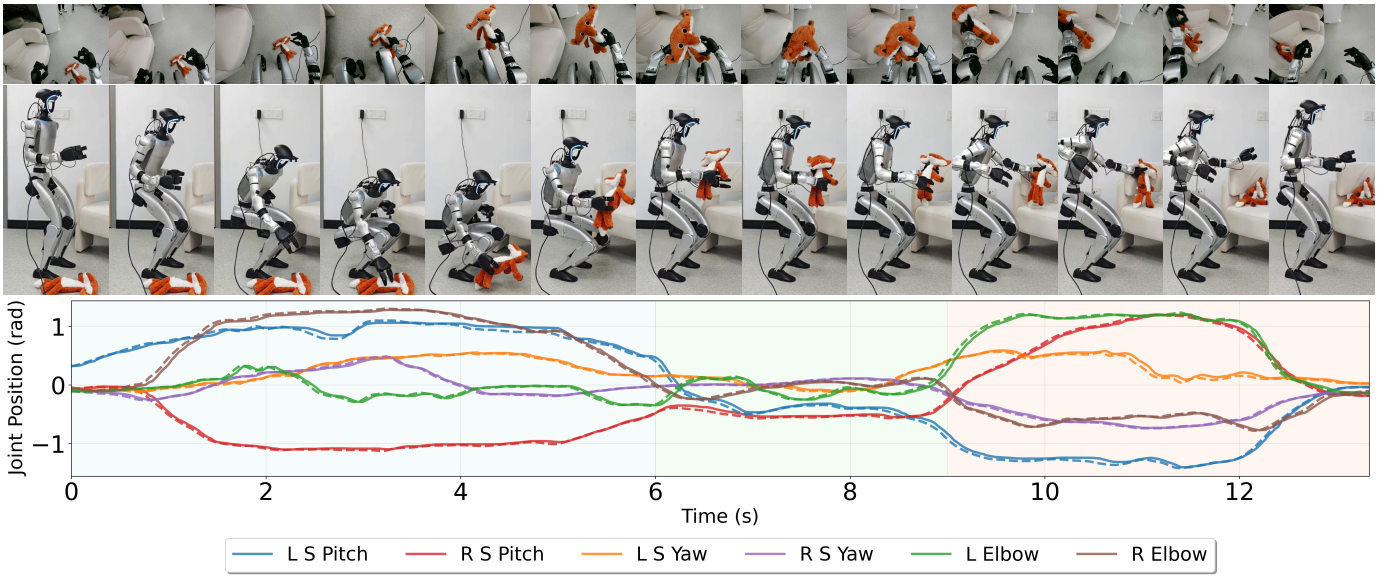

Fig. 3: Time-series visualization of the doll pick-and-place task, covering all key stages: squatting to pick up the doll, hand switching, and placing the doll at the target location.

curriculum learning but suffers from restricted yaw range and elevated minimum root height, with dual-arm control relying entirely on motion capture data. **AMO** addresses the OOD limitation through its motion adaptation module, achieving asymmetric pitch control, but remains constrained in roll orientation. **R^{2}S^2** utilizes a pre-defined skill library, but the reliance on pre-defined primitives constrains torso rotation capabilities.

ULC overcomes these limitations through unified coordinated control. Our approach achieves the maximum root height range, complete torso rotation tracking across all axes, and procedurally sampled dual-arm control without MoCap constraints.

C. Comparison of Tracking Accuracy

Table III evaluates tracking performance across four scenarios with commands sampled within each method's operational ranges: (1) **Whole command space**; (2) **Edge command space**: extreme torso rotation cases; (3) **Wrist loaded**: 2kg external loads; (4) **Command mutation**: random command delays (IV-B).

Locomotion Control: AMO demonstrates exceptional linear and angular velocity tracking due to its hierarchical design combining trajectory optimization with RL. AMO and ULC achieves competitive performance through unified whole-body control, while HOMIE and FALCON show moderate performance due to decoupled architectures.

Torso Orientation Control: ULC, HOMIE-3-DoF-Waist, and AMO support full 3-DoF torso control. ULC excels in yaw tracking and achieves superior pitch and roll control. AMO shows competitive yaw but higher pitch and roll errors. HOMIE-3-DoF-Waist relies entirely on PD control, resulting in degraded accuracy. HOMIE and FALCON lack pitch/roll capabilities, while R^{2}S^2 provides pitch and yaw control but lacks roll control capability.

Dual-Arm Tracking: HOMIE and AMO rely on PD controllers, achieving moderate performance. FALCON's upper-body RL policy achieves better tracking but remains constrained by MoCap dependency. ULC outperforms all methods through residual action modeling and sequential skill acquisition.

Robustness Under Extreme Conditions: In edge command space scenarios, HOMIE-3-DoF-Waist suffers severe degradation due to inadequate coordination between PD-controlled torso and RL-controlled legs. ULC maintains robust performance owing to its unified architecture.

External Load Adaptation: Under 2kg wrist loads, AMO maintains its locomotion advantage, but PD-based arm control shows noticeable degradation. FALCON's force-adaptive curriculum provides partial load robustness, while ULC achieves superior load adaptation across all metrics.

Command Mutation Robustness: Under stochastic command delays, ULC demonstrates superior robustness, while other methods show significant deterioration. AMO experiences substantial degradation in torso control, and HOMIE variants suffer severe performance loss.

D. Ablation on Policy Training

We evaluate the contribution of each key component by removing one module at a time: Sequential Skill Acquisition, Residual Action Model, Load Randomization, and center of mass Tracking. The results are visualized in Table V, with evaluation performed under 2kg wrist load.

Removing Sequential Skill Acquisition leads to significant degradation in torso orientation control (E_p: +100%, E_r: +100%), confirming the importance of progressive skill composition. Excluding the Residual Action Model causes the largest arm tracking degradation (E_a: +140%), validating its role in fine-grained upper body control. Without Load Randomization, locomotion and arm tracking deteriorate notably

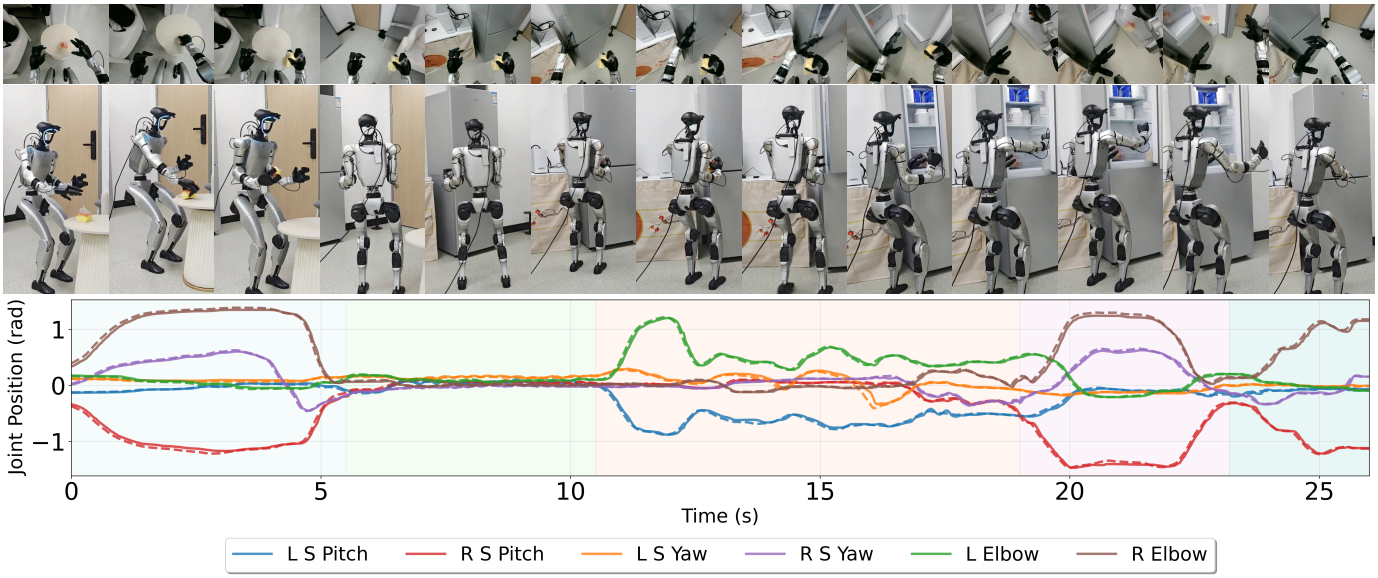

Fig. 4: Time-series visualization of the refrigerator task, covering all five stages: picking up the bread, walking to the refrigerator, opening the door, placing the bread inside, and closing the door.

under payload. Omitting center of mass Tracking results in the most severe balance degradation (E_v: +10%, E_ω: +38%, E_h: +100%), demonstrating its essential role in dynamic stability.

The full ULC model achieves the lowest errors across all metrics, validating that each component is indispensable for robust, high-precision loco-manipulation.

Method	E_v (m/s)	E_ω (rad/s)	E_h (m)	E_y (rad)	E_p (rad)	E_r (rad)	E_a (rad)
w/o Seq. Skill Acq.	0.11 ±0.02	0.17 ±0.02	0.05 ±0.01	0.10 ±0.02	0.14 ±0.02	0.12 ±0.02	0.09 ±0.02
w/o Residual Action	0.12 ±0.01	0.15 ±0.02	0.04 ±0.01	0.08 ±0.01	0.10 ±0.02	0.09 ±0.01	0.12 ±0.02
w/o Load Random.	0.11 ±0.02	0.16 ±0.02	0.04 ±0.01	0.08 ±0.01	0.09 ±0.01	0.08 ±0.01	0.10 ±0.02
w/o CoM Tracking	0.11 ±0.02	0.18 ±0.03	0.06 ±0.01	0.09 ±0.02	0.11 ±0.02	0.10 ±0.02	0.08 ±0.01
ULC (Ours)	**0.10** ±0.01	**0.13** ±0.02	**0.03** ±0.01	**0.06** ±0.01	**0.07** ±0.01	**0.06** ±0.01	**0.05** ±0.01

TABLE V: Ablation study on policy training components. Best results are in **bold**.

E. Real World Results

We evaluate ULC in real-world scenarios to validate how its height control, torso rotation capabilities, and dual-arm tracking precision contribute to practical task performance.

*1) **Teleoperation Results**:* We evaluate two representative teleoperation scenarios requiring coordinated locomotion and manipulation, as illustrated in Fig. 3 and Fig. 4.

Pick and place the doll on the sofa: This task evaluates ULC's ability to perform coordinated whole-body manipulation: (1) squatting down and grasping the doll using precise height control and torso pitch adjustment; (2) standing up and passing the doll to the other hand; (3) placing the doll onto the sofa. As shown in Fig. 3, the execution demonstrates ULC's locomotion stability during height transitions and dynamic balance during coordinated arm movements. The tracking curves show consistently low dual-arm tracking error throughout execution.

Put the bread in the refrigerator: This task demonstrates ULC's ability to execute a complex, multi-step sequence: (1) grasp the bread from the table; (2) walk to the refrigerator while maintaining a secure hold; (3) open the refrigerator door with the left hand; (4) place the bread inside with accurate positioning; (5) close the door after releasing the bread. Fig. 4 presents the full teleoperated process, with consistently low arm tracking error highlighting ULC's precision in practical force-interactive loco-manipulation scenarios.

VI. CONCLUSIONS AND LIMITATIONS

We presented ULC, a unified controller for humanoid loco-manipulation that simultaneously achieve unified whole-body control, large operational workspace, and high-precision tracking. By integrating all degrees of freedom in a single controller with principled procedural command sampling, ULC enables robust and versatile performance across diverse tasks and challenging scenarios. Extensive experiments demonstrate that ULC outperforms prior decoupled or MoCap-based methods in tracking accuracy, workspace coverage, and robustness. Ablation studies further confirm the necessity of each component in our framework.

A current limitation is that simplified locomotion commands preclude complex leg patterns achievable through motion capture approaches. Addressing this limitation to enhance locomotion expressiveness for more complex real-world tasks will be the focus of our future work.

REFERENCES

[1] Q. Ben, F. Jia, J. Zeng, J. Dong, D. Lin, and J. Pang, *HOMIE: Humanoid Loco-Manipulation with Isomorphic Exoskeleton Cockpit*, Feb. 2025. arXiv: 2502.13013 [cs].

[2] Y. Zhang et al., *FALCON: Learning Force-Adaptive Humanoid Loco-Manipulation*, May 2025. arXiv: 2505.06776 [cs].

[3] Z. Ding et al., *JAEGER: Dual-Level Humanoid Whole-Body Controller*, May 2025. arXiv: 2505.06584 [cs].

[4] J. Li, X. Cheng, T. Huang, S. Yang, R.-Z. Qiu, and X. Wang, *AMO: Adaptive Motion Optimization for Hyper-Dexterous Humanoid Whole-Body Control*, May 2025. arXiv: 2505.03738 [cs].

[5] Y. Li et al. "Hold My Beer: Learning Gentle Humanoid Locomotion and End-Effector Stabilization Control." arXiv: 2505.24198 [cs], Accessed: Jul. 4, 2025. [Online]. Available: http://arxiv.org/abs/2505.24198, pre-published.

[6] Z. Zhang et al., *Unleashing Humanoid Reaching Potential via Real-world-Ready Skill Space*, May 2025. arXiv: 2505.10918 [cs].

[7] W. Sun, B. Cao, L. Chen, Y. Su, Y. Liu, and Z. Xie, "Learning Perceptive Humanoid Locomotion over Challenging Terrain,"

[8] T. He et al., *ASAP: Aligning Simulation and Real-World Physics for Learning Agile Humanoid Whole-Body Skills*, Feb. 2025. arXiv: 2502.01143 [cs].

[9] A. Allshire et al., *Visual Imitation Enables Contextual Humanoid Control*, May 2025. arXiv: 2505.03729 [cs].

[10] B. Qingwei et al., "Gallant: Voxel grid-based humanoid locomotion and local-navigation across 3d constrained terrains," *arXiv preprint arXiv:2511.14625*, 2025.

[11] C. Packer, K. Gao, J. Kos, P. Krähenbühl, V. Koltun, and D. Song, *Assessing generalization in deep reinforcement learning*, 2019. arXiv: 1810.12282 [cs.LG].

[12] Z. Zhuang et al., "Robot parkour learning," *arXiv preprint arXiv:2309.05665*, 2023. arXiv: 2309.05665.

[13] Z. Zhuang, S. Yao, and H. Zhao, *Humanoid Parkour Learning*, Sep. 2024. arXiv: 2406.10759 [cs].

[14] R. Zheng et al., *FLARE: Robot Learning with Implicit World Modeling*, May 2025. arXiv: 2505.15659 [cs].

[15] R.-Z. Qiu et al., *Humanoid Policy ˜ Human Policy*, Mar. 2025. arXiv: 2503.13441 [cs].

[16] T. Lin, K. Sachdev, L. Fan, J. Malik, and Y. Zhu, *Sim-to-Real Reinforcement Learning for Vision-Based Dexterous Manipulation on Humanoids*, Feb. 2025. arXiv: 2502.20396 [cs].

[17] C. Chen, Z. Yu, H. Choi, M. Cutkosky, and J. Bohg, *DexForce: Extracting Force-informed Actions from Kinesthetic Demonstrations for Dexterous Manipulation*, Jan. 2025. arXiv: 2501.10356 [cs].

[18] P. Intelligence et al., $\pi_{0.5}$*: A vision-language-action model with open-world generalization*, 2025. arXiv: 2504.16054 [cs.LG].

[19] NVIDIA et al., *Gr00t n1: An open foundation model for generalist humanoid robots*, 2025. arXiv: 2503.14734 [cs.RO].

[20] C. Chi et al., *Diffusion policy: Visuomotor policy learning via action diffusion*, 2024. arXiv: 2303.04137 [cs.RO].

[21] T. Z. Zhao, V. Kumar, S. Levine, and C. Finn, *Learning fine-grained bimanual manipulation with low-cost hardware*, 2023. arXiv: 2304.13705 [cs.RO].

[22] K. Black et al., π_0*: A vision-language-action flow model for general robot control*, 2024. arXiv: 2410.24164 [cs.LG].

[23] T. He et al., "HOVER: Versatile Neural Whole-Body Controller for Humanoid Robots," *arXiv preprint arXiv:2410.21229*, 2024. arXiv: 2410.21229.

[24] T. He et al., "OmniH2O: Universal and Dexterous Human-to-Humanoid Whole-Body Teleoperation and Learning," *arXiv preprint arXiv:2406.08858*, 2024. arXiv: 2406.08858.

[25] Z. Fu, Q. Zhao, Q. Wu, G. Wetzstein, and C. Finn, "HumanPlus: Humanoid Shadowing and Imitation from Humans," *arXiv preprint arXiv:2406.10454*, 2024. arXiv: 2406.10454.

[26] Z. Zhuang and H. Zhao, *Embrace Collisions: Humanoid Shadowing for Deployable Contact-Agnostics Motions*, Feb. 2025. arXiv: 2502.01465 [cs].

[27] T. Zhang et al., *HuB: Learning Extreme Humanoid Balance*, May 2025. arXiv: 2505.07294 [cs].

[28] A. Allshire et al., "Visual imitation enables contextual humanoid control," *arXiv preprint arXiv:2505.03729*, 2025.

[29] Z. Chen, M. Ji, X. Cheng, X. Peng, X. B. Peng, and X. Wang, "Gmt: General motion tracking for humanoid whole-body control," *arXiv preprint arXiv:2506.14770*, 2025.

[30] Z. Zhang et al., "Track any motions under any disturbances," *arXiv preprint arXiv:2509.13833*, 2025.

[31] S. Zhao et al., "Resmimic: From general motion tracking to humanoid whole-body loco-manipulation via residual learning," *arXiv preprint arXiv:2510.05070*, 2025.

[32] Q. Liao, T. E. Truong, X. Huang, G. Tevet, K. Sreenath, and C. K. Liu, "Beyondmimic: From motion tracking to versatile humanoid control via guided diffusion," *arXiv preprint arXiv:2508.08241*, 2025.

[33] F. Jenelten, J. He, F. Farshidian, and M. Hutter, "DTC: Deep Tracking Control," *Science Robotics*, vol. 9, no. 86, eadh5401, Jan. 2024.

[34] Y. Xue, W. Dong, M. Liu, W. Zhang, and J. Pang, *A Unified and General Humanoid Whole-Body Controller for Fine-Grained Locomotion*, Feb. 2025. arXiv: 2502.03206 [cs].

[35] T. Silver, K. Allen, J. Tenenbaum, and L. Kaelbling, "Residual policy learning," *arXiv preprint arXiv:1812.06298*, 2018.

[36] T. Johannink et al., "Residual reinforcement learning for robot control," in *2019 international conference on robotics and automation (ICRA)*, IEEE, 2019, pp. 6023–6029.

[37] Z. Luo, J. Cao, A. Winkler, K. Kitani, and W. Xu, *Perpetual Humanoid Control for Real-time Simulated Avatars*, Sep. 2023. arXiv: 2305.06456 [cs].

[38] X. Gu et al., "Advancing humanoid locomotion: Mastering challenging terrains with denoising world model learning," *arXiv preprint arXiv:2408.14472*, 2024. arXiv: 2408.14472.

[39] 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. arXiv: 2404.05695.

[40] M. Ji et al., *ExBody2: Advanced Expressive Humanoid Whole-Body Control*, Dec. 2024. arXiv: 2412.13196 [cs].

[41] M. Mittal et al., "Orbit: A unified simulation framework for interactive robot learning environments," *arXiv preprint arXiv:2301.04195*, 2023. arXiv: 2301.04195.

APPENDIX

A. Communication Architecture

The teleoperation system employs a distributed communication architecture that coordinates multiple components across different computational nodes. Dual cameras mounted on the robot's onboard computer transmit stereo images to the host computer via TCP and ZeroMQ protocols. The host computer processes these images for VR visualization while receiving operator commands through a network router connection. Robot actuators (dexterous hands and joints) communicate bidirectionally with the host computer using DDS protocol, transmitting states and receiving action commands. The system uses an asynchronous architecture where the teleoperation solver processes VR inputs to generate robot commands, while a separate deployment module runs at 50Hz to continuously read and execute the latest commands. This design ensures responsive control while maintaining system modularity.

The complete communication architecture can be summarized as follows:

$$\text{Cameras} \xrightarrow{\text{TCP/ZeroMQ}} \text{Host} \xrightarrow{\text{Network}} \text{VR Headset} \tag{27}$$

$$\text{VR Headset} \xrightarrow{\text{Network}} \text{Host} \xrightarrow{\text{DDS}} \text{Solver} \tag{28}$$

$$\text{Solver} \xrightarrow{\text{DDS}} \text{Deployment} \xrightarrow{\text{DDS}} \text{Robot Actuators} \tag{29}$$

This distributed architecture enables scalable and responsive teleoperation while maintaining the modularity necessary for system development and debugging.

B. Domain Randomization

We use domain randomization to simulate the sensor noise and physical variations in the real-world. The randomization parameters are shown in Table VI.

Parameter	Unit	Range	Operator
Angular Velocity	rad/s	± 0.2	scaling
Projected Gravity	-	± 0.05	scaling
Joint Position	rad	± 0.01	scaling
Joint Velocity	rad/s	± 1.5	scaling
Static Friction	-	[0.7, 1.0]	uniform
Dynamic Friction	-	[0.4, 0.7]	uniform
Restitution	-	[0.0, 0.005]	uniform
Wrist Mass	kg	[0.0, 2.0]	additive
Base Mass	kg	[-5.0, 5.0]	additive

TABLE VI: Domain randomization parameters. Additive randomization adds a random value within a specified range to the parameter, while scaling randomization adjusts the parameter by a random multiplication factor within the range.

C. Real world Loaded Comparison

We conduct a controlled experiment comparing ULC with traditional PD control under external wrist loads of 0.5 kg, 1.0 kg, and 1.5 kg. ULC and PD controller share the same PD parameters (PD gains: $K_p = 80$, $K_d = 3$). Both methods are required to maintain dual-arm poses with target joint angles set to zero (forearms parallel to the ground). The tracking errors under each load condition are visualized in Fig. 5.

Across all load levels, ULC consistently achieves lower joint angle deviations than PD control. Notably, even at the highest load of 1.5 kg, ULC maintains high tracking accuracy, while PD control exhibits significant errors due to inadequate gravity compensation. This performance gap is evident at every tested load (0.5 kg, 1.0 kg, 1.5 kg), where ULC's learned dynamics naturally incorporate force adaptation, resulting in superior robustness and precision. In contrast, PD control struggles to maintain parallel positioning even without load, and its errors increase substantially as the load increases. These results validate the advantage of ULC in real-world manipulation tasks requiring reliable force adaptation and precise tracking under varying external disturbances.

D. Reward Function

Our reward function is a sum of the following terms:

- **Tracking Linear Velocity Reward** (r_{vel}): This term encourages the robot to track the commanded linear velocity in the xy-plane.

$$r_{vel} := \exp(-\|v_{xy} - v^*_{xy}\|_2^2 / \sigma^2_{vel}),$$

  where v_{xy} and v^*_{xy} represent the actual and commanded linear velocities, respectively. σ_{vel} is set to 0.5. Weight: 1.0.
- **Tracking Angular Velocity Reward** (r_{ang}): This term encourages the robot to track the commanded angular velocity.

$$r_{ang} := \exp(-\|\omega_z - \omega^*_z\|_2^2 / \sigma^2_{ang}),$$

  where ω_z and ω^*_z represent the actual and commanded angular velocities, respectively. σ_{ang} is set to 0.5. Weight: 1.25.
- **Root Height Tracking Reward** (r_{height}): This term encourages tracking of the commanded pelvis height.

$$r_{height} := \exp(-|h - h^*|^2 / \sigma^2_{height}),$$

  where h and h^* are the actual and commanded root heights. σ_{height} is set to 0.4. Weight: 1.0.
- **Upper Body Position Tracking Reward** (r_{upper}): This term encourages tracking of arm joint positions.

$$r_{upper} := \exp(-\|\boldsymbol{q}_{upper} - \boldsymbol{q}^*_{upper}\|_2^2 / \sigma^2_{upper}),$$

  where $\boldsymbol{q}_{upper}$ and $\boldsymbol{q}^*_{upper}$ are the actual and desired upper body joint positions. σ_{upper} is set to 0.35. Weight: 1.0.
- **Torso Yaw Tracking Reward** (r_{yaw}): This term encourages tracking of torso yaw orientation commands.

$$r_{yaw} := \exp(-e^2_{yaw} / \sigma^2_{torso}),$$

  where e_{yaw} is the yaw orientation error. σ_{torso} is set to 0.2. Weight: 0.25.
- **Torso Roll Tracking Reward** (r_{roll}): This term encourages tracking of torso roll orientation commands.

$$r_{roll} := \exp(-e^2_{roll} / \sigma^2_{torso}),$$

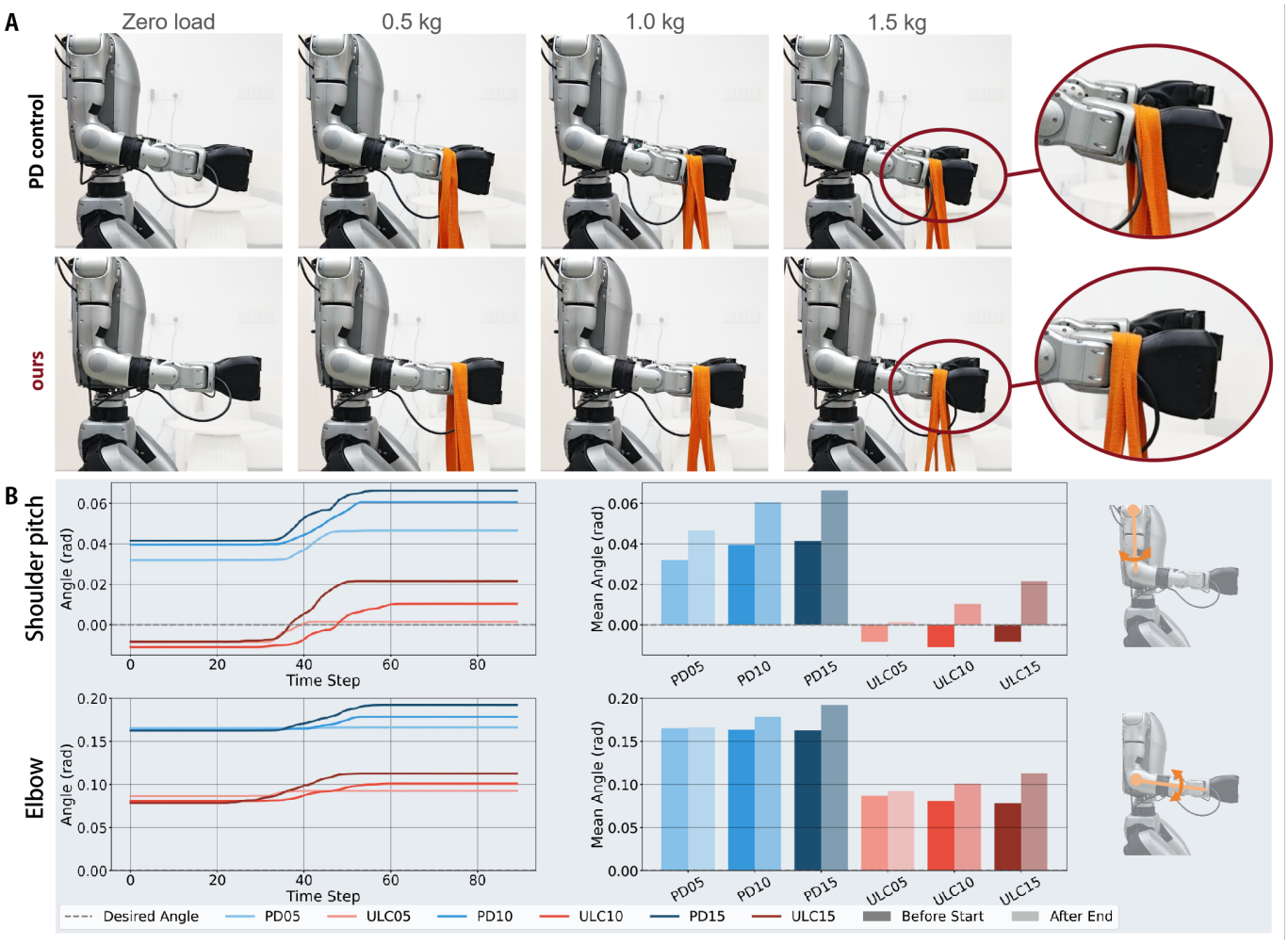

Fig. 5: Comparison of joint angle tracking errors for ULC and traditional PD control (gains: $K_p = 80$, $K_d = 3$) under different external loads (0.5 kg, 1.0 kg, 1.5 kg) in real-world experiments. ULC consistently achieves lower errors than PD control at all load levels, demonstrating superior force adaptation and robustness to external disturbances.

where e_{roll} is the roll orientation error. σ_{torso} is set to 0.2. Weight: 0.25.

- **Torso Pitch Tracking Reward** (r_{pitch}): This term encourages tracking of torso pitch orientation commands with higher weight.

$$r_{pitch} := \exp(-e_{pitch}^2/\sigma_{torso}^2),$$

where e_{pitch} is the pitch orientation error. σ_{torso} is set to 0.2. Weight: 0.5.

- **Center of Mass Tracking Reward** (r_{CoM}): This term maintains stability by keeping the center of mass near the support base.

$$r_{CoM} := \exp(-\|\boldsymbol{p}_{CoM}^{xy} - \boldsymbol{p}_{feet}^{xy}\|_2^2/\sigma_{CoM}^2),$$

where $\boldsymbol{p}_{CoM}^{xy}$ is the horizontal center of mass projection and $\boldsymbol{p}_{feet}^{xy}$ is the midpoint between ankles. σ_{CoM} is set to 0.2. Weight: 0.5.

- **Termination Reward**: This term penalizes episode termination.

$$r_{ter} := -200.0 \cdot \mathbb{I}_{\text{terminated}}$$

where $\mathbb{I}_{\text{terminated}}$ is 1 if the episode terminates, otherwise 0.

- **Z-axis Linear Velocity Reward**: This term penalizes the robot for moving along the z-axis.

$$r_z := -1.0 \cdot (v_z)^2$$

where v_z is the z-axis linear velocity.

- **Energy Reward**: This term penalizes output torques to reduce energy consumption.

$$r_e := -0.001 \cdot \sum_i |\tau_i \cdot \dot{q}_i|$$

where τ represents the joint torques and $\dot{q}$ represents the joint velocities.

- **Joint Acceleration Reward**: This term penalizes excessive joint accelerations to promote smooth motions.

$$r_{ja} := -2.5 \times 10^{-7} \cdot \|\ddot{q}\|_2^2$$

where $\ddot{q}$ represents the joint accelerations of the configured joints.

- **Action Rate Reward**: This term penalizes rapid changes in actions to encourage smooth control.

$$r_{ar} := -0.1 \cdot \|a_t - a_{t-1}\|_2^2$$

where a_t represents the current action and a_{t-1} represents the previous action.
- **Base Orientation Reward**: This term penalizes non-flat base orientation to maintain an upright posture.

$$r_{ori} := -5.0 \cdot (\text{roll}^2 \cdot \text{mask}_{roll} + \text{pitch}^2 \cdot \text{mask}_{pitch})$$

where mask_{roll} and mask_{pitch} are adaptive masks based on torso command magnitudes.
- **Joint Position Limit Reward**: This term penalizes joint positions that exceed their soft limits.

$$r_{\text{jpl}} := -2.0 \cdot \sum_i \max(|q_i| - q_{i,\text{limit}}, 0)$$

where q_i represents the position of joint i, and $q_{i,\text{limit}}$ is the soft limit.
- **Joint Effort Limit Reward**: This term penalizes excessive torques on waist joints.

$$r_{\text{jel}} := -2.0 \cdot \sum_i \max(|\tau_i| - 0.999 \cdot \tau_{i,\max}, 0)$$

where τ_i is the torque and $\tau_{i,\max}$ is the maximum torque limit.
- **Joint Deviation Reward**: This term penalizes joint positions that deviate from their default positions.

$$r_{jd} := -0.15 \cdot \sum_i |q_i - q_{i,\text{default}}| - 0.3 \cdot \sum_j |q_j - q_{j,\text{default}}|$$

where i represents hip yaw and ankle roll joints, and j represents hip roll joints.
- **Feet Air Time Reward**: This term rewards appropriate stepping behavior for bipedal locomotion.

$$r_{\text{fat}} := 0.3 \cdot \min(t_{\text{air}}, 0.4)$$

where t_{air} is the air time when exactly one foot is in contact and velocity command is above 0.1 m/s.
- **Feet Slide Reward**: This term penalizes feet sliding during ground contact.

$$r_{\text{sl}} := -0.25 \cdot \sum_i \|v_{i,xy}\|_2 \cdot \mathbb{I}(\text{contact}_i)$$

where $v_{i,xy}$ is the horizontal velocity of foot i, and $\mathbb{I}(\text{contact}_i)$ indicates if the foot is in contact.
- **Feet Force Reward**: This term encourages maintaining appropriate ground reaction forces.

$$r_{\text{ff}} := -3 \times 10^{-3} \cdot \sum_i \min(\max(f_{z,i} - 500, 0), 400)$$

where $f_{z,i}$ is the vertical ground reaction force on foot i.
- **Feet Stumble Reward**: This term penalizes lateral forces that indicate stumbling.

$$r_{\text{fs}} := -2.0 \cdot \sum_i \mathbb{I}(\|f_{xy,i}\|_2 > 5|f_{z,i}|)$$

where $f_{xy,i}$ represents the horizontal ground reaction forces.
- **Flying State Reward**: This term penalizes the robot when it is airborne.

$$r_{\text{fly}} := -1.0 \cdot \mathbb{I}(\text{all feet off ground})$$

- **Undesired Contacts Reward**: This term penalizes undesired contacts with the environment.

$$r_{\text{uc}} := -1.0 \cdot \sum_{i \in \mathcal{C}} \mathbb{I}\left(\|\mathbf{F}_i\|_2 > 1.0\right)$$

where $\mathcal{C}$ represents the set of contact points excluding ankle contacts.
- **Ankle Orientation Reward**: This term penalizes excessive ankle roll orientations.

$$r_{\text{ankle}} := -0.5 \cdot \sum_i \|\text{gravity}_{xy,i}\|_2^2$$

where $\text{gravity}_{xy,i}$ is the projected gravity vector in each ankle frame.

E. ULC Hyperparameters

We illustrate the hyperparameters of ULC in Table VII.

Parameter	Value
Number of Environments	8192
Training Iteration	10000
Environment Steps	24
Number of Training Epochs	5
Mini Batch Size	4
Max Clip Value Loss	0.2
Discount Factor	0.99
GAE discount factor	0.95
Entropy Regularization Coefficient	0.006
Learning rate	1.0e-3
Schedule	adaptive
Desired KL	0.01
Max Grad Norm	1.0
Value Loss Coefficient	1.0
Observation History Length	6
Action Scale	0.25
Episode Length	20.0 s
Simulation Timestep	0.005 s
Control Decimation	4

TABLE VII: Hyperparameters of ULC.

F. Architecture Details

Table VIII illustrates the network architecture of ULC.

G. ULC Training Curves

We illustrate the training curves of ULC in Figure 6, showing the reward convergence over the first 10,000 training iterations. The training process is divided into four distinct stages corresponding to the sequential skill acquisition with adaptive curriculum:

1) **Stage 1: Base velocity tracking initialization** (T_1 active, $\alpha_2 = 0$, $\alpha_3 = 0$): The policy learns fundamental locomotion skills by tracking base linear and angular

velocity commands (v_{xy}, ω_z) without additional curriculum constraints.

2) **Stage 2: Height tracking curriculum activation** (T_1 and T_2 active, α_2 increasing): Once the height curriculum advancement condition $\mathcal{C}_2$ is met, the base height tracking skill T_2 is activated. The curriculum parameter α_2 progressively increases from 0 to 1.0, enabling the policy to learn pelvis height control (h_{pelvis}) while maintaining velocity tracking performance.
3) **Stage 3: Upper body tracking curriculum activation** (T_1, T_2, and T_3 active, α_3 increasing): When $\alpha_2 \geq 0.98$ and the upper body curriculum advancement condition $\mathcal{C}_3$ is satisfied, the torso and arm tracking skill T_3 is introduced. The curriculum parameter α_3 increases from 0 to 1.0, allowing the policy to master torso orientation ($\boldsymbol{g}_{\text{torso}}$) and arm joint position ($\boldsymbol{g}_{\text{arms}}$) commands.
4) **Stage 4: Full curriculum completion and convergence** (T_1, T_2, and T_3 active, $\alpha_2 = 1.0$, $\alpha_3 = 1.0$): All curriculum parameters reach their maximum values, enabling full command space exploration. The policy enters the final training phase with complete skill integration, leading to reward convergence across all tracking objectives.

This staged progression ensures stable learning and prevents catastrophic forgetting, with each stage building upon the competencies acquired in previous phases.

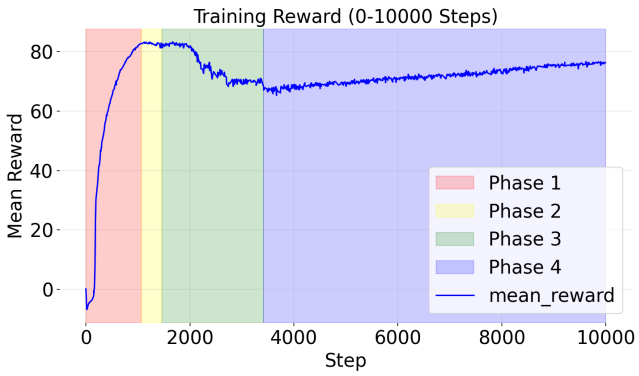

Fig. 6: Reward convergence curve during ULC training (first 10,000 iterations), showing four distinct stages of curriculum progression.

Component	Configuration
Actor Network	
Input Layer	Observation (History × Features)
Hidden Layer 1	Linear(Input → 1024) + ELU
Hidden Layer 2	Linear(1024 → 512) + ELU
Hidden Layer 3	Linear(512 → 512) + ELU
Hidden Layer 4	Linear(512 → 256) + ELU
Output Layer	Linear(256 → 29)
Critic Network	
Input Layer	Observation (History × Features)
Hidden Layer 1	Linear(Input → 1024) + ELU
Hidden Layer 2	Linear(1024 → 512) + ELU
Hidden Layer 3	Linear(512 → 512) + ELU
Hidden Layer 4	Linear(512 → 256) + ELU
Output Layer	Linear(256 → 1)
Policy Distribution	
Distribution Type	Gaussian
Initial Noise Std	1.0
Noise Type	log

TABLE VIII: ULC network architecture details.