Generalizing from References using a Multi-Task Reference and Goal-Driven RL Framework

Jiashun Wang[1,2] M. Eva Mungai[1] He Li[1] Jean Pierre Sleiman[1] Jessica Hodgins[1,2] Farbod Farshidian[1]
[1]RAI Institute [2]Carnegie Mellon University

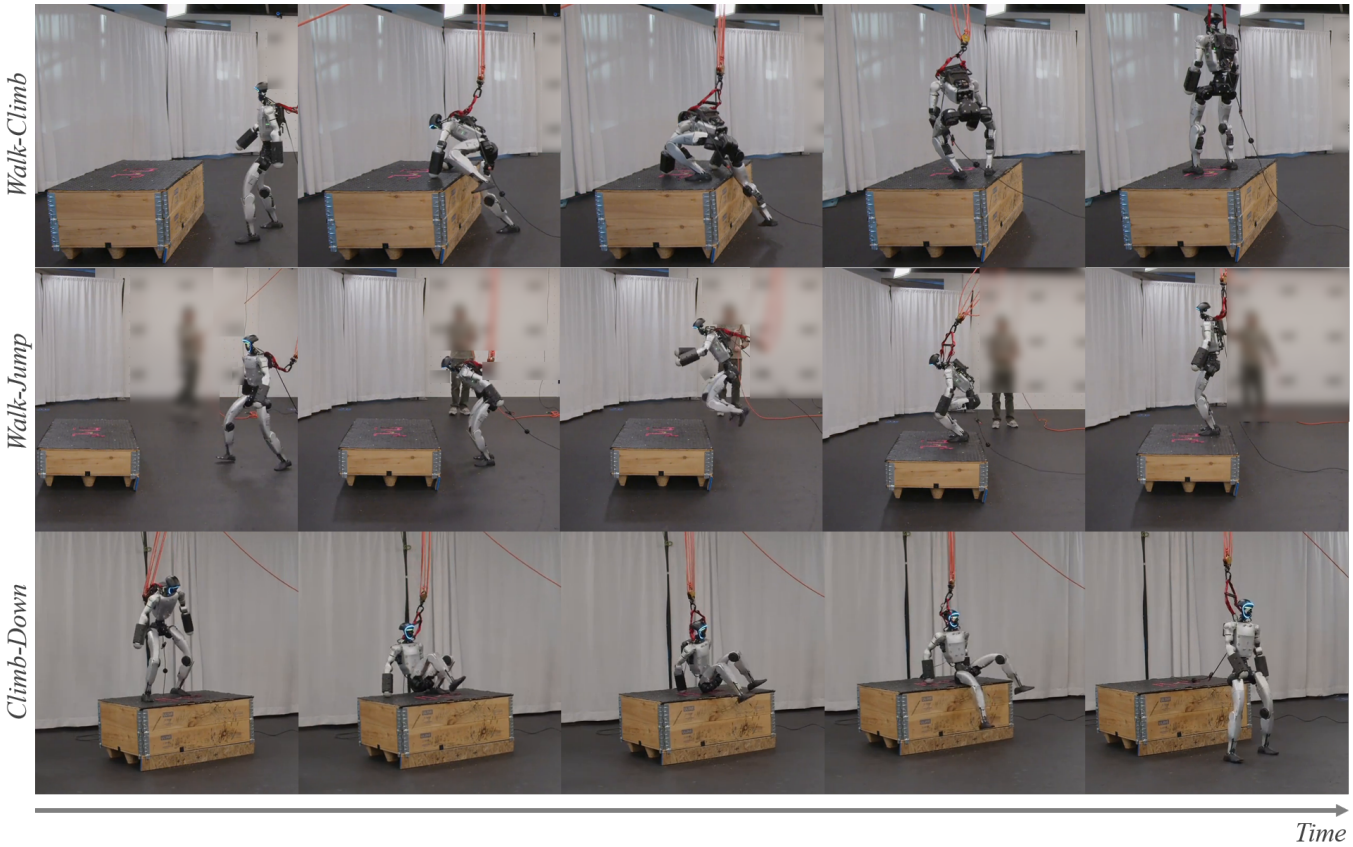


Fig. 1: A humanoid robot performs human-like walking, jumping, and climbing behaviors in a box-based environment.

***Abstract*—Learning agile humanoid behaviors from human motion offers a powerful route to natural, coordinated control, but existing approaches face a persistent trade-off: reference-tracking policies are often brittle outside the demonstration dataset, while purely task-driven Reinforcement Learning (RL) can achieve adaptability at the cost of motion quality. We introduce a multi-task RL training paradigm that bridges this gap by treating reference motion as a prior for behavioral shaping rather than a deployment-time constraint. A goal-conditioned policy is trained jointly on two tasks that share the same observation and action spaces, but differ in their initialization schemes, command spaces, and reward structures: (i) a reference-guided imitation task in which reference trajectories define dense imitation rewards but are not provided as policy inputs, and (ii) a goal-conditioned generalization task in which goals are sampled independently of any reference and where rewards reflect only task success. By co-optimizing these objectives within a shared observation space, the policy acquires structured, human-like motor skills from dense reference supervision while learning to adapt these skills to novel goals and initial conditions. This is achieved without adversarial objectives, explicit trajectory tracking, phase variables, or reference-dependent inference. We evaluate the method on a challenging box-based parkour playground that demands diverse athletic behaviors (e.g., jumping and climbing), and show that the learned controller transfers beyond the reference distribution while preserving motion naturalness. Finally, we demonstrate that the learned skills can be composed in long-horizon scenarios using a simple state-machine-based evaluation protocol, highlighting their robustness and ability to generalize across diverse task conditions. Results are best visualized through https://jiashunwang.github.io/GfR/.**

I. Introduction

Deep reinforcement learning (DRL) has become a central tool for synthesizing feedback controllers for humanoid robots, enabling policies that can robustly execute task objectives under complex contacts and high-dimensional dynamics. Much of this progress has been driven by goal-oriented formulations, in which policies are trained to directly optimize some

task-centric reward. However, achieving complex, contact-rich skills while avoiding unintended emergent behaviors often requires heavy reward shaping and extensive iteration.

In parallel, motion imitation has emerged as a powerful paradigm for learning expressive and coordinated whole-body behaviors by leveraging motion capture data or reconstructed human motion. By providing a strong behavioral prior, imitation-based reinforcement learning can produce agile and human-like skills. However, controllers trained around fixed reference motions can reduce flexibility in steerable or target-driven settings, where the robot must react to new goals, environments, or task conditions, and supplying suitable trajectories across diverse situations becomes impractical.

These two perspectives expose a persistent trade-off: controllers that are tightly guided by reference behaviors can struggle to generalize beyond demonstrated scenarios, while controllers optimized primarily for task objectives may lose motion quality and style. Below, we summarize representative reference-driven RL-based approaches and highlight why closing this gap remains challenging.

Motion tracking. Early reference-driven approaches rely on motion-tracking objectives that encourage policies to closely follow reference trajectories, enabling stable reproduction of complex whole-body motions in simulation and on real robots [19, 25, 11, 4, 36, 35, 37]. *DeepMimic* [19] is an early reference-guided RL framework that learns robust physics-based character skills by optimizing an imitation reward over motion clips, producing agile behaviors that can recover from perturbations and can be combined with simple task objectives. More recently, variants of this paradigm have explicitly targeted sim-to-real transfer to deploy such behaviors on physical systems. For instance, *GMT* [4] scales whole-body motion tracking to large, diverse motion datasets and demonstrates transfer to a physical humanoid (Unitree's G1 robot). *ZEST* [25] streamlines sim-to-real motion imitation across heterogeneous reference sources (mocap, monocular video, and animation) using adaptive sampling and an automatic assistive-wrench curriculum, enabling zero-shot deployment of long-horizon, contact-rich skills across multiple robot embodiments. While these methods demonstrate that faithful imitation of human motion data can produce agile, highly coordinated behaviors, the learned controllers remain fundamentally coupled to the demonstrations, limiting their ability to generalize to novel goals, environments, or task conditions.

Distillation. To relax strict tracking and improve adaptability and steerability, several approaches combine motion tracking with distillation [10, 1, 8, 13, 27]. In these formulations, tracking-based controllers act as teachers, and their behaviors are distilled into policies that no longer require reference motion at execution time, allowing them to be directed by alternative commands or target specifications. Such approaches have been explored in applications including teleoperation [10, 14], locomotion [8, 12], and loco-manipulation [38]. However, because distilled policies inherit their behavior from motion-tracking controllers trained on a fixed set of reference motions, they remain strongly constrained by the reference data used during training, making it difficult to handle out-of-distribution task conditions or environments that differ significantly from the demonstration data. In many such pipelines, the distillation stage primarily provides a strong initialization, but achieving robustness on downstream, task-driven objectives still requires additional RL fine-tuning and task-specific reward design. This results in a multi-stage process—e.g., training a teacher, distilling to a student, and then optionally applying additional RL fine-tuning for a target task—rather than directly optimizing a single policy for the final objective.

Adversarial imitation. Another line of work adopts Adversarial Imitation Learning (AIL) to relax strict trajectory tracking by matching the distributions of generated motion and reference data [20, 23, 16, 29, 26, 18, 21, 31, 6]. These methods typically combine task rewards with imitation rewards learned through discriminators. By operating at the distribution level, AIL approaches allow greater flexibility than step-by-step tracking. However, learning a stable imitation objective becomes particularly challenging in dynamic or contact-rich settings, where small deviations in contact timing and configuration can induce large, discontinuous changes in system behavior. In such regimes, matching motion distributions does not reliably translate to physically valid or robust behavior, and conflicts between task and imitation rewards can undermine both motion quality and task performance.

Other formulations. Beyond these categories, several works explore alternative formulations for incorporating reference motion. SONIC [15] scales motion tracking to large motion dataset and introduces encoders that map multimodal conditions to kinematic plans. Wen et al. [34] propose alternating between tracking-based and style-based rewards to trade off task success and motion consistency. Hybrid Imitation Learning (HIL) [32] adopts a similar joint training approach, combining motion tracking with adversarial imitation learning to balance motion fidelity and task performance. However, these adversarial objectives can be difficult to stabilize and tune for hardware. In addition, reference-free tracking is particularly challenging without a proper curriculum.

Overall, most existing reference-driven approaches remain fundamentally constrained by how the reference motion is used: as a trajectory to be followed, a teacher to be distilled, or a distribution to be matched. These formulations make it difficult to decouple motion quality from the specific reference data and to robustly adapt learned behaviors to novel task conditions that fall outside the training distribution.

To address this gap, we propose a multi-task RL training paradigm that combines reference imitation with goal-conditioned Reinforcement Learning (RL) to learn motor behaviors that can be adapted to new task conditions. Rather than treating reference tracking as the ultimate objective, we use it as a form of behavioral shaping, providing a strong inductive bias toward natural and coordinated motion. We train a goal-conditioned policy across two tasks that share a common observation space. In the reference-guided imitation task, the policy does not receive reference motions as input. Instead,

reference motions are used only to define goal conditions and to shape imitation rewards. Consequently, the same policy is trained on a goal-conditioned generalization task in which goals are randomly specified without any reference motion, and rewards are defined purely by task objectives, requiring the policy to adapt its behavior to reach the target. By jointly optimizing both tasks within a shared observation space, the policy acquires motor behaviors that are structured by reference motion yet transferable to novel goals. In particular, this multi-task formulation facilitates transfer from imitation to general task learning when the downstream objective can be achieved by adapting behaviors learned during imitation.

We study this question in the context of whole-body mobility over challenging terrain and obstacles. Rough-terrain traversal has become a key testbed for evaluating humanoid motor capabilities, and many recent works have demonstrated impressive adaptability in such complex scenarios [5, 9, 30, 39, 29, 3, 7]. However, the resulting behaviors are primarily locomotion-centric, with the core motor behavior remaining walking and adaptation achieved through continuous adjustment of foot placement and body posture. In this work, we focus on extending beyond locomotion-centric terrain traversal toward agile parkour behaviors shaped by human data. While our hardware experiments rely on motion capture (MoCap) for global state (pose) feedback, we believe the same framework can be readily extended to incorporate onboard exteroceptive sensing. To evaluate this capability, we design a box-based parkour playground in which the humanoid must navigate obstacles using a range of athletic skills, including walking, jumping, climbing, and turning. We learn a diverse set of motion skills (as shown in Fig. 1) and study their composition in parkour-style box environments based on the box layout. For these long-horizon scenarios, we use a simple rule-based state machine to provide task-level goals. This setup highlights that the learned behaviors are robust and reusable: the robot can execute extended parkour sequences without careful tuning of initial conditions or task-specific resets, demonstrating reliable performance across varied environments.

II. Methodology

A. Overview

We study the problem of learning humanoid controllers for challenging dynamic tasks that require both human-like movements and adaptation to changing environments. Specifically, we consider a parkour environment constructed from boxes, where a humanoid must combine skills such as walking, jumping, and climbing to navigate varied obstacle layouts. To address this challenge, we train goal-conditioned policies in a multi-task setup that combines (i) reference-guided imitation and (ii) goal-driven generalization. The resulting policy does not rely on reference motions at inference time; instead, it conditions only on the current state and a goal specification.

In the motion-imitation task, reference motions are used to construct goal conditions and define tracking rewards, providing dense supervision that shapes the policy toward stable and human-like behaviors. In the generalization task, goals are specified independently of reference motion, and rewards are defined purely by task objectives. Jointly optimizing these two tasks allows the policy to acquire natural motor skills through imitation while learning to adapt and extend these skills across diverse conditions.

B. Design Rationale

The reference-guided imitation task serves purposes beyond enforcing stylistic qualities. In our learning framework, it plays two complementary roles:

1) **Representation learning.** The imitation task encourages the policy to learn a structured mapping between goal conditions and motor behaviors. Because the policy is conditioned only on goal information rather than explicit reference trajectories or phase variables, the resulting behaviors are not tied to individual demonstrations. Instead, the policy learns to represent a family of motor skills, enabling reuse and adaptation.
2) **Training stability and efficiency.** The imitation task provides dense reward signals that directly supervise how the humanoid should move, encoding rich information about pose coordination and timing derived from human motion data. These dense, human demonstration-based rewards provide stronger, more informative learning signals than task-level objectives alone, enabling the policy to efficiently acquire high-quality motor skills.

Overall, our formulation yields a single goal-conditioned policy that combines natural motion with goal-driven adaptability, without relying on adversarial discriminators, explicit trajectory tracking, or reference motions at inference time.

C. Training Setup and MDP Formulation

In our goal-conditioned RL framework, both the imitation and the generalization tasks are formulated in a unified setting that shares the same policy parameters, observation, and action spaces. The two tasks differ only in how goals, rewards, and value estimation are defined. The following presents the Markov Decision Process (MDP) formulation and the training setup of the proposed method.

Observation and goal representation. At each timestep, the policy observes the robot state s_t, which includes the humanoid's joint configuration and velocity, projected gravity, and torso angular velocities in a character-centric coordinate frame. The policy is also conditioned on a goal variable g_t that specifies the motion's target. Importantly, the policy does not observe reference motions, future trajectories, or explicit phase variables. All task-specific information is conveyed through the goal condition, ensuring a consistent observation space across training tasks. The goal variable g_t is a target root (torso) location in the horizontal plane, represented as a 2D position (x, y) relative to the character. In the imitation task, the goal, g_t, is derived from the references. In the generalization task, the goal is randomly sampled, independently of any reference motion.

Action space. The humanoid is actuated using joint-level Proportional-Derivative (PD) controllers. The policy outputs

residual actions $\boldsymbol{a}_t$, which are mapped to commanded joint position targets and converted to joint torques:

$$\boldsymbol{q}_j^{\mathrm{cmd}} = \bar{\boldsymbol{q}} + \boldsymbol{\Sigma}\,\boldsymbol{a}_t,$$

where $\bar{\boldsymbol{q}}$ denotes a set of default joint positions and $\boldsymbol{\Sigma}$ is a positive-definite diagonal matrix of per-DoF action scales. The PD gains are specified according to the procedure in [25], modeling each joint as an independent second-order system. Unlike standard motion-tracking formulations that use reference joint positions as a feedforward term in the command to stabilize and accelerate learning, we omit this term to preserve the ability to deviate from the reference when transitioning to task-driven objectives.

Reward design. The imitation task and the generalization task differ in their reward definitions. In the imitation task, the reward encourages the character to match the reference motion, providing dense supervision on how natural human motion should be executed. The total imitation reward consists of three components: a tracking reward, a regularization reward, and a survival reward:

$$r_t^{\mathrm{imi}} \;=\; r_{\mathrm{track},t} \;+\; r_{\mathrm{reg},t} \;+\; r_{\mathrm{surv},t}. \tag{1}$$

The tracking reward encourages the policy to follow the reference motion and provides dense behavioral shaping. It measures the discrepancy between the current state and the reference in terms of base pose, base velocity, and joint configurations:

$$r_{\mathrm{track}} \;=\; \sum_i c_{t_i} \exp\!\Big(-\kappa \frac{\|\mathbf{e}_i\|^2}{\sigma_i^2}\Big), \tag{2}$$

where $\mathbf{e}_i$ denotes the tracking error for term i, σ_i is a normalization scale, c_{t_i} is a weighting coefficient, and κ is a temperature parameter. The regularization reward aggregates penalties that promote physically plausible and smooth behavior, including action smoothness, joint torque usage, and violations of joint limits. The survival reward is a constant positive term provided at each timestep to discourage premature termination and encourage longer, stable rollouts.

For the generalization task, the reward no longer depends on reference motion and instead combines a sparse goal-driven task reward with the same regularization and survival terms as in the imitation task:

$$r_t^{\mathrm{gen}} \;=\; r_{\mathrm{goal},t} \;+\; r_{\mathrm{reg},t} \;+\; r_{\mathrm{surv},t}. \tag{3}$$

The goal-driven reward encourages progress toward and achievement of task-specific objectives, such as reaching a target base position and orientation, and is intentionally sparse compared to the imitation tracking reward. In general, we instantiate it as a weighted combination of progress and completion terms:

$$r_{\mathrm{goal},t} = -c_p \left\|\mathbf{e}_t^{xy}\right\|_2^2 - c_o \left\|\boldsymbol{\theta}_t\right\|_2^2 + r_{\mathrm{reach},t}, \tag{4}$$

where $\mathbf{e}_t^{xy} = \mathbf{p}_t^{xy} - \mathbf{p}_{goal}^{xy}$ denotes the base position distance in the horizontal plane, and $\boldsymbol{\theta}_t = \mathrm{AxisAngle}\big(\boldsymbol{\Phi}_t \otimes (\boldsymbol{\Phi}_{goal})^{-1}\big)$ represents the base orientation error computed from the axis–angle magnitude of the quaternion difference between the current and target base orientations. The weights c_p and c_o control the relative weights. The constant reward term $r_{\mathrm{reach},t}$ is activated only when the robot reaches the goal.

As a result, the policy relies on behaviors shaped during imitation and adapts them to satisfy the task objectives under varying conditions. The regularization and survival rewards serve the same role as in the imitation task. Further details regarding the rewards are provided in the appendix.

Curriculum. Building on prior work on assistive-wrench curricula for RL-based motion tracking [25], we first introduce a virtual assistive wrench to stabilize early-stage learning and enable the learning of highly dynamic behaviors. In addition, to improve generalization beyond reference-guided motion, we incorporate a task-level curriculum that gradually shifts training from imitation to goal-conditioned generalization. Together, these components form a unified curriculum that jointly regulates both physical assistance and the balance between imitation and generalization tasks.

Concretely, we maintain a global scalar curriculum variable $\lambda \in [0,1]$ that is updated online based on a tracking similarity score s. Specifically, $\lambda = \mathrm{clip}(s/s_{\max}, 0, 1)$, where higher values of λ correspond to better tracking performance. This scalar controls both (a) the magnitude of a virtual spatial assistive wrench applied at the base and (b) the probability of sampling the reference-guided imitation task versus the goal-conditioned generalization task.

Virtual wrench computation. Let $(\mathbf{p}, \mathbf{v}, \boldsymbol{\Phi}, \boldsymbol{\omega})$ denote the base position, linear velocity, orientation, and angular velocity, and let $(\hat{\mathbf{p}}, \hat{\mathbf{v}}, \hat{\boldsymbol{\Phi}}, \hat{\boldsymbol{\omega}})$ denote their reference counterparts. We compute a nominal spatial wrench at the base using a PD term on the base pose tracking error, together with a feedforward component that compensates nominal torso dynamics:

$$\mathbf{F}_b \;=\; M\Big(\hat{\dot{\mathbf{v}}} \;+\; k_p^v\,(\hat{\mathbf{p}} - \mathbf{p}) \;+\; k_d^v\,(\hat{\mathbf{v}} - \mathbf{v}) \;-\; \mathbf{g}\Big), \tag{5a}$$

$$\begin{aligned}\mathbf{M}_b \;=\; \mathbf{I}\,\hat{\dot{\boldsymbol{\omega}}} \;+\; k_p^{\omega}\,\mathbf{I}\,\big(\hat{\boldsymbol{\Phi}} \ominus \boldsymbol{\Phi}\big) \;+\; k_d^{\omega}\,\mathbf{I}\,(\hat{\boldsymbol{\omega}} - \boldsymbol{\omega}) \\ +\; \boldsymbol{\omega} \times (\mathbf{I}\boldsymbol{\omega}) \;-\; \mathbf{r}_{\mathrm{b,com}} \times M\mathbf{g},\end{aligned} \tag{5b}$$

where M and $\mathbf{I}$ are the whole-body mass and nominal base inertia at a default configuration, $\mathbf{g}$ is gravity, $\mathbf{r}_{\mathrm{b,com}}$ is the position of the whole-body CoM with respect to the base, and $\ominus$ denotes the Lie-group difference on $SO(3)$. The applied assistive wrench is

$$\mathbf{w}_e \;=\; \beta(\lambda)\begin{bmatrix}\mathbf{F}_b \\ \mathbf{M}_b\end{bmatrix}, \qquad \beta(\lambda) \;=\; (1-\lambda)\,\beta_{\max}, \quad \beta_{\max} < 1,$$

so assistance is strong during the early stage of training ($\lambda \approx 0$) and vanishes as training progresses ($\lambda \to 1$), while remaining partial.

Task mixing. The same scalar λ governs the transition from pure imitation to mixed training via a linear interpolation of the imitation sampling probability:

$$p_{\mathrm{imi}}(\lambda) \;=\; (1-\lambda)\,p_0 \;+\; \lambda\,p_{\mathrm{target}}, \quad p_0 > p_{\mathrm{target}},$$

Algorithm 1 Multi-Task Training with Coupled Curriculum

Initialize policy π_θ and curriculum scalar $\lambda \leftarrow 0$
for each training iteration **do**
  Compute tracking similarity score s
  $\lambda \leftarrow \text{clip}(s/s_{\max}, 0, 1)$
  $\beta \leftarrow (1-\lambda)\,\beta_{\max}$
  $p_{\text{imi}} \leftarrow (1-\lambda)\,p_0 + \lambda\,p_{\text{target}}$
  Sample task $k \sim \text{Bernoulli}(p_{\text{imi}})$
  if $k =$ imitation **then**
    Sample state from reference motion
    Set goal from reference
    Compute imitation reward r^{imi}
  else
    Sample initial state and goal randomly
    Compute task reward r^{gen}
  end if
  Apply base wrench scaled by β
  Collect rollouts and update π_θ
end for

with p_0 and p_{target} denoting the imitation-task sampling probabilities at the lowest and highest difficulty, respectively. In parallel, we expand the range of training configurations (initial states and task goals) as λ increases.

The training procedure, including curriculum updates and task sampling, is summarized in Algorithm 1. Overall, this curriculum allows the policy to first acquire structured, coordinated motor skills under strong stylistic regularization and partial physical assistance, and then progressively adapt these skills to broader task conditions as external support is removed. The curriculum scalar λ jointly governs both the level of physical assistance and the balance between imitation and generalization tasks during training.

State initialization and randomization. State initialization plays a critical role in our framework, as it guides how skills shaped in the motion-imitation task are transferred and generalized in the generalization task. We adopt different state and goal initialization strategies for the motion-tracking and generalization tasks. In the motion-tracking task, the humanoid's state and goal are initialized by sampling from reference motions with only small perturbations. This initialization strategy anchors the policy in reference-like configurations, ensuring that learning focuses on acquiring high-quality motor skills rather than executing in novel situations. In contrast, the generalization task adopts a much broader initialization strategy. Both the humanoid state and the goal are sampled from wide distributions, similar to standard RL settings. States are drawn from a more diverse set of configurations, and goals are specified randomly across different locations. This initialization strategy exposes the policy to conditions that deviate from the reference data, requiring it to adapt to novel goals and environments.

To further improve robustness on hardware, we apply domain randomization to some simulation parameters during training, including contact friction coefficients and link masses. Additionally, occasional external pushes are introduced to the torso, and noise is added to observations. These perturbations expose the policy to different dynamics and sensing conditions, helping it learn behaviors that remain stable when transferred to the hardware.

Training setup. Training is performed in Isaac Lab [17] utilizing Proximal Policy Optimization (PPO) [24] as the RL algorithm. We adopt an asymmetric actor-critic architecture [22]. The policy (actor) is parameterized as a three-layer MLP that maps the state (s_t, g_t) to a Gaussian action distribution over PD target joint positions. The value function (critic) is parameterized as a three-layer MLP, but is also provided with an additional task indicator variable k_t that specifies whether the current sample comes from the motion imitation task or the generalization task. In addition, the critic receives privileged information available in simulation but not in the real world, such as the full root state, contact forces, and the assistive wrench signal. This privileged input is used solely for value estimation to improve training stability and sample efficiency.

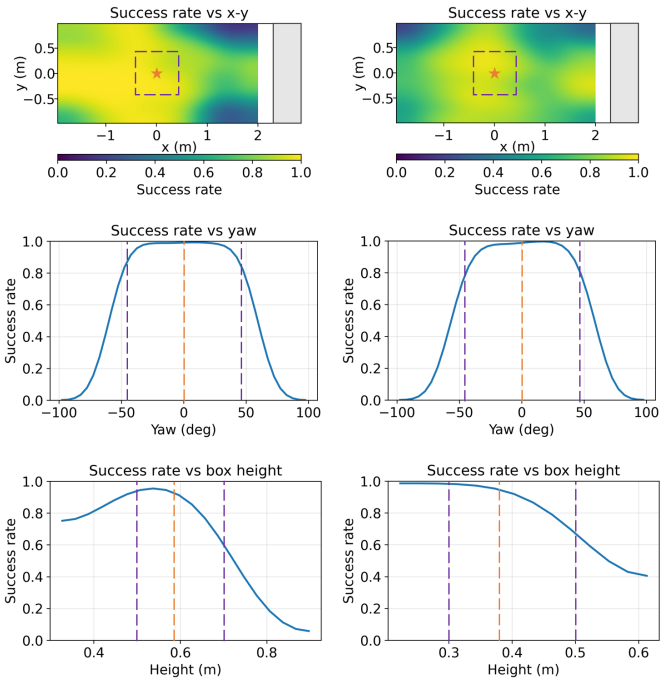
Reference data. We obtain the reference data from a single monocular video for each skill. From the captured video, we reconstruct the 3D human motion sequence and the corresponding 3D scene using CRISP [33], which provides a consistent human–scene reconstruction from monocular input. The reconstructed human motion is then retargeted to the humanoid robot using GMR [2], mapping the human kinematics to the robot. The resulting retargeted motion is used to define tracking rewards, reference state initialization, and goal conditions during training, and is not provided as input to the policy at execution time.

III. Evaluation

In this section, we conduct experiments both in simulation and on hardware to answer the following questions:

- ***Robustness and Generalization in Simulation and on Hardware:*** *How robust are the learned policies under nominal and beyond-nominal conditions, and how well do they generalize to variations in initial states and task configurations in both simulation and real-world experiments?*
- ***Comparison to Alternative Training Paradigms:*** *How does our method perform overall compared to tabula rasa (pure) RL, tracking-based RL in terms of task success, robustness, and motion quality?*
- ***Long-Horizon Skill Composition:*** *Can the learned policies be composed through task-level goals to produce long-horizon parkour behaviors?*
- ***Key Components and Ablations:*** *Which components are essential for the pipeline to work effectively, as determined through simulation-based ablation studies?*
- ***Method Generality and Extensions:*** *Can the proposed framework extend to multi-skill learning and perceptual inputs within a unified formulation?*

All experiments are performed using the Unitree G1 humanoid (29 DoF, $1.2\,\text{m}$ tall, $35\,\text{kg}$), with simulation evalu-



(a) Walk-Climb (b) Walk-Jump

Fig. 2: Success rate of our method under different initial conditions for walk-climb and walk-jump skills. When varying one initial condition, all other conditions are held at their nominal values. Orange markers show the nominal configuration of the initial state, while the purple markers show the randomness level of the initialization during training. The gray rectangle represents the box with its edge positioned at 2.3m.

ations conducted either in MuJoCo [28] or Isaac Lab [17]. We evaluate three representative behaviors: walk-jump, walk-climb, and climb-down. We train one policy per behavior using a single reference motion, with the walk–climb and walk–jump behaviors augmented by their mirrored motions.

A. How Robust and Generalizable Are the Learned Policies in Simulation and on Hardware?

We assess robustness and generalization by introducing controlled variability in initial conditions, and measure how reliably the learned behaviors extend beyond the nominal training configuration while preserving motion quality. Specifically, we report task success rates under systematic variations of the initial root position, heading orientation, and box height, using the walk-climb and walk-jump tasks as representative examples. We evaluate a thousand trials for each task and a trial is considered successful if the robot ends in a stable standing configuration on top of the box, with the base height within 10cm of the target height (0.8m), and the base position within 20cm of the box center in the horizontal plane (the box half-width from edge to center is 40cm). We focus on walk–climb and walk–jump because these tasks naturally involve a walking phase, which allows the robot to approach the box from a wide range of initial positions and thus provides a meaningful test of generalization under beyond-nominal conditions. As shown

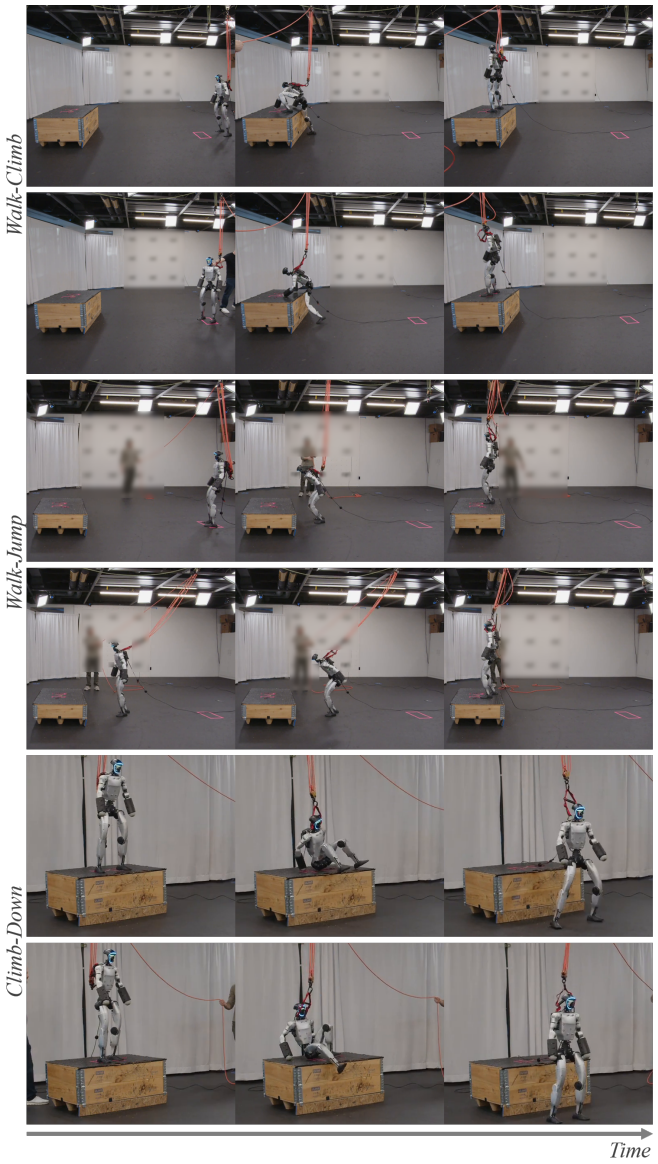


Fig. 3: Hardware experiments with varied initial conditions for the walk–climb, walk–jump, and climb–down skills. Each behavior is depicted from two different initial conditions. Despite changes in initial conditions, the robot adapts its strategy and successfully executes the skills.

in Fig. 2, our method maintains high success rates across a wide range of perturbations to initial conditions, demonstrating that the learned behaviors are not narrowly tied to specific reference executions but instead generalize well beyond the nominal starting state.

To provide a qualitative illustration of the policy's generalization capability, we conduct a sim-to-real evaluation on hardware, as shown in Fig. 3. The figure includes all three representative behaviors: walk-climb, walk-jump, and climb-down. For the walk-climb skill, the robot is initialized at different distances from the box, ranging from far to near. The robot is able to walk toward the box and successfully climb onto it. Notably, the policy naturally adapts its strategy by

TABLE I: Quantitative comparison with alternative training paradigms. Task success rate reflects robustness, while base orientation (root R) and joint position (joint pos q) errors measure similarity to the reference motion. The latter metrics are omitted for tabula rasa RL, which does not use reference information. Arrows in the table indicate the desired direction (increase or decrease) for each metric.

Methods	**Walk-Jump**			**Walk-Climb**			**Climb-Down**		
	success rate ↑	root R ↓	joint pos q ↓	success rate ↑	root R ↓	joint pos q ↓	success rate ↑	root R ↓	joint pos q ↓
Nominal Initializations									
ZEST mocap	0.98	0.45	1.60	0.92	0.83	1.60	0.90	1.02	1.75
Tabula rasa RL	1.00	-	-	1.00	-	-	0.91	-	-
Ours	1.00	0.32	2.26	1.00	0.31	1.87	1.00	0.73	3.43
Beyond Nominal Initializations									
ZEST mocap	0.17	0.65	2.15	0.57	1.44	2.87	0.90	1.04	1.81
Tabula rasa RL	0.54	-	-	0.53	-	-	0.91	-	-
Ours	0.62	0.99	3.34	0.76	1.41	4.41	0.98	0.74	3.44

leading with either the left or the right leg, depending on the initial configuration and dynamics. For the walk-jump skill, we observe similar adaptive behavior. When the robot starts farther from the box, it first walks forward, then jumps. In contrast, when the initial distance is sufficiently small, the robot directly initiates a jump without any walking motion. For the climb-down skill, the robot also exhibits diverse strategies under different initial conditions. In particular, we observe cases in which the robot repeatedly uses one foot (e.g., the left foot) to push against the box, gradually shifting its center of mass forward before stepping down. These results demonstrate that the learned policies do not simply replay a single reference trajectory, but instead generalize effectively to different task configurations.

B. How Does Our Method Compare to Tabula Rasa RL and Pure Motion Imitation?

We compare our method against two representative baselines: a tracking-based RL approach (*MoCap-based ZEST* [25]) and *tabula rasa RL*. We augment ZEST with motion capture data to get the relative position and orientation between the robot and the box, and directly track reference motions during execution. In contrast, while tabula rasa RL also takes motion capture data as input, it is trained purely with carefully handcrafted task rewards, without any reference motion or imitation signals.

We evaluate these methods using three metrics: *task success rate* (as defined in Section III-A), *root orientation error*, and *joint position error*. The root orientation error measures the deviation of the projected gravity direction and captures overall balance and stability, while the joint position error reflects motion naturalness. We consider both nominal and beyond-nominal initializations, where beyond-nominal conditions introduce randomized offsets to test robustness. For walk–jump and walk–climb, the initial conditions are perturbed by up to $\pm 2, \mathrm{m}$ forward, $\pm 1, \mathrm{m}$ laterally, and $\pm 45^\circ$ in yaw. For climb-down, where the robot starts on top of the box, smaller perturbations are applied ($-0.3, \mathrm{m}$ to $+0.1, \mathrm{m}$ forward, $\pm 0.2, \mathrm{m}$ laterally, and $\pm 30^\circ$ in yaw).

Quantitative results are summarized in Table I. Across all tasks and conditions, our method achieves the highest success rates under both nominal and out-of-distribution (beyond-nominal) initializations, while maintaining low root orientation error and reasonable joint position error. This indicates that the learned behaviors are both stable and natural, and that they generalize effectively to initial conditions that differ significantly from those seen during training.

ZEST mocap achieves low joint position error under nominal conditions, as expected, since the reference joint positions are explicitly provided as inputs. However, its success rate drops substantially under beyond-nominal initializations. We find that this failure mode is often caused by a strong bias toward tracking the reference motion. For example, when the robot starts closer to the box or ahead of the reference timing, the policy attempts to jump backward abruptly to re-align with the reference trajectory, leading to motion instability. As a result, while joint-level tracking remains accurate when execution succeeds, overall robustness is limited.

The tabula rasa RL policy, on the other hand, exhibits less structured and less natural behavior. Although it can succeed under nominal conditions, its motions are noticeably uncoordinated and aggressive. For example, in the climb-down task, the policy directly jumps off a box of approximately $50\,\mathrm{cm}$ height instead of executing a controlled descent. In the walk-jump and walk-climb tasks, the policy also tends to move aggressively toward the goal, often completing the entire task in roughly $2\,\mathrm{s}$, whereas the corresponding reference motions usually take $10\,\mathrm{s}$. These behaviors reflect the aggressive, greedy nature of the tabula rasa RL policy, which primarily exploits large task rewards without developing robust, coordinated strategies or structured motion. As a result, when evaluated under beyond-nominal initializations, the failure rate of tabula rasa RL increases substantially.

C. Can the Learned Skills Be Composed to Solve Long-Horizon Parkour Scenarios?

We examine whether the learned policies can be reused as modular skills and composed into longer sequences to solve long-horizon box-parkour scenarios. To this end, we compose skills using a rule-based state machine that issues task-level

TABLE II: Quantitative comparison with ablative baselines. Task success rate reflects robustness, while base orientation (root R) and joint position (joint pos q) errors measure similarity to the reference motion. Arrows in the table indicate the desired direction (increase or decrease) for each metric.

Methods	**Walk-Jump**			**Walk-Climb**			**Climb-Down**		
	success rate ↑	root R ↓	joint pos q ↓	success rate ↑	root R ↓	joint pos q ↓	success rate ↑	root R ↓	joint pos q ↓
Nominal Initializations									
Ours w/o task curriculum	0.00	0.38	2.71	0.00	0.51	2.70	0.96	0.58	2.83
Ours w/o imitation	0.00	4.52	11.3	0.00	2.88	8.00	0.98	1.88	16.12
Ours w/o generalization	0.99	0.29	2.25	0.97	0.31	1.99	0.91	0.56	2.56
Ours	1.00	0.32	2.26	1.00	0.31	1.87	1.00	0.73	3.43
Beyond Nominal Initializations									
Ours w/o task curriculum	0.00	1.01	3.34	0.00	1.20	4.04	0.69	1.01	3.81
Ours w/o imitation	0.00	4.75	12.97	0.00	3.57	10.09	0.97	3.75	16.30
Ours w/o generalization	0.27	2.10	4.95	0.53	1.78	5.06	0.70	0.79	3.30
Ours	0.62	0.99	3.34	0.76	1.41	4.41	0.98	0.74	3.44

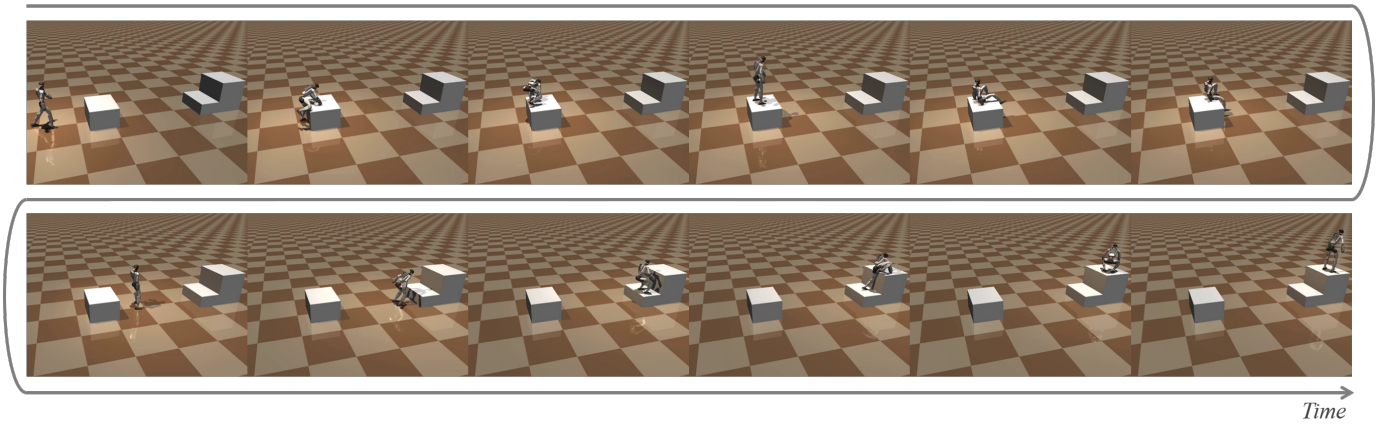


Fig. 4: Multi-skill composition in a sim-to-sim evaluation in MuJoCo. Learned policies are composed to execute walk-climb, walk-jump, and climb-down behaviors over long horizons.

goals based on the box layout. Fig. 4 shows representative long-horizon parkour executions composed from the learned skills, evaluated in MuJoCo, demonstrating sim-to-sim transfer to a different physics engine. In this example, the robot sequentially walks, climbs up, climbs down, walks again, jumps, and finally climbs up once more. Importantly, these behaviors remain reliable across different initial configurations and target goal specifications, without careful tuning or task-specific resets. These results show that the learned skills remain robust when sequenced over longer horizons and under different initial configurations and target specifications. We further evaluate long-horizon composition on the physical humanoid platform, as shown in Fig. 5. The robot is able to sequentially execute jumping, climbing down, walking, and climbing behaviors using the same composition strategy. Importantly, because the individual learned skills remain robust across different initial configurations and task conditions, they can be reliably reused and composed over long horizons under real-world dynamics and execution noise.

D. Which Components Are Necessary for the Pipeline to Work?

We perform simulation-based ablation studies to isolate the contributions of key design choices in our pipeline and identify which components are essential for achieving robust, generalizable, and natural behaviors. In particular, we study the roles of the task-mixing curriculum, the reference-guided imitation task, and the goal-conditioned generalization task by removing each component in turn. Quantitative results are summarized in Table II.

When the task mixing curriculum is removed, the policy is trained by jointly optimizing the imitation and generalization tasks from the beginning ("Ours w/o task curriculum"). The resulting policy fails completely on walk-jump and walk-climb skills. While the policy is able to learn basic locomotion behaviors such as walking, it consistently fails to acquire more dynamic, contact-rich skills such as jumping and climbing. This result indicates that directly optimizing imitation and generalization objectives is challenging for complex behaviors. Instead, using task curriculum with more imitation for behavior shaping, followed by more generalization task, provides a more effective learning process.

Removing the imitation task ("Ours w/o imitation task") significantly degrades both task success and motion quality. Because the goal-conditioned reward is simple and intentionally sparse, depending on the 2D distance to the goal and a final reach reward, the policy lacks sufficient guidance

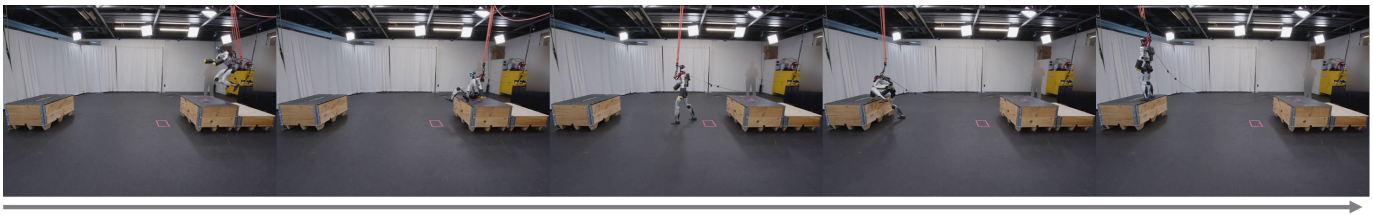


Fig. 5: Multi-skill composition on hardware. Learned skills are composed through task-level goals to execute parkour behaviors in the real world.

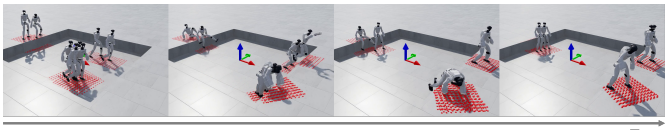


Fig. 6: Method extensions. A single policy performs multiple skills via a one-hot embedding and incorporates elevation map perceptual inputs, as shown by the red elevation-map grids.

to discover coordinated, contact-rich whole-body behaviors. As a result, the learned behavior becomes highly greedy: the robot tends to rush directly toward the goal but fails to exploit structured interactions such as stepping, jumping, and climbing to get onto the box. This variant consistently fails on walk-jump and walk-climb tasks and exhibits substantially worse motion quality, as reflected by large root orientation and joint position errors. For the climb-down task, the policy often resorts to simply jumping straight down. These results demonstrate that sparse goal rewards alone are insufficient for learning agile, coordinated motor skills.

Removing the generalization task ("Ours w/o generalization task") preserves high success rates and stylistic motion quality under nominal initializations. However, performance degrades sharply under beyond-nominal conditions, with significant drops in success rates for walk-jump and walk-climb. This highlights that imitation alone is insufficient to support robust adaptation to novel task conditions.

E. Can the Framework Support Multi-Skill Learning and Perceptual Observations?

While the primary focus of this work is on learning individual generalizable skills, the proposed framework is not restricted to this setting. Instead, it provides a flexible training paradigm that can be naturally extended to more complex scenarios with minimal modification.

To demonstrate this, we consider a unified extension in which both multi-skill learning and perceptual inputs are incorporated into a single policy. Specifically, we augment the policy with a one-hot skill embedding to enable multi-skill execution, and simultaneously incorporate perceptual inputs in the form of an elevation map to support more complex environments. These modifications require no changes to the core training objective or curriculum, and are seamlessly integrated into the existing framework.

Qualitative results are shown in Fig. 6. A single policy is able to perform multiple distinct skills conditioned on the skill embedding, while adapting its behavior based on perceptual inputs derived from the height map. This joint extension highlights that the proposed framework can support both skill diversity and environment complexity within a unified formulation. Taken together, these results suggest that the method provides a flexible foundation for learning reusable low-level skills that can extend beyond the specific settings considered in this work.

IV. Conclusion

We presented a unified multi-task RL framework for acquiring agile, natural, and deployable humanoid motor skills. The key idea is to treat reference motion as a prior for *behavioral shaping* rather than a deployment-time constraint: a single goal-conditioned policy is trained jointly on a reference-guided imitation task that provides dense supervision and a reference-free generalization task that optimizes task success under diverse initial conditions and commands. Because reference trajectories are never used as policy inputs, the resulting controller executes purely from state and goal, enabling generalization beyond the demonstration dataset. We evaluated the approach via challenging box-based parkour tasks that require a range of athletic behaviors, including jumping and climbing, and showed that multi-task training yields robust transfer while preserving motion naturalness. Finally, we demonstrated long-horizon skill composition using a simple state-machine composer that produces task-level goals to sequence and deploy the learned skills to accomplish parkour objectives.

This work opens several directions for future research. Promising extensions include integrating perception and scene understanding into the high-level composer, enriching the goal interface beyond root targets and discrete behavior labels, and scaling the skill library to more complex contact-rich behaviors. Another key next step is to replace the hand-designed state machine with a learned high-level policy that selects and parametrizes goals for long-horizon decision making. Ultimately, we believe this framework provides a practical pathway toward more flexible, goal-driven humanoid autonomy in complex environments.

Acknowledgments

We thank Zach Nobles, Francesco Iacobelli, Scott Biddlestone, Jonathan Foster, Ashley Dodge, and Sylvain Bertrand for their invaluable assistance with the hardware experiments and their support in developing the software infrastructure required for the tests.

References

[1] Arthur Allshire, Hongsuk Choi, Junyi Zhang, David McAllister, Anthony Zhang, Chung Min Kim, Trevor Darrell, Pieter Abbeel, Jitendra Malik, and Angjoo Kanazawa. Visual imitation enables contextual humanoid control. *arXiv preprint arXiv:2505.03729*, 2025.

[2] Joao Pedro Araujo, Yanjie Ze, Pei Xu, Jiajun Wu, and C Karen Liu. Retargeting matters: General motion retargeting for humanoid motion tracking. *arXiv preprint arXiv:2510.02252*, 2025.

[3] Qingwei Ben, Botian Xu, Kailin Li, Feiyu Jia, Wentao Zhang, Jingping Wang, Jingbo Wang, Dahua Lin, and Jiangmiao Pang. Gallant: Voxel grid-based humanoid locomotion and local-navigation across 3d constrained terrains. *arXiv preprint arXiv:2511.14625*, 2025.

[4] Zixuan Chen, Mazeyu Ji, Xuxin Cheng, Xuanbin Peng, Xue Bin Peng, and Xiaolong Wang. Gmt: General motion tracking for humanoid whole-body control. *arXiv preprint arXiv:2506.14770*, 2025.

[5] Wenhao Cui, Shengtao Li, Huaxing Huang, Bangyu Qin, Tianchu Zhang, Liang Zheng, Ziyang Tang, Chenxu Hu, NING Yan, Jiahao Chen, et al. Adapting humanoid locomotion over challenging terrain via two-phase training. In *8th Annual Conference on Robot Learning*, 2024.

[6] Zhiyang Dou, Xuelin Chen, Qingnan Fan, Taku Komura, and Wenping Wang. C· ase: Learning conditional adversarial skill embeddings for physics-based characters. In *SIGGRAPH Asia 2023 Conference Papers*, pages 1–11, 2023.

[7] Xinyang Gu, Yen-Jen Wang, Xiang Zhu, Chengming Shi, Yanjiang Guo, Yichen Liu, and Jianyu Chen. Advancing humanoid locomotion: Mastering challenging terrains with denoising world model learning. *arXiv preprint arXiv:2408.14472*, 2024.

[8] Jinrui Han, Weiji Xie, Jiakun Zheng, Jiyuan Shi, Weinan Zhang, Ting Xiao, and Chenjia Bai. Kungfubot2: Learning versatile motion skills for humanoid whole-body control. *arXiv preprint arXiv:2509.16638*, 2025.

[9] Junzhe He, Chong Zhang, Fabian Jenelten, Ruben Grandia, Moritz Bächer, and Marco Hutter. Attention-based map encoding for learning generalized legged locomotion. *Science Robotics*, 10(105):eadv3604, 2025.

[10] Tairan He, Zhengyi Luo, Xialin He, Wenli Xiao, Chong Zhang, Weinan Zhang, Kris Kitani, Changliu Liu, and Guanya Shi. Omnih2o: Universal and dexterous human-to-humanoid whole-body teleoperation and learning. *arXiv preprint arXiv:2406.08858*, 2024.

[11] Tairan He, Jiawei Gao, Wenli Xiao, Yuanhang Zhang, Zi Wang, Jiashun Wang, Zhengyi Luo, Guanqi He, Nikhil Sobanbab, Chaoyi Pan, et al. Asap: Aligning simulation and real-world physics for learning agile humanoid whole-body skills. *arXiv preprint arXiv:2502.01143*, 2025.

[12] Mazeyu Ji, Xuanbin Peng, Fangchen Liu, Jialong Li, Ge Yang, Xuxin Cheng, and Xiaolong Wang. Exbody2: Advanced expressive humanoid whole-body control. *arXiv preprint arXiv:2412.13196*, 2024.

[13] Qiayuan Liao, Takara E Truong, Xiaoyu Huang, Yuman Gao, Guy Tevet, Koushil Sreenath, and C Karen Liu. Beyondmimic: From motion tracking to versatile humanoid control via guided diffusion. *arXiv preprint arXiv:2508.08241*, 2025.

[14] Chenhao Lu, Xuxin Cheng, Jialong Li, Shiqi Yang, Mazeyu Ji, Chengjing Yuan, Ge Yang, Sha Yi, and Xiaolong Wang. Mobile-television: Predictive motion priors for humanoid whole-body control. *arXiv preprint arXiv:2412.07773*, 2024.

[15] Zhengyi Luo, Ye Yuan, Tingwu Wang, Chenran Li, Sirui Chen, Fernando Castañeda, Zi-Ang Cao, Jiefeng Li, David Minor, Qingwei Ben, et al. Sonic: Supersizing motion tracking for natural humanoid whole-body control. *arXiv preprint arXiv:2511.07820*, 2025.

[16] Le Ma, Ziyu Meng, Tengyu Liu, Yuhan Li, Ran Song, Wei Zhang, and Siyuan Huang. Styleloco: Generative adversarial distillation for natural humanoid robot locomotion. *arXiv preprint arXiv:2503.15082*, 2025.

[17] Mayank Mittal, Pascal Roth, James Tigue, Antoine Richard, Octi Zhang, Peter Du, Antonio Serrano-Muñoz, Xinjie Yao, René Zurbrügg, Nikita Rudin, et al. Isaac lab: A GPU-accelerated simulation framework for multi-modal robot learning. *arXiv preprint arXiv:2511.04831*, 2025.

[18] Liang Pan, Zeshi Yang, Zhiyang Dou, Wenjia Wang, Buzhen Huang, Bo Dai, Taku Komura, and Jingbo Wang. Tokenhsi: Unified synthesis of physical human-scene interactions through task tokenization. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 5379–5391, 2025.

[19] Xue Bin Peng, Pieter Abbeel, Sergey Levine, and Michiel Van de Panne. Deepmimic: Example-guided deep reinforcement learning of physics-based character skills. *ACM Transactions On Graphics (TOG)*, 37(4):1–14, 2018.

[20] Xue Bin Peng, Ze Ma, Pieter Abbeel, Sergey Levine, and Angjoo Kanazawa. Amp: Adversarial motion priors for stylized physics-based character control. *ACM Transactions on Graphics (ToG)*, 40(4):1–20, 2021.

[21] Xue Bin Peng, Yunrong Guo, Lina Halper, Sergey Levine, and Sanja Fidler. Ase: Large-scale reusable adversarial skill embeddings for physically simulated characters. *ACM Transactions On Graphics (TOG)*, 41 (4):1–17, 2022.

[22] Lerrel Pinto, Marcin Andrychowicz, Peter Welinder, Wojciech Zaremba, and Pieter Abbeel. Asymmetric actor critic for image-based robot learning. *arXiv preprint arXiv:1710.06542*, 2017.

[23] Junli Ren, Junfeng Long, Tao Huang, Huayi Wang, Zirui Wang, Feiyu Jia, Wentao Zhang, Jingbo Wang, Ping Luo, and Jiangmiao Pang. Humanoid goalkeeper: Learning from position conditioned task-motion constraints. *arXiv preprint arXiv:2510.18002*, 2025.

[24] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[25] Jean Pierre Sleiman, He Li, Alphonsus Adu-Bredu, Robin Deits, Arun Kumar, Kevin Bergamin, Mohak Bhardwaj, Scott Biddlestone, Nicola Burger, Matthew A. Estrada, Francesco Iacobelli, Twan Koolen, Alexander Lambert, Erica Lin, M. Eva Mungai, Zach Nobles, Shane Rozen-Levy, Yuyao Shi, Jiashun Wang, Jakob Welner, Fangzhou Yu, Mike Zhang, Alfred Rizzi, Jessica Hodgins, Sylvain Bertrand, Yeuhi Abe, Scott Kuindersma, and Farbod Farshidian. ZEST: Zero-shot embodied skill transfer for athletic robot control. *arXiv preprint arXiv:2602.00401*, 2026.

[26] Annan Tang, Takuma Hiraoka, Naoki Hiraoka, Fan Shi, Kento Kawaharazuka, Kunio Kojima, Kei Okada, and Masayuki Inaba. Humanmimic: Learning natural locomotion and transitions for humanoid robot via wasserstein adversarial imitation. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 13107–13114. IEEE, 2024.

[27] Chen Tessler, Yunrong Guo, Ofir Nabati, Gal Chechik, and Xue Bin Peng. Maskedmimic: Unified physics-based character control through masked motion inpainting. *ACM Transactions on Graphics (TOG)*, 43(6):1–21, 2024.

[28] Emanuel Todorov, Tom Erez, and Yuval Tassa. MuJoCo: A physics engine for model-based control. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, pages 5026–5033. IEEE, 2012.

[29] Dewei Wang, Xinmiao Wang, Xinzhe Liu, Jiyuan Shi, Yingnan Zhao, Chenjia Bai, and Xuelong Li. More: Mixture of residual experts for humanoid lifelike gaits learning on complex terrains. *arXiv preprint arXiv:2506.08840*, 2025.

[30] Huayi Wang, Zirui Wang, Junli Ren, Qingwei Ben, Tao Huang, Weinan Zhang, and Jiangmiao Pang. Beamdojo: Learning agile humanoid locomotion on sparse footholds. *arXiv preprint arXiv:2502.10363*, 2025.

[31] Jiashun Wang, Jessica Hodgins, and Jungdam Won. Strategy and skill learning for physics-based table tennis animation. In *ACM SIGGRAPH 2024 Conference Papers*, pages 1–11, 2024.

[32] Jiashun Wang, Yifeng Jiang, Haotian Zhang, Chen Tessler, Davis Rempe, Jessica Hodgins, and Xue Bin Peng. Hil: Hybrid imitation learning of diverse parkour skills from videos. *arXiv preprint arXiv:2505.12619*, 2025.

[33] Zihan Wang, Jiashun Wang, Jeff Tan, Yiwen Zhao, Jessica Hodgins, Shubham Tulsiani, and Deva Ramanan. Crisp: Contact-guided real2sim from monocular video with planar scene primitives. *arXiv preprint arXiv:2512.14696*, 2025.

[34] Kehan Wen, Chenhao Li, Junzhe He, and Marco Hutter. Constrained style learning from imperfect demonstrations under task optimality. *arXiv preprint arXiv:2507.09371*, 2025.

[35] Haoyang Weng, Yitang Li, Nikhil Sobanbabu, Zihan Wang, Zhengyi Luo, Tairan He, Deva Ramanan, and Guanya Shi. Hdmi: Learning interactive humanoid whole-body control from human videos. *arXiv preprint arXiv:2509.16757*, 2025.

[36] Weiji Xie, Jinrui Han, Jiakun Zheng, Huanyu Li, Xinzhe Liu, Jiyuan Shi, Weinan Zhang, Chenjia Bai, and Xuelong Li. Kungfubot: Physics-based humanoid whole-body control for learning highly-dynamic skills. *arXiv preprint arXiv:2506.12851*, 2025.

[37] Michael Xu, Yi Shi, KangKang Yin, and Xue Bin Peng. Parc: Physics-based augmentation with reinforcement learning for character controllers. In *Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers*, pages 1–11, 2025.

[38] Shaofeng Yin, Yanjie Ze, Hong-Xing Yu, C Karen Liu, and Jiajun Wu. Visualmimic: Visual humanoid loco-manipulation via motion tracking and generation. *arXiv preprint arXiv:2509.20322*, 2025.

[39] Ziwen Zhuang, Shenzhe Yao, and Hang Zhao. Humanoid parkour learning. *arXiv preprint arXiv:2406.10759*, 2024.

APPENDIX

This appendix provides detailed implementation descriptions of the proposed framework. We describe the observation design, goal representation, reward formulation, curriculum schedules, network architecture, RL training, state initialization, and domain randomization used in all experiments, with an emphasis on faithful reproducibility.

In addition, we provide a supplementary video that presents extensive qualitative results. The video includes hardware experiments demonstrating successful execution and generalization of learned behaviors, as well as side-by-side comparisons with baseline methods in simulation under varied initial conditions and environment configurations.

A. Observations and Goal Representation

We adopt an asymmetric actor–critic formulation. The policy receives only deployment-available proprioceptive observations and a low-dimensional goal command, while the critic is additionally provided with privileged information available in simulation to improve value estimation. Zero-mean Gaussian noise is injected into selected policy observations to improve robustness. Unless otherwise noted, we do not apply per-term scaling or clipping. The observation is summarized in Table III

a) Goal representation.: At each timestep t, the policy receives the proprioceptive state s_t and a goal variable g_t. In all box-based tasks, we instantiate the goal as a target *root pose* expressed in the character-centric frame:

$$g_t \;=\; \big(\Delta x_{\text{goal}},\, \Delta y_{\text{goal}},\, q_{\text{goal}}\big), \tag{6}$$

where $(\Delta x_{\text{goal}}, \Delta y_{\text{goal}})$ denotes the desired horizontal displacement of the root. The orientation component q_{goal} specifies a desired yaw orientation represented as a *relative* quaternion; importantly, this yaw is defined in the world frame and aligned with the box orientation, rather than with the reference motion. The policy does not observe reference trajectories, future reference windows, or explicit phase variables; all task intent is conveyed solely through this goal specification.

b) Goal definition relative to the box.: In box-based parkour tasks, the goal is defined relative to the box through a canonical, task-dependent location. For the walk–jump and walk–climb tasks, the goal corresponds to the *center of the top surface of the box*, expressed in the character-centric frame. For the climb–down task, the goal corresponds to a target root location on the ground positioned at a fixed distance in front of the box (0.9 m along the forward direction from the box center). The policy is not explicitly provided with box geometry such as height or dimensions; instead, the box influences behavior implicitly through physical interaction and feasibility constraints.

B. Reward Function

a) Tracking Rewards: During imitation, we employ dense tracking rewards that encourage the humanoid to remain consistent with reference motion statistics without explicitly conditioning the policy on the reference. Tracking rewards are implemented using exponentially weighted penalties of the form

$$r_{\text{track}} = \exp\!\left(-\kappa \frac{\|e\|^2}{\sigma_i^2}\right), \tag{7}$$

where e denotes a task-specific tracking error, σ_i is a normalization scale, and κ controls sensitivity.

We include tracking terms for base position and orientation, base linear and angular velocity, joint positions, and key body positions and orientations. In addition, we explicitly encourage alignment between the robot's projected gravity direction and the reference, which improves balance and orientation stability during dynamic motions.

b) Task Rewards: For goal-conditioned generalization tasks, reference-dependent terms are removed and replaced with task-level objectives. We include penalties on target base position and orientation errors relative to the goal, as well as a constant success reward that is activated when the robot reaches the target region. These terms provide minimal task supervision, forcing the policy to reuse behaviors acquired during imitation.

c) Regularization: To promote smooth and physically feasible behavior, we apply penalties on action smoothness, joint accelerations, applied torques, and violations of joint position and torque limits. We further penalize large horizontal contact forces at the feet, foot slippage, excessive foot jerk, and undesirable ankle configurations. These terms are critical for stabilizing training in contact-rich scenarios such as jumping and climbing.

d) Survival Reward: To discourage early termination and promote sustained execution, we include a constant survival reward that is applied at every timestep.

All reward terms are summarized in Table IV, along with their weights and normalization constants.

C. Curriculum

We employ a coupled curriculum that combines an assistive-wrench mechanism with a task-mixing schedule to stabilize early-stage learning and enable gradual transfer from imitation to generalization. All curriculum-related parameters are summarized in Table V.

a) Assistive Wrench Curriculum: A virtual assistive wrench is applied at the robot base during early training to reduce catastrophic failures. Let (p, v, Φ, ω) denote the base position, linear velocity, orientation, and angular velocity, and $(\hat{p}, \hat{v}, \hat{\Phi}, \hat{\omega})$ their reference counterparts. We compute a nominal spatial wrench using PD feedback on base tracking errors combined with feedforward terms. The applied wrench is scaled by a difficulty-dependent factor $\beta(\lambda)$, which decreases monotonically as training progresses.

b) Task Mixing Schedule: The same scalar difficulty variable $\lambda \in [0, 1]$ also controls the probability of sampling the imitation task versus the generalization task. Training begins with imitation-dominated sampling and gradually shifts toward more frequent generalization-task sampling as the policy becomes more stable.

TABLE III: Observation terms summary. Noise is zero-mean Gaussian and additive. Privileged observations are used by the critic only.

Term Name	Definition	Noise
Policy Observations		
Torso angular velocity	${}^{T}\omega_{IT}$ (IMU on torso)	$\mathcal{N}(0,\ 0.10^2)$
Projected gravity	${}^{T}g_I$ (gravity direction expressed in torso/IMU frame)	$\mathcal{N}(0,\ 0.015^2)$
Joint positions (relative)	q_{rel}	$\mathcal{N}(0,\ 0.005^2)$
Joint velocities (relative)	$\dot{q}_{\text{rel}}$	$\mathcal{N}(0,\ 0.25^2)$
Previous action	a_{t-1}	–
Target command (relative pose)	$(\Delta x_{\text{goal}},\ \Delta y_{\text{goal}},\ q_{\text{goal}})$	$\mathcal{N}(0,\ 0.015^2)$
Privileged Observations (critic only)		
Projected gravity (base frame)	${}^{B}g_I$	–
Base linear velocity (base frame)	${}^{B}v_{IB}$	–
Base angular velocity (base frame)	${}^{B}\omega_{IB}$	–
Base height	${}^{I}r^{z}_{IB}$	–
End-effector incoming wrenches	$\{\ {}^{B}w_{\text{ee}}\ \}$ (contact wrenches on selected bodies)	–
End-effector positions w.r.t. base	$\{\ {}^{B}r_{B\text{ee}}\ \}$	–
End-effector linear velocities (base frame)	$\{\ {}^{B}v_{I\text{ee}}\ \}$	–
Assistive wrench (fictitious force)	f_{assist}	–
Assistive wrench (fictitious torque)	τ_{assist}	–
Wrench scale	β	–
Similarity metric	$\hat{s}$ (reference-based similarity / tracking score)	–
Task binary indicator	$k_t \in \{0, 1\}$ (imitation vs. generalization)	–
Reference look-ahead joint delta	$q^*_{t+1} - q_t$ (relative reference, look-ahead $= 1$)	–

c) Tracking Similarity and Curriculum Update: The curriculum scalar λ is updated online based on a tracking similarity score $s \in [0, 1]$, which measures how well the current behavior matches the reference motion.

The similarity score is computed by aggregating multiple normalized tracking errors using bounded exponential kernels:

$$s = \frac{1}{Z} \sum_i c_i \exp\left(-\kappa \frac{\|e_i\|^2}{\sigma_i^2}\right), \tag{8}$$

where e_i denotes tracking errors for different terms (e.g., joint positions, base pose, and velocities), σ_i are normalization factors, c_i are weighting coefficients, and Z is a normalization constant.

The curriculum scalar is then defined as:

$$\lambda = \text{clip}(s/s_{\max}, 0, 1), \tag{9}$$

where $s_{\max}$ corresponds to a threshold for high-quality tracking. In our experiments, we use $s_{\max} = 0.8$.

The curriculum update is performed periodically during training (every 50 iterations), ensuring that both the assistive wrench scaling and the task sampling distribution are adjusted in a synchronized manner as performance improves.

D. Network Architecture and Training Details

This section summarizes the Markov Decision Process (MDP) configuration, network architecture, and optimization hyperparameters used in all experiments. Unless otherwise noted, the same settings are shared across tasks and behaviors. We report all relevant hyperparameters explicitly to facilitate reproducibility in Table VI.

E. State Initialization and Domain Randomization

We adopt Reference State Initialization (RSI) [19]. To encourage robustness and prevent overfitting to the exact reference trajectory, we apply randomized perturbations to the initialized base pose. These perturbations are applied across the entire reference trajectory.

For the walk–climb and walk–jump tasks, we apply larger perturbations to distances and orientations relative to the box. For the climb–down task, smaller perturbations are used to maintain feasibility near the top of the box. All perturbation ranges are summarized in Table VII.

To improve robustness and facilitate transfer, we apply domain randomization during training, including perturbations to dynamics, contacts, and observations. All domain randomization ranges are summarized in Table VIII.

TABLE IV: Reward terms and hyperparameters used in training. For exponential tracking terms, we use the form $\exp(-\kappa\|e\|^2/\sigma_i^2)$, where κ is a global sensitivity parameter shared across all tracking terms and σ_i is a per-term normalization scale. We use r_{IB} and Φ_{IB} to denote the base position and orientation in the world frame, v_{IB} and ω_{IB} for base linear and angular velocities, and q for joint positions; starred quantities $(\cdot)^*$ denote reference values. The operator $\ominus$ denotes the $\mathrm{SO}(3)$ difference implemented via the rotation-log map. For the joint position term, n_j denotes the number of actuated joints. For contact-related terms, let $F^w_{t,h,b} \in \mathbb{R}^3$ denote the net contact force in the world frame acting on body b at history index h, and let $\mathcal{B}$ denote the set of foot bodies. Let $v^w_{\text{foot}}(b) \in \mathbb{R}^2$ denote the planar velocity of foot b. In the inequality ankle position limit penalty, A and b define a convex polytope constraint on ankle joint configurations. In the flat ankle penalty, $\tilde{g}_z^{(i)}$ denotes the z-component of the gravity vector expressed in the aligned local frame of ankle i. For the foot clearance reward, h_b denotes the vertical height of foot b, h^* is the target clearance height, and α is a scalar controlling velocity gating inside the $\tanh(\cdot)$ term. Finally, Δt denotes the simulation timestep used to compute foot jerk.

Term Name	**Definition (per-env scalar)**	**Weight**	$\boldsymbol{\sigma_i}$
	Tracking Rewards		
Base position tracking	$\exp\left(-\kappa \left\|r_{IB} - r^*_{IB}\right\|^2 / \sigma_1^2\right)$	1	0.4
Base orientation	$\exp\left(-\kappa \left\|\Phi_{IB} \ominus \Phi^*_{IB}\right\|^2 / \sigma_2^2\right)$	1	0.5
Base angular velocity	$\exp\left(-\kappa \left\|\omega_{IB} - \omega^*_{IB}\right\|^2 / \sigma_3^2\right)$	1	1.5
Base linear velocity	$\exp\left(-\kappa \left\|v_{IB} - v^*_{IB}\right\|^2 / \sigma_4^2\right)$	1	0.6
Joint position	$\exp\left(-\kappa \left\|q - q^*\right\|^2 / \sigma_5^2\right)$	1	$0.3 \cdot \sqrt{n_j}$
Base height tracking penalty	$\lvert z_{\text{base}} - z_{\text{ref}}\rvert$	-10.0	–
	Goal-Conditioned Task Rewards		
Target base position penalty	$\left\|p^{xy}_{\text{base}} - p^{xy}_{\text{goal}}\right\|$	-5.0	–
Target base orientation penalty	$\left\|\Phi_{\text{base}} \ominus \Phi_{\text{goal}}\right\|$	-1.0	–
Target reach reward	$\mathbb{1}[\text{reach}]$	10.0	–
	Regularization and Contact Terms		
Large horizontal foot force (indicator)	$\mathbb{1}\left[\,\overline{f}^{xy}_{\max} > 10\,\right],\ \overline{f}^{xy}_{\max} = \frac{1}{\lvert\mathcal{B}\rvert}\sum_{b\in\mathcal{B}} \max_h \left\|(F^w_{t,h,b})_{xy}\right\|_2$	-10.0	–
Action smoothness penalty	$\left\|a_t - a_{t-1}\right\|_2$	-1.0	–
Applied torque penalty	$\left\|\tau_{\text{applied}}\right\|_2$	-5×10^{-4}	–
Joint position limit penalty	$\sum_i \left[\max(0, q_i^{\min} - q_i) + \max(0, q_i - q_i^{\max})\right]$	-5.0	–
Applied torque limit penalty	$\sum_i \lvert\tau_{\text{applied},i} - \tau_{\text{computed},i}\rvert$	-0.1	–
Inequality ankle position limit penalty	$\text{clamp}\left(\sum_k \max\left(0,\ (q_{\text{ankle}}A^\top)_k - b_k\right),\ 10\right)$	-2.0	–
Foot slip penalty	$\text{clamp}\left(\sum_{b\in\mathcal{B}} \left\|v^w_{\text{foot}}(b)\right\|_2\ \mathbb{1}\left[\max_h \left\|F^w_{t,h,b}\right\|_2 > 1\right],\ 10\right)$	-2.0	–
Foot jerk penalty	$\text{clamp}\left(\left\|\frac{a^w_{\text{foot}}(t) - a^w_{\text{foot}}(t-1)}{\Delta t}\right\|_F,\ 10\right)$	-5×10^{-4}	–
Flat ankle penalty	$\left(\tilde{g}_z^{(1)} + 1\right)^2 + \left(\tilde{g}_z^{(2)} + 1\right)^2$	-20.0	–
Foot clearance reward	$\exp\left(-\frac{1}{\sigma_6}\sum_{b\in\mathcal{B}}\left(h_b - h^*\right)^2\ \tanh(\alpha\|v^w_{\text{foot}}(b)\|_2)\right)$	2.0	0.05
	Survival Reward		
Survival bias	1	30.0	–

TABLE V: Curriculum hyperparameters.

Component	**Specification**
Assistive wrench scaling	$\beta(\lambda) = (1-\lambda)\,\beta_{\max}$
Maximum assistive scale $\beta_{\max}$	0.75
Virtual force PD gains (k_p^v, k_d^v)	$(0, 15)$
Virtual torque PD gains (k_p^ω, k_d^ω)	$(200, 1)$
Imitation sampling probability	$p_{\text{imi}}(\lambda) = (1-\lambda)p_0 + \lambda p_{\text{target}}$
Initial imitation probability p_0	1.0
Final imitation probability p_{target}	0.5

TABLE VI: MDP configuration, network architecture, and PPO hyperparameters.

Hyperparameter	**Value**
MDP and Simulation Setup	
Episode length ($\bar{L}_{\text{episode}}$)	10.0 s
Simulation time-step (dt)	0.004 s
Control decimation	5
Control frequency	50 Hz
Policy and Value Networks	
Actor network	MLP(1024, 512, 256) with ELU activations
Critic network	MLP(1024, 512, 256) with ELU activations
Actor observations	Proprioception + goal
Critic observations	Actor obs + privileged information
PPO Hyperparameters	
Learning rate (start of training)	1×10^{-4}
Discount factor (γ)	0.99
GAE discount factor (λ_{GAE})	0.95
Desired KL-divergence	0.01
PPO clip range	0.2
Entropy coefficient	0.001
Value function loss coefficient	0.5
Number of epochs per update	5
Number of environments	4096
Batch size	245,760 (4096×24)
Mini-batch size	61,440 (4096×6)

TABLE VII: Reference-based initialization perturbations applied to the base pose. All offsets are sampled uniformly within the specified ranges and applied relative to the reference state.

Task	Δx (m)	Δy (m)	Yaw (rad)	Roll / Pitch (rad)
Walk–jump	± 0.4	± 0.4	± 0.8	± 0.15
Walk–climb	± 0.4	± 0.4	± 0.8	± 0.15
Climb–down	± 0.2	± 0.2	± 0.6	± 0.15

TABLE VIII: Domain randomization terms used during training.

Term	**Value**
Static friction	$\mathcal{U}(0.8,\ 2.5)$
Dynamic friction	$\mathcal{U}(0.7,\ 2.5)$
Restitution	$\mathcal{U}(0.0,\ 0.2)$
Torso mass	default $+\ \mathcal{U}(-2.5,\ 4.0)$ kg
Pelvis mass	default $+\ \mathcal{U}(-1.0,\ 1.0)$ kg
External disturbance (impulsive push at base)	Interval $= \mathcal{U}(0.0\,\text{s},\ 4.0\,\text{s})$, $v_{xy} = 0.4\ \text{m}\,\text{s}^{-1}$