Robotics: Science and Systems 2026
Sydney, Australia, July 13–17, 2026

Perceptive Humanoid Parkour: Chaining Dynamic Human Skills via Motion Matching

Zhen Wu*[1], Xiaoyu Huang*[1,2], Lujie Yang*[1], Yuanhang Zhang[1,3], Xi Chen[1],
Pieter Abbeel†[1,2], Rocky Duan†[1], Angjoo Kanazawa†[1,2], Carmelo Sferrazza†[1], Guanya Shi†[1,3], C. Karen Liu†[1,4]
[1]Amazon FAR, [2]UC Berkeley, [3]CMU, [4]Stanford University, †Amazon FAR team co-lead
Page: https://php-parkour.github.io/

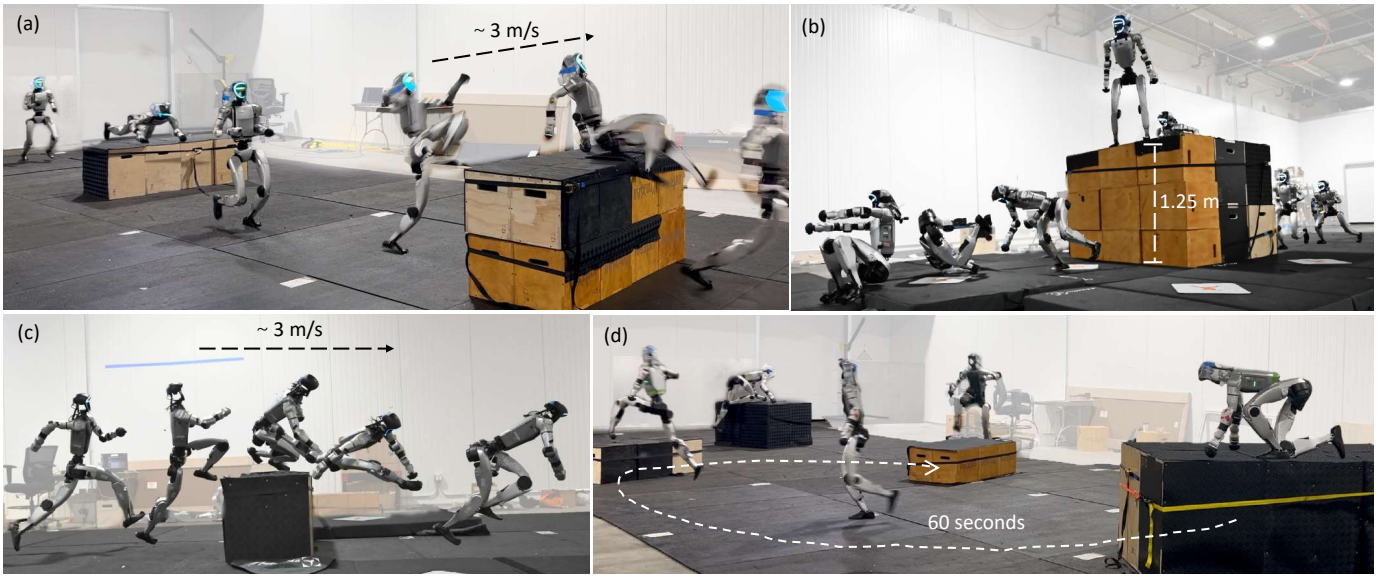

Fig. 1: **Perceptive Humanoid Parkour (PHP)** enables a Unitree G1 humanoid robot to execute highly dynamic, long-horizon parkour behaviors using onboard perception. By composing various agile human skills via motion matching and a teacher-student training pipeline, we train a single multi-skill visuomotor policy capable of complex contact-rich maneuvers including (a) cat-vaulting over a short obstacle followed by dash-vaulting over a higher obstacle at approximately 3 m/s, (b) climbing onto a 1.25 m (96% of robot height) wall, and rolling down, (c) speed-vaulting over an obstacle at approximately 3 m/s, and (d) a 60-second continuous traversal of a complex parkour course with autonomous skill selection and seamless transitions.

Abstract*—While recent advances in humanoid locomotion have achieved stable walking on varied terrains, capturing the agility and adaptivity of highly dynamic human motions remains an open challenge. In particular, agile parkour in complex environments demands not only low-level robustness, but also human-like motion expressiveness, long-horizon skill composition, and perception-driven decision-making. In this paper, we present Perceptive Humanoid Parkour (PHP), a modular framework that enables humanoid robots to autonomously perform long-horizon, vision-based parkour across challenging obstacle courses. Our approach first leverages motion matching, formulated as nearest-neighbor search in a feature space, to compose retargeted atomic human skills into long-horizon kinematic trajectories. This framework enables the flexible composition and smooth transition of complex skill chains while preserving the elegance and fluidity of dynamic human motions. Next, we train motion-tracking reinforcement learning (RL) expert policies for these composed motions, and distill them into a single depth-based, multi-skill student policy, using a combination of DAgger and RL. Crucially, the combination of perception and skill composition enables autonomous, context-aware decision-making: using only onboard depth sensing and a discrete 2D velocity command, the robot selects and executes whether to step over, climb onto, vault or roll off obstacles of varying geometries and heights. We validate our framework with extensive real-world experiments on a Unitree G1 humanoid robot, demonstrating highly dynamic parkour skills such as climbing tall obstacles up to 1.25 m (96% robot height), as well as long-horizon multi-obstacle traversal with closed-loop adaptation to real-time obstacle perturbations.

* Equal contribution.

I. Introduction

Achieving the agility and adaptivity of human motion in traversing complex terrains remains a central challenge for humanoid robotics. Humans traverse challenging terrains of drastically different dimensions by rapidly selecting and chaining dynamic whole-body skills based on perceived environmental context. Our goal is to endow humanoids with the same capability. In this work, we study parkour as a concrete, self-contained testbed for this broader objective.

Parkour highlights several core challenges. First, the robot must perform highly dynamic and contact-rich skills, such as

climbing walls around or above its body height or vaulting over obstacles within fractions of a second. This requires effective control in the humanoid's vast, high-dimensional action space. Second, these skills must be tightly coupled with exteroception, such as vision, to enable adaptation to environmental variation and rapid reaction to unexpected perturbations. Furthermore, to generalize beyond isolated maneuvers and traverse complex obstacle courses, the robot must consolidate many highly dynamic skills into a single visuomotor policy, which becomes increasingly difficult as the number and diversity of required skills grow.

Human motion data has become essential for learning highly dynamic humanoid behaviors. Prior work [20, 43] has used human motion data to successfully demonstrate highly dynamic skills such as jumping, rolling, and flipping. However, highly dynamic motion data is inherently scarce: capturing fast, contact-rich maneuvers typically requires specialized setups and careful curation, so datasets often include only one or two demonstrations per skill, each lasting just a few seconds. This scarcity is not unique to parkour but applies broadly to all dynamic human skills. Yet long-horizon tasks such as parkour require both rich within-skill variation that adapts to how the robot approaches an obstacle, and smooth, natural transitions between multiple skills across complex courses.

To address this challenge, we adopt motion matching [5, 9] as a simple yet powerful mechanism. Motion matching synthesizes long-horizon motion by retrieving and stitching motion fragments via nearest-neighbor search in a designed feature space. Crucially, this process densifies a sparse motion library by producing diverse transitions across approach distances, headings, and timing, while preserving the realism of captured motions. In our framework, motion matching enables the generation of a large set of obstacle-adaptive, long-horizon kinematic reference trajectories for downstream policy learning.

Learning a visuomotor policy that executes dozens of highly dynamic skills requires perceptive inputs that can be efficiently simulated and reliably transferred to the real world. To improve training efficiency, prior work typically trains privileged state-based experts in simulation and distills them into vision-based students using DAgger [32]. However, for humanoid parkour, pure imitation loss is limited: compounding errors can quickly derail highly dynamic skills such as climbing and vaulting. To address this, we augment distillation with an RL objective that provides task-level corrective feedback, steering the student towards successful traversal and yielding a scalable recipe across many skills.

To this end, we present Perceptive Humanoid Parkour (PHP), a modular framework that integrates human motion priors, long-horizon skill composition, and perceptive control. We first retarget human motion data into a library of robot-compatible atomic skills using OmniRetarget [43]. We then employ motion matching to compose these skills into a diverse set of long-horizon kinematic trajectories. These composed trajectories preserve agility and smooth transitions, while providing sufficient variation to learn adaptive long-horizon behaviors. We then train motion-tracking expert policies and distill them into a single depth-conditioned, multi-skill policy that enables the robot to autonomously select and transition among behaviors, such as stepping, climbing, and vaulting, using onboard depth sensing.

Our contributions are threefold:

1) An efficient kinematic skill composition pipeline that chains retargeted human motions into diverse long-horizon trajectories via motion matching.
2) A scalable training framework that distills multiple experts into a single visuomotor policy, enabling seamless transitions across diverse parkour skills.
3) Successful zero-shot sim-to-real transfer of depth-based policies on a physical humanoid robot, achieving highly dynamic parkour over various obstacles.

II. Related Works

The goal of parkour is to traverse challenging terrains agilely by perceiving, reacting, and chaining skills for different obstacles. We review related work in these areas.

A. Perceptive Terrain Traversal for Legged Robots

While blind locomotion has achieved strong robustness on moderately structured terrains such as slopes and stairs on quadrupedal robots [19, 28, 21, 18], perception enables traversal of substantially more challenging terrains [26]. In particular, perception is critical for handling sparse footholds [1, 44, 47] and discontinuous terrain such as gaps and tall obstacles [1, 46]. Building on these capabilities, prior work has enabled quadrupeds to traverse parkour-style terrain courses with consecutive gap jumps and obstacle climbs [8, 23, 13, 33].

However, translating the success on quadrupeds to humanoids remains challenging. While quadrupedal parkour skills can often be trained from scratch via reward shaping, this approach scales poorly to humanoids due to high-dimensional whole-body control. As a result, prior perceptive humanoid locomotion has primarily focused on lower-dynamic terrain traversal, including stair climbing [22, 45], walking on sparse terrain [37, 12, 2], and stepping onto low platforms [53]. Moreover, to reduce exploration difficulty in RL when training from scratch, most works adopt a teacher-student pipeline where an expert is trained with privileged states and a vision-based student is distilled via DAgger [8, 33]. We follow this paradigm but find pure DAgger insufficient for highly dynamic humanoid skills, and therefore augment it with RL to improve distillation performance. Note that this differs from the fine-tuning stage in [33], which primarily focuses on adapting an already performant DAgger-distilled policy to unseen terrains.

B. Humanoid Skill Chaining with Human Motion Data

Using human motion references effectively reduces reward engineering and produces agile, natural humanoid behaviors [20, 43, 49, 29, 40, 7], but comes at the cost of more challenging skill chaining. With reward shaping, quadrupeds can learn transitions either implicitly by a single policy [8, 23, 12, 2], or through specialist switching or distillation using a shared locomotion state [13, 52, 6, 33]. In contrast, human motion data spans heterogeneous styles that can lie in disjoint

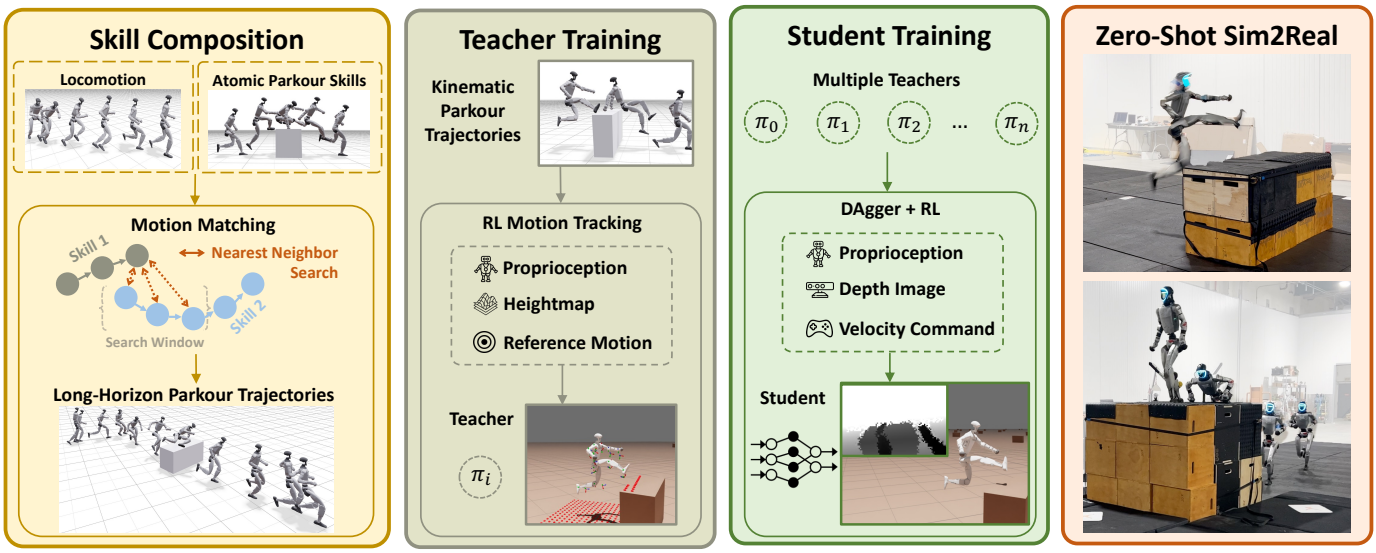

Fig. 2: **Perceptive Humanoid Parkour overview.** Atomic parkour skills are composed into long-horizon kinematic reference trajectories via motion matching. Single-skill teacher policies are trained with privileged information using RL-based motion tracking. Multiple teachers are distilled into a single depth-based student policy using a hybrid DAgger and RL objective. This scalable recipe enables zero-shot sim-to-real transfer onto a physical humanoid robot that adaptively traverses through complex terrains by autonomously executing highly agile parkour skills using onboard perception.

regions of the state space, making long-horizon composition a fundamental challenge.

AMP [31] addresses this challenge by training a single policy to learn a distribution of skills, allowing transitions to implicitly emerge from RL exploration, but replaces hand-crafted rewards with a learned style reward from motion data. While promising in animation and quadrupeds [42, 39], humanoid hardware demonstrations have so far been limited to less agile skills such as walking, stepping, and box lifting [51, 35, 38].

To address the transition problem more explicitly, another line of work generates intermediate kinematic trajectories using learned kinematics models (e.g., MDM [36]) and executes them with tracking controllers (e.g., DeepMimic [30]). These kinematics models can provide smooth transition references at test time [41, 24, 10, 48] or training time [16], but their trajectory quality degrades significantly in the low-data regimes common in parkour. This often requires either costly iterative co-training [41] to recover usable motion or receding-horizon replanning [15], which is costly with perception in real time.

In contrast, we adopt motion matching [5, 14] as a simple yet highly effective source of kinematic references for humanoid skill chaining. Motion matching has been widely adopted in video games and character animation for its simplicity and practical controllability, while still producing high-quality motion [5, 11]. While a mature technique in animation [3], it has so far been applied in robotics only to relatively simple quadruped behaviors [17]. In this work, we show that it is a powerful tool for chaining dynamic and expressive human skills over difficult terrain courses for humanoid robots, substantially improving both success rate and transition smoothness.

III. Adaptive and Agile Long-Horizon Parkour

A. Overview

The objective of this work is to enable a humanoid robot to execute agile parkour behaviors over multiple obstacles autonomously using onboard perception. We first generate long-horizon kinematic reference motions via motion matching by composing locomotion with atomic parkour skills. We then train motion-tracking expert policies with privileged observations in simulation, and finally distill them into a depth-based student policy using DAgger in combination with a PPO objective, enabling zero-shot sim-to-real deployment. An overview of the system is shown in Fig. 2.

B. Skill Composition via Motion Matching

Motion matching [5, 9] is a technique originally developed in the video game industry for interactive character control, where motion is generated online by selecting, at each frame or transition point, the animation frame from a large database whose motion features best match the current pose and desired future behavior. In this work, we adopt motion matching as an offline motion synthesis module for composing scarce atomic parkour skills with locomotion into long-horizon references.

1) Basic motion matching: We briefly summarize the standard motion matching formulation; implementation details are provided in Appx. A. Let a motion database consist of N frames, where each frame i is associated with a kinematic pose $\boldsymbol{q}_i$ and a matching feature vector $\boldsymbol{x}_i$ derived from $\boldsymbol{q}_i$. Following [14], $\boldsymbol{x}_i$ concatenates (i) short-horizon future trajectory positions and facing directions, (ii) local foot joint positions and velocities, and (iii) root velocity, all expressed in the character's local coordinate frame.

When generating transitions, given the current character state and a desired 2D velocity command, we first convert the command into a desired future trajectory and facing directions, and then concatenate these with foot positions, velocities and root velocities from the current state to form the query feature $\hat{\boldsymbol{x}}_t$. The best matching frame is then retrieved via

$$i_t^\star = \arg\min_{i \in \mathcal{C}_t} \|\hat{\boldsymbol{x}}_t - \boldsymbol{x}_i\|^2, \tag{1}$$

where $\mathcal{C}_t$ denotes the search window of the user-specified upcoming skill. This nearest-neighbor search is performed periodically (every M frames) or when the commanded velocity changes significantly. After selecting $i_t^\star$, the system transitions playback to frame $i_t^\star$ and plays forward from that frame until the next search is performed. A short blending window is applied around transitions to avoid discontinuities.

2) Long-Horizon Parkour Trajectory Synthesis: Since locomotion (walking and running) serves as a ubiquitous and naturally reusable connector between more challenging parkour skills, we generate long-horizon parkour trajectories by composing locomotion segments with short parkour skill clips in the form of `Locomotion` → `Parkour Skill` → `Locomotion`. By routing all skills through a shared locomotion manifold, this formulation enables consistent transitions across heterogeneous skills without requiring specific, hand-captured transitions between every possible skill pair, and supports scalable composition of long-horizon behaviors.

We maintain (i) a locomotion database $\mathcal{D}_{\text{loco}} = \{(\boldsymbol{x}_i^{\text{loco}}, \boldsymbol{q}_i^{\text{loco}})\}$ and (ii) a set of skill databases $\{\mathcal{D}_k\}$, one for each parkour skill k. Each skill clip is paired with a corresponding terrain asset. For every atomic skill clip in $\mathcal{D}_k$, we manually annotate the skill start and end frame indices (s_k, e_k). We additionally define a pre-skill entry window of skill-dependent length H_k:

$$\mathcal{E}_k := [\, s_k - H_k,\ s_k \,], \tag{2}$$

which corresponds to the approach phase right before the main contact-rich maneuver, where transitioning into the clip is meaningful. For example, for a vault clip, $\mathcal{E}_k$ captures the final approach steps before takeoff, avoiding transitions outside the intended approach phase.

Locomotion mode. During locomotion, we run standard motion matching via Eq.(1) with $\mathcal{C}_t = \mathcal{D}_{\text{loco}}$, and advance playback sequentially as described in Sec. III-B1.

Locomotion → Skill transition. When a transition into skill k is required, we restrict the search window $\mathcal{C}_t$ to the pre-skill entry window $\mathcal{E}_k$, and transition to the matched entry frame through Eq.(1). After the transition, the skill clip is replayed sequentially until the annotated end frame e_k. At the switch, we place the paired terrain by applying the terrain-to-root offset at the matched entry frame in the reference clip to the robot's current root pose. During skill execution, we disable further motion matching and simply advance the playback index to preserve the contact-rich human motion.

Skill → Locomotion transition. After reaching e_k, we return to locomotion by resuming motion matching via Eq.(1) with $\mathcal{C}_t = \mathcal{D}_{\text{loco}}$, and continue sequential playback.

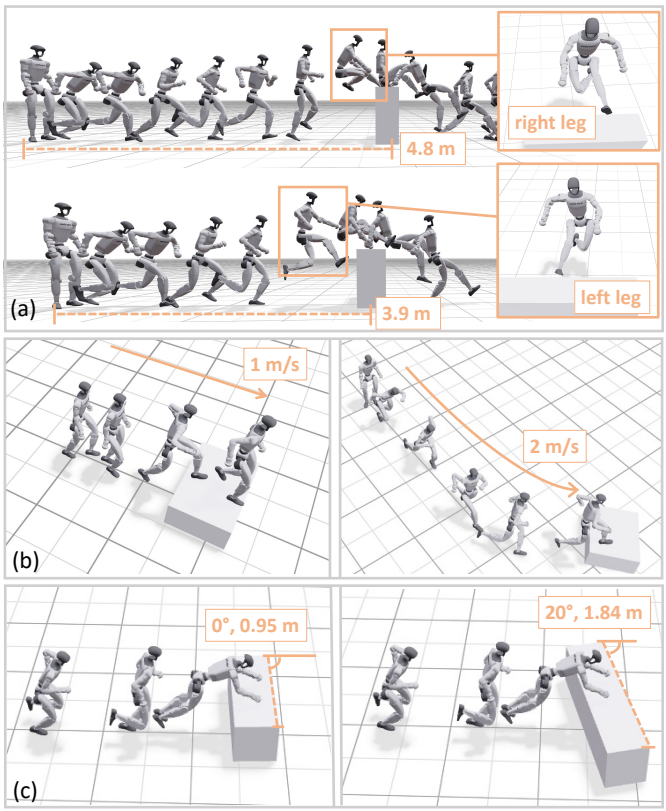

Fig. 3: **Diverse variations of composed parkour skills synthesized via motion matching.** (a) Different approach distances trigger varying stride phases and entry poses. (b) Diverse locomotion speeds, directions, and durations. (c) Randomized terrain poses and shapes.

Dataset construction. We synthesize long-horizon reference trajectories by rolling out the motion-matching composition procedure as follows. As visualized in Fig. 3(b), each trajectory starts from a standing state and enters a locomotion phase driven by 2D velocity commands sampled from two speed levels (`low` (1m/s), `high` (2m/s)) and five turning directions ($-90°$, $-45°$, $0°$, $45°$, $90°$). We then transition into skill k and replay the skill clip sequentially; during the skill, we set the command to go straight ($0°$) while keeping the same speed level as the preceding locomotion segment. After reaching the annotated end frame e_k, we return to locomotion and continue for an additional 2 seconds before stopping. Throughout synthesis, we record the per-timestep velocity commands alongside the generated kinematic reference poses $\{\boldsymbol{q}_t\}$, and use these paired trajectories for subsequent policy training.

Transition Density. Motion matching naturally induces a high density of transitions by allowing a skill to be entered from multiple locomotion states that are nearby in the motion feature space. We exploit this to generate diverse skill entrances spanning different approach distances and stride phases (e.g., adding a preparatory step before a jump, or initiating a vault from different phases of a running gait), densifying the distribution of pre-skill states. As illustrated in Fig. 3 (a), varying the initial approach distance (e.g., 3.9 m vs.

4.8 m) forces the motion matching engine to select different stride sequences, resulting in distinct entry poses such as left-leg versus right-leg leads. To prevent non-causal shortcuts (e.g., relying on elapsed time or step count), we randomize the pre-skill locomotion duration by sampling it uniformly from $[0.1, 3]$ s, with an average interval of 0.3 s. Such diverse motion-terrain pairs encourage context-based reaction, and are critical for learning a policy that can reliably trigger the correct skill under varying distances and timings.

Terrain Randomization. To improve robustness beyond the training obstacles while keeping the reference feasible, we randomize obstacle geometry and pose around each synthesized trajectory. Specifically, obstacle width is sampled from the minimum required by the reference up to 1.5 m; the remaining dimensions are perturbed within ± 5 cm; and obstacle yaw is randomized within $\pm 45°$, as illustrated in Fig. 3(c). This exposes the policy to variations in obstacle shape and pose without invalidating the underlying reference.

Distractors. We place distractor boxes with random sizes and poses near the reference trajectory to improve robustness to irrelevant objects and reduce overfitting in the real world.

C. Learning a Highly-Dynamic Visuomotor Policy

Our goal is to train a single perceptive policy capable of various long-horizon parkour skills. Commanded by a target velocity, the humanoid will autonomously perform various parkour skills based on the obstacles it perceives. Because the skills are highly dynamic, we train skill-specific experts to achieve high motion quality and then distill them into a single visuomotor policy. To ensure scalability, we use a unified expert and distillation formulation without motion-specific tuning.

1) Training Expert Policies with Motion Tracking: We follow BeyondMimic [20] and OmniRetarget [43] for motion tracking, and refer readers to these prior works for details.

Observations include reference joint position/velocity, reference pelvis pose error, pelvis linear/angular velocity, joint position/velocity, and the previous action. We additionally provide the expert with a 0.7 m × 0.7 m height scan, allowing it to adapt to terrain randomizations.

Unlike [43], we enable global tracking with privileged observations (pelvis global position and velocity) so the expert can learn recovery behaviors. This is important because the reference motion is tightly coupled with the terrain, meaning small drift or timing errors can quickly accumulate and must be corrected to stay on the intended trajectory. While these privileged states are not available on hardware, they can be inferred from visual inputs by the student policy.

Adaptive Sampling is essential for learning difficult skills in expert training, which prioritizes sampling from regions that fail more frequently. For example, without it, the high-wall climbing expert fails to converge to a meaningful behavior.

Rewards, Terminations, and Domain Randomization follow BeyondMimic [20]: DeepMimic-style tracking rewards with action rate, joint limits, and collision penalties, tracking-based early termination, and lightweight randomizations.

Actions are joint PD targets normalized by a fixed action scale. Due to challenging RL exploration, we set the action scale to 1 for all experts, instead of the heuristics used in [20].

2) Distilling a Unified Student Policy with DAgger and RL: A common approach for learning a unified policy from multiple experts is to apply DAgger-style imitation learning [32, 8, 52, 33, 20]. While effective for easier motions such as stepping, we find that DAgger alone is insufficient for highly dynamic skills such as climbing and vaulting. These skills depend on brief, high-magnitude torque bursts, but per-step imitation objectives like DAgger do not account for episode outcomes and therefore do not explicitly favor such high-torque actions. For example, actions that result in higher or lower root positions that are symmetric about the reference may receive identical DAgger loss, even though only the higher-root trajectory successfully clears the obstacle.

To address this, we apply PPO alongside DAgger with a curriculum,

$$\mathcal{L} = \lambda_{\text{PPO}}\,\mathcal{L}_{\text{PPO}} + \lambda_D\,\mathcal{L}_D, \qquad \lambda_{\text{PPO}} + \lambda_D = 1, \tag{3}$$

where λ_{PPO} and λ_D are their curriculum weights. Note that the primary role of PPO is to provide a success-driven signal that encourages exploiting expert behaviors, such as high-torque actions, rather than exploring beyond the expert skill distribution. This hybrid setup substantially improves the unified policy's performance on diverse, highly dynamic skills.

Observations, Actions, and Domain Randomization. The policy observes proprioception signals including pelvis gravity vector and angular velocity, joint positions and velocities, and the previous action. For vision, we use depth images rendered with Nvidia WARP [25] for high-throughput training. The policy also receives velocity commands defined in Sec. III-B2. The action space and domain randomization for sim-to-real transfer are identical to those used in expert training setup.

Camera modeling and depth artifacts. We calibrate the simulated camera by matching robot self-visibility across a set of poses in simulation and hardware using ROI overlap, and randomize camera extrinsics within 2.5 cm translation and 2.5° rotation around the calibrated value to improve robustness to viewpoint shifts and mounting variability. We inject realistic depth noise following prior work [33], but exclude Gaussian blur since it can obscure obstacles at high speed. Finally, we randomize observation delay between 60 ms and 80 ms to simulate hardware latency fluctuations.

Curriculum. Since PPO gradients are noisy in the early stages and can otherwise undermine distillation, we apply a warmup curriculum that gradually shifts from DAgger to PPO. The curriculum includes three parts.

First, we linearly tune down λ_D in Eq. (3) during the first half of training, capped at 0.1: $\lambda_D(k) = \max\left(0.1, 1 - \frac{k}{K/2}\right)$, where k is the current iteration and K is the total iterations.

Second, left-right symmetry introduces multimodality: many skills admit two equally valid mirrored executions, for example, clearing a hurdle with either the left- or right-leg lead, while the reference trajectory represents only one of them. As a result, the distilled policy may perform the

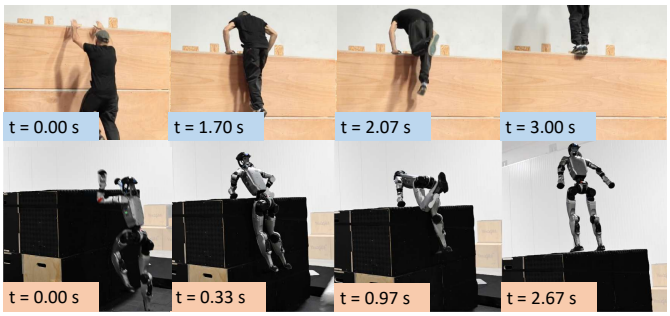

Fig. 4: Side-by-side comparison of high-climb agility. The robot climbs onto a 1.25 m wall within 2.67 s.

mirrored mode, which still completes the skill but incurs a large tracking error and would be terminated incorrectly. This spurious failure signal can cause high reward variance for PPO. To mitigate this issue, we relax the termination threshold from 0.5 m for the expert to 1 m using the same linear schedule, so mirrored modes are not terminated prematurely.

Finally, we enable adaptive learning rate and KL-based exploration control only when λ_{PPO} exceeds 0.1.

Adaptive Sampling of rollout start points is disabled during student training. While it helps experts focus on failures to learn more difficult segments, it can undersample "borderline" clips that do not fail in simulation but exhibit jittery behavior, which often leads to large sim-to-real gap on hardware. To further avoid data imbalance across skills, we sample each skill evenly and sample uniformly within each skill.

IV. Experiments

We evaluate the proposed framework through a series of simulation and real-world experiments on a Unitree G1 humanoid (1.3 m tall with 29 DoFs). For training, we use a 3-layer CNN and a 5-layer MLP with hidden sizes [2048, 1024, 512, 256, 128], trained with 16,384 parallel environments. Both expert and student policies are trained for 20K iterations.

A. Real-World Results

We evaluate our system on real-world parkour tasks requiring both highly-dynamic individual skills, long-horizon multi-skill composition and adaptation to environmental changes. All skill execution is autonomous, while only simple 2D velocity commands are provided for navigation.

1) Human-Level Agility: We first demonstrate that the robot can execute highly dynamic parkour skills, including a direct comparison with a human parkourist on a challenging high-wall climb.

High-Wall Climb with Human Comparison. We compare the robot's high-wall climb against a human performing the same maneuver [34]. While often considered a fundamental parkour technique, the high-wall climb demands substantial upper-body strength and precise whole-body coordination, and remains difficult for untrained individuals. Despite this, the robot successfully performs the climb at a pace comparable to the human. As shown in Fig. 4, the robot executes a fast and coherent sequence with closely matched timing across key events (toe-off $\rightarrow$ pull-up $\rightarrow$ swing $\rightarrow$ stable stand). For a 1.25 m wall (96% of the robot's height), the robot climbs onto the platform in 2.67 s measured from toe-off.

Additional Parkour Skills. We demonstrate additional highly dynamic parkour skills that require rapid contact transitions and momentum preservation. As shown in Fig. 5(a), the robot clears a 0.4 m-high, 0.5 m-long obstacle within 0.8 s from toe-off to toe-on, while covering more than 2 m forward (154% of its height). The motion reaches a peak forward speed of 3.41 m/s with an average speed of 2.53 m/s, highlighting its effective momentum preservation across the contact. Fig. 5(b) shows a drop landing from a 1.25 m platform. Upon landing, the robot flexes its lower-body joints to absorb impact and stabilize its posture.

2) Multi-Obstacle Course: A key strength of our framework is that the policy generalizes to complex, multi-obstacle courses despite the training data containing only single-obstacle traversal. This capability emerges from motion-matching-based composition, which synthesizes long-horizon reference trajectories that explicitly chain skills through shared locomotion segments and expose the policy to diverse approach distances and timings. As a result, the policy learns to execute skills reliably across varying obstacle sequences without explicit multi-obstacle supervision.

Various Skills and Adaptivity to Obstacle Changes. As shown in Fig. 5(c), the robot composes multiple skills, including stepping, low and high wall climb, into a continuous run over courses with several obstacles. The visuomotor policy generates transitions online, enabling smooth skill switching throughout the traversal. We further demonstrate closed-loop adaptivity by randomly displacing multiple obstacles by approximately 0.5 m during execution. The policy adapts by adjusting its approach and maneuver timing, allowing the robot to continue and complete the traversal in response to these obstacle changes. These results demonstrate the adaptivity of our policy in long-horizon terrain traversal.

B. Quantitative Results in Simulation

1) Experiment Setup: We evaluate all methods using success rate on parkour traversal tasks. Each task requires the robot to move forward at a fixed command speed (1.0 m/s or 2.0 m/s) and clear a single obstacle of a specified height and 20° yaw randomization. To vary approach conditions, the humanoid is initialized at a random distance in front of the obstacle: for 1 m/s tasks, distances are sampled uniformly from 1.5 m to 3.0 m; for 2 m/s tasks, from 3.0 m to 4.5 m. For each task, we evaluate 100 obstacle instances with different initial distances. A trial is successful if the robot traverses the obstacle and travels an additional 1.5 m without falling within a fixed time horizon. We report average success rates of 5 trials per obstacle per task (500 trials total per task). We train all variants with the full skill set, with details in Appx.B.

2) Baseline Comparison: We evaluate the contribution of three key components: human reference motions, motion-matching-based skill composition, and the two-stage teacher-student training framework, by comparing our method against the following baselines, as shown in Table I.

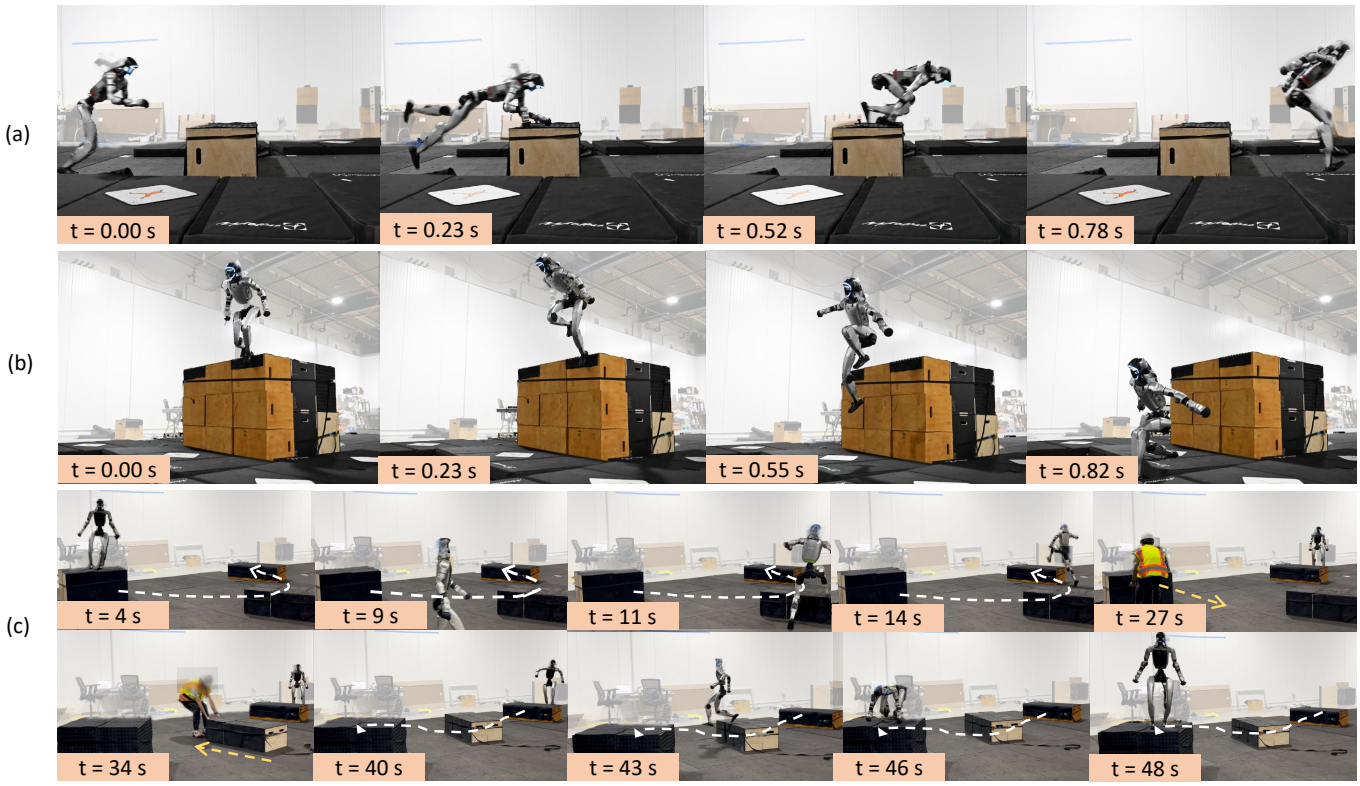

Fig. 5: Hardware results demonstrating agile, long-horizon parkour behaviors, including (a) a cat vault, (b) a drop landing from a 1.25 m wall, and (c) a 48-second terrain traversal with online adaptation to real-time obstacle displacement.

TABLE I: Baseline success rate on parkour tasks with different commanded speeds and obstacle heights.

Commanded Velocity	1.0 m/s			2.0 m/s		
	36 cm	58 cm	76 cm	36 cm	58 cm	76 cm
Velocity Tracking	**1.00**	0.00	0.00	**1.00**	0.00	0.00
Uncomposed Data	0.06	0.02	0.00	0.37	0.27	0.07
End-to-end Depth	0.95	0.07	0.08	0.78	0.19	0.14
Ours	**1.00**	**0.99**	**0.95**	**1.00**	**0.99**	**0.95**

- **Velocity Tracking.** We train a humanoid to traverse terrain using IsaacLab's [27] standard velocity-tracking RL pipeline. The policy is learned purely with reward shaping, without any human reference motion.
- **Uncomposed Motion Data.** This baseline removes motion matching, training instead on uncomposed locomotion data and atomic parkour skill clips.
- **End-to-end Depth Policy.** This baseline removes distillation and trains a single depth-based visuomotor policy end-to-end on the motion-matching data, using the same observations as the student and the same motion-tracking reward as the experts.

We found that the **velocity-tracking** baseline achieves similar performance to prior reward-shaping works [53, 22] and succeeds in traversing the 36 cm obstacle, but fails on higher obstacles. Specifically, it largely relies on foot-only stepping and does not discover whole-body climbing strategies that use the arms for support, highlighting the limitations of reward-shaping RL alone for highly dynamic parkour.

The **uncomposed motion data** baseline performs poorly despite access to atomic skills, showing that isolated motions are insufficient. A common failure mode is that the robot walks up to an obstacle but fails to climb or jump over it. Without explicit long-horizon composition, the policy neither experiences skill transitions during training nor observes obstacles during the walking phase to prepare the appropriate upcoming skill. In comparison, our motion-matching-based approach addresses this limitation by both generating coherent long-horizon skill composition with smooth transitions and exposing the policy to diverse visual contexts during skill execution.

While **end-to-end depth-based** training can handle low obstacles, its performance degrades on more challenging tasks, suggesting difficulty in RL exploration when training from scratch. In contrast, our expert-distillation pipeline achieves substantially higher success rates across obstacle heights, particularly for highly dynamic skills.

3) Ablation Study: We conduct ablation studies to study the effect of motion-matching reference data density, training scalability, and the role of RL during policy distillation (Table II).

Motion Matching Density. We hypothesize that diversity in motion-matching data, especially approach distances, is critical for accurate timing and task success. To test this, we ablate approach-distance coverage in the reference dataset:

- **Extreme Distances.** Only minimum and maximum approach distances.

TABLE II: Success rate on parkour tasks with different motion matching densities and RL strategies during distillation.

Method	**1.0 m/s**			**2.0 m/s**		
	58 cm	76 cm	94 cm	36 cm	58 cm	76 cm
Extreme Distances	0.99	0.62	0.64	0.98	0.60	0.58
Half Density	0.95	0.32	0.57	0.99	0.85	0.81
DAgger Only	0.16	0.03	0.12	0.63	0.09	0.10
DAgger & Alive Reward	**1.00**	0.90	0.96	0.94	0.91	0.84
DAgger & Root Tracking	**1.00**	0.79	0.75	**1.00**	0.92	0.87
1/4 Training Envs	0.97	0.00	0.59	0.94	0.65	0.58
1/2 Training Envs	0.94	0.60	0.68	0.97	0.79	0.75
3-layer MLP	0.99	0.02	0.00	0.98	0.89	0.81
4-layer MLP	**1.00**	0.94	0.08	1.00	0.94	0.88
Ours	0.99	**0.95**	**1.00**	**1.00**	**0.98**	**0.90**

- **Half Density.** Randomly selected half of the full motion-matching data.

Using **Extreme distances** data leads to reduced success rates across all tasks, as the policy fails to generalize to intermediate distances where contact-timing is critical. Training on **Half density** data generally yields lower success rates on harder skills, especially when the remaining samples are skewed toward one end of the distance range. For example, in the 1.0 m/s climbing task on 76 cm and 94 cm obstacles, reduced local density leads to unreliable hand-placement timing. In contrast, the full dataset densely covers approach conditions, enabling robust skill execution across varying approach distances.

Training Scalability. We ablate the number of parallel training environments and model capacity to assess the scalability.

- **1/4 Training Envs.** Use 1/4 of the training environments.
- **1/2 Training Envs.** Use 1/2 of the training environments.
- **3-Layer MLP.** Use a 3-layer MLP with hidden sizes of [512, 256, 128].
- **4-Layer MLP.** Use a 4-layer MLP with hidden sizes of [1024, 512, 256, 128].

Unlike training from scratch, where additional rollouts often yield diminishing returns due to exploration limits, our distillation framework scales favorably with both model capacity and rollout throughput. Increasing the number of parallel environments or using a deeper network generally improves success, especially on more challenging parkour tasks.

RL in Distillation. We ablate the RL objective and its reward design in the distillation stage to understand its role.

- **DAgger Only.** Remove the RL loss during distillation.
- **DAgger + RL Alive Reward.** Use only an alive/progress reward, without motion-tracking terms.
- **DAgger + RL Root Tracking Reward.** Use a root-tracking reward instead of full whole-body tracking.

We find that RL is critical for effective distillation. The **DAgger only** student exhibits a clear performance drop, indicating that DAgger alone is insufficient to capture highly dynamic skills even with strong experts. For example, on the 76 cm obstacle, the DAgger student consistently stalls at the pull-up phase: although it learns the accurate hand placement, it fails to produce the brief, high-magnitude torque burst needed to lift the torso. As discussed in Sec. III-C2, this likely occurs because the decisive torque burst spans only a few timesteps, and per-step imitation loss barely penalizes slightly underestimated actions. In contrast, since RL accounts for episode success, it encourages torque bursts that are more likely to complete the pull-up, yielding both higher reward and lower DAgger loss.

We further evaluate how sensitive the DAgger+RL stage is to reward design. Interestingly, using **root tracking** or even only an **alive reward** achieves success rates comparable to whole-body tracking on difficult skills. This suggests that, when co-trained with DAgger, RL is relatively robust to reward choice and mainly acts as a success-driven exploitation signal that compensates for DAgger's underestimation, rather than relying on detailed task-specific shaping. Accordingly, while we use whole-body tracking in this work, a simple alive reward may suffice when scaling to larger skill sets.

In addition, our approach differs from prior work that first trains a strong DAgger policy and then applies a separate RL fine-tuning stage [33]. Here, we use RL during distillation to correct imitation-induced conservatism and improve skill learning. We also find that the DAgger term must remain active throughout training: if we drop the DAgger loss after the curriculum and continue with pure RL, the policy often develops jittery, unnatural behaviors, suggesting that in a high-dimensional action space the behavior cloning objective provides a critical regularization for RL.

V. Conclusion

We have presented Perceptive Humanoid Parkour, a modular framework that enables humanoid robots to autonomously execute long-horizon, highly dynamic parkour behaviors using onboard perception. By combining motion-matching-based skill composition with a teacher-student RL pipeline, our approach preserves the agility of human motions while enabling perception-driven adaptation to diverse obstacles. We find that dense motion matching is critical in providing coherent long-horizon references and exposes the policy to a wide range of approach conditions, while augmenting distillation with RL transfers the capability from single-skill, privileged-information experts to the multi-skill, depth-based student efficiently. Through extensive simulation studies and zero-shot deployment on a Unitree G1 robot, we demonstrate state-of-the-art agile, adaptive, whole-body parkour in the real world.

While our pipeline enables long-horizon, highly dynamic humanoid parkour, it currently lacks semantic scene understanding. Incorporating richer conditioning signals, such as language, could enable finer control over diversity and styles. In addition, our real-world capabilities are constrained by perception and hardware. With a short-range, narrow field-of-view camera at a high running speed, obstacle geometries may not be visible sufficiently early, forcing the robot to commit under perceptual ambiguity. Improved sensing and semantic scene understanding could reduce this ambiguity and support richer context reasoning. Finally, our hardware lacks sufficiently strong hands or grippers for interactions with edges and bars to be tested, preventing more extreme climbing beyond the robot's height or hanging maneuvers.

References

[1] Ananye Agarwal, Ashish Kumar, Jitendra Malik, and Deepak Pathak. Legged locomotion in challenging terrains using egocentric vision. In *Conference on robot learning*, pages 403–415. PMLR, 2023.

[2] 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, 2025. URL https://arxiv.org/abs/2511.14625.

[3] Kevin Bergamin, Simon Clavet, Daniel Holden, and James Richard Forbes. Drecon: data-driven responsive control of physics-based characters. *ACM Transactions On Graphics (TOG)*, 38(6):1–11, 2019.

[4] David Bollo. Inertialization: High-performance animation transitions in Gears of War. Proc. of GDC, 2018.

[5] Michael Büttner and Simon Clavet. Motion matching - the road to next gen animation. Proc. of Nucl.ai, 2015.

[6] Ken Caluwaerts, Atil Iscen, J Chase Kew, Wenhao Yu, Tingnan Zhang, Daniel Freeman, Kuang-Huei Lee, Lisa Lee, Stefano Saliceti, Vincent Zhuang, et al. Barkour: Benchmarking animal-level agility with quadruped robots. *arXiv preprint arXiv:2305.14654*, 2023.

[7] 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.

[8] Xuxin Cheng, Kexin Shi, Ananye Agarwal, and Deepak Pathak. Extreme parkour with legged robots. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11443–11450. IEEE, 2024.

[9] Simon Clavet. Motion matching and the road to next-gen animation. Proc. of GDC, 2016.

[10] Zipeng Fu, Qingqing Zhao, Qi Wu, Gordon Wetzstein, and Chelsea Finn. Humanplus: Humanoid shadowing and imitation from humans. *arXiv preprint arXiv:2406.10454*, 2024.

[11] Ruiyu Gou, Michiel van de Panne, and Daniel Holden. Control operators for interactive character animation. *ACM Transactions on Graphics (TOG)*, 2025.

[12] 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.

[13] David Hoeller, Nikita Rudin, Dhionis Sako, and Marco Hutter. Anymal parkour: Learning agile navigation for quadrupedal robots. *Science Robotics*, 9(88):eadi7566, 2024.

[14] Daniel Holden, Anas Kanoun, Michiel Bŭttner, Sofien Bouaziz, Sebastian Thrun, and Aaron Hertzmann. Learned motion matching. *ACM Transactions on Graphics (TOG)*, 2020.

[15] Xiaoyu Huang, Takara Truong, Yunbo Zhang, Fangzhou Yu, Jean Pierre Sleiman, Jessica Hodgins, Koushil Sreenath, and Farbod Farshidian. Diffuse-cloc: Guided diffusion for physics-based character look-ahead control. *arXiv preprint arXiv:2503.11801*, 2025.

[16] Dvij Kalaria, Sudarshan S Harithas, Pushkal Katara, Sangkyung Kwak, Sarthak Bhagat, Shankar Sastry, Srinath Sridhar, Sai Vemprala, Ashish Kapoor, and Jonathan Chung-Kuan Huang. Dreamcontrol: Human-inspired whole-body humanoid control for scene interaction via guided diffusion. *arXiv preprint arXiv:2509.14353*, 2025.

[17] Dongho Kang, Simon Zimmermann, and Stelian Coros. Animal gaits on quadrupedal robots using motion matching and model-based control. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 8500–8507. IEEE, 2021.

[18] Ashish Kumar, Zipeng Fu, Deepak Pathak, and Jitendra Malik. Rma: Rapid motor adaptation for legged robots. *arXiv preprint arXiv:2107.04034*, 2021.

[19] Joonho Lee, Jemin Hwangbo, Lorenz Wellhausen, Vladlen Koltun, and Marco Hutter. Learning quadrupedal locomotion over challenging terrain. *Science robotics*, 5 (47):eabc5986, 2020.

[20] 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.

[21] Junfeng Long, Zirui Wang, Quanyi Li, Jiawei Gao, Liu Cao, and Jiangmiao Pang. Hybrid internal model: Learning agile legged locomotion with simulated robot response. *arXiv preprint arXiv:2312.11460*, 2023.

[22] Junfeng Long, Junli Ren, Moji Shi, Zirui Wang, Tao Huang, Ping Luo, and Jiangmiao Pang. Learning humanoid locomotion with perceptive internal model. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9997–10003. IEEE, 2025.

[23] Shixin Luo, Songbo Li, Ruiqi Yu, Zhicheng Wang, Jun Wu, and Qiuguo Zhu. Pie: Parkour with implicit-explicit learning framework for legged robots. *IEEE Robotics and Automation Letters*, 2024.

[24] 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.

[25] Miles Macklin. Warp: A high-performance python framework for gpu simulation and graphics. https://github.com/nvidia/warp, March 2022. NVIDIA GPU Technology Conference (GTC).

[26] Takahiro Miki, Joonho Lee, Jemin Hwangbo, Lorenz Wellhausen, Vladlen Koltun, and Marco Hutter. Learning robust perceptive locomotion for quadrupedal robots in the wild. *Science robotics*, 7(62):eabk2822, 2022.

[27] 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.

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

[29] Yixuan Pan, Ruoyi Qiao, Li Chen, Kashyap Chitta, Liang Pan, Haoguang Mai, Qingwen Bu, Hao Zhao, Cunyuan Zheng, Ping Luo, et al. Agility meets stability: Versatile humanoid control with heterogeneous data. *arXiv preprint arXiv:2511.17373*, 2025.

[30] 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.

[31] 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)*, 2021.

[32] Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 627–635. JMLR Workshop and Conference Proceedings, 2011.

[33] Nikita Rudin, Junzhe He, Joshua Aurand, and Marco Hutter. Parkour in the wild: Learning a general and extensible agile locomotion policy using multi-expert distillation and rl fine-tuning. *arXiv preprint arXiv:2505.11164*, 2025.

[34] Salgadopk. Learn parkour - climb up tutorial. URL https://youtu.be/6U1sIgqgPFo?si=339TPTxlFB5lWGB1.

[35] Jingkai Sun, Gang Han, Pihai Sun, Wen Zhao, Jiahang Cao, Jiaxu Wang, Yijie Guo, and Qiang Zhang. Dpl: Depth-only perceptive humanoid locomotion via realistic depth synthesis and cross-attention terrain reconstruction. *arXiv preprint arXiv:2510.07152*, 2025.

[36] Guy Tevet, Sigal Raab, Brian Gordon, Yoni Shafir, Daniel Cohen-or, and Amit Haim Bermano. Human motion diffusion model. In *ICLR*, 2023.

[37] Huayi Wang, Zirui Wang, Junli Ren, Qingwei Ben, Tao Huang, Weinan Zhang, and Jiangmiao Pang. Beamdojo: Learning agile humanoid locomotion on sparse footholds. In *Robotics: Science and Systems (RSS)*, 2025.

[38] Huayi Wang, Wentao Zhang, Runyi Yu, Tao Huang, Junli Ren, Feiyu Jia, Zirui Wang, Xiaojie Niu, Xiao Chen, Jiahe Chen, et al. Physhsi: Towards a real-world generalizable and natural humanoid-scene interaction system. *arXiv preprint arXiv:2510.11072*, 2025.

[39] Jinze Wu, Guiyang Xin, Chenkun Qi, and Yufei Xue. Learning robust and agile legged locomotion using adversarial motion priors. *IEEE Robotics and Automation Letters*, 8(8):4975–4982, 2023.

[40] 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.

[41] Michael Xu, Yi Shi, KangKang Yin, and Xue Bin Peng. Parc: Physics-based augmentation with reinforcement learning for character controllers. In *ACM SIGGRAPH*, 2025.

[42] Pei Xu, Zhen Wu, Ruocheng Wang, Vishnu Sarukkai, Kayvon Fatahalian, Ioannis Karamouzas, Victor Zordan, and C Karen Liu. Learning to ball: Composing policies for long-horizon basketball moves. *ACM Transactions on Graphics (TOG)*, 44(6):1–14, 2025.

[43] Lujie Yang, Xiaoyu Huang, Zhen Wu, Angjoo Kanazawa, Pieter Abbeel, Carmelo Sferrazza, C Karen Liu, Rocky Duan, and Guanya Shi. Omniretarget: Interaction-preserving data generation for humanoid whole-body loco-manipulation and scene interaction. *arXiv preprint arXiv:2509.26633*, 2025.

[44] Ruihan Yang, Ge Yang, and Xiaolong Wang. Neural volumetric memory for visual locomotion control. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1430–1440, 2023.

[45] Tae Hoon Yang, Haochen Shi, Jiacheng Hu, Zhicong Zhang, Daniel Jiang, Weizhuo Wang, Yao He, Zhen Wu, Yuming Chen, Yifan Hou, et al. Locomotion beyond feet. *arXiv preprint arXiv:2601.03607*, 2026.

[46] Alan Yu, Ge Yang, Ran Choi, Yajvan Ravan, John Leonard, and Phillip Isola. Learning visual parkour from generated images. In *8th Annual Conference on Robot Learning*, 2024.

[47] Ruiqi Yu, Qianshi Wang, Yizhen Wang, Zhicheng Wang, Jun Wu, and Qiuguo Zhu. Walking with terrain reconstruction: Learning to traverse risky sparse footholds. *arXiv preprint arXiv:2409.15692*, 2024.

[48] Yanjie Ze, Siheng Zhao, Weizhuo Wang, Angjoo Kanazawa, Rocky Duan, Pieter Abbeel, Guanya Shi, Jiajun Wu, and C Karen Liu. Twist2: Scalable, portable, and holistic humanoid data collection system. *arXiv preprint arXiv:2511.02832*, 2025.

[49] Tong Zhang, Boyuan Zheng, Ruiqian Nai, Yingdong Hu, Yen-Jen Wang, Geng Chen, Fanqi Lin, Jiongye Li, Chuye Hong, Koushil Sreenath, et al. Hub: Learning extreme humanoid balance. *CoRL*, 2025.

[50] Ziyu Zhang, Sergey Bashkirov, Dun Yang, Michael Taylor, and Xue Bin Peng. Add: Physics-based motion imitation with adversarial differential discriminators. *arXiv preprint arXiv:2505.04961*, 2025.

[51] Shaoting Zhu, Ziwen Zhuang, Mengjie Zhao, Kun-Ying Lee, and Hang Zhao. Hiking in the wild: A scalable perceptive parkour framework for humanoids. *arXiv preprint arXiv:2601.07718*, 2026.

[52] Ziwen Zhuang, Zipeng Fu, Jianren Wang, Christopher Atkeson, Soeren Schwertfeger, Chelsea Finn, and Hang Zhao. Robot parkour learning. *arXiv preprint arXiv:2309.05665*, 2023.

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

APPENDIX

A. Motion Matching Implementation Details

This section provides implementation details for the motion matching procedure used to synthesize long-horizon parkour reference trajectories.

1) Motion Database and Feature Precomputation: All motion clips are first retargeted to a 29-DOF Unitree G1 humanoid using OmniRetarget [43] and represented as frame sequences. At each frame i, we store the robot configuration $\boldsymbol{q}_i = (\boldsymbol{p}_i, \boldsymbol{r}_i, \boldsymbol{\theta}_i)$, consisting of the root translation $\boldsymbol{p}_i \in \mathbb{R}^3$, root quaternion $\boldsymbol{r}_i \in \mathbb{R}^4$, and joint angles $\boldsymbol{\theta}_i \in \mathbb{R}^{29}$. For each frame, we also precompute a matching feature vector $\boldsymbol{x}_i$ derived from $\boldsymbol{q}_i$. Following [14], $\boldsymbol{x}_i \in \mathbb{R}^{27}$ is expressed in the character's local coordinate frame and consists of:

- **Future root trajectory $\boldsymbol{t}_i \in \mathbb{R}^{12}$:** Planar root positions and facing directions at 0.33s, 0.67s, and 1s into the future.
- **Local foot state $\boldsymbol{f}_i \in \mathbb{R}^{12}$:** Positions and linear velocities of the left and right feet expressed in the root frame.
- **Root velocity $\boldsymbol{h}_i \in \mathbb{R}^3$:** Root linear velocity.

To improve data coverage, we augment the motion database by mirroring all motion clips.

For parkour motion clips, we manually fit a box-shaped terrain aligned with each motion. The terrain is not included in the motion-matching query; the query remains motion-centric and follows standard motion-matching practice. Instead, we treat each parkour clip together with its fitted terrain as a paired motion-terrain example. After motion matching selects a skill entry frame, the corresponding terrain is instantiated using the precomputed terrain-to-root offset at that frame, preserving the contact relationship between the motion and the obstacle during synthesis. Since this terrain pairing is also required for downstream RL training, it is not an additional cost specific to motion matching. The overhead for terrain annotation is also small: each box takes minutes to fit, and typically only 1–2 paired clips are needed per new skill.

2) Query Feature Construction: At runtime, a query feature $\hat{\boldsymbol{x}}_t$ is constructed from the current robot configuration $\boldsymbol{q}_t$ and a 2D velocity command. We first extract the kinematic features from $\boldsymbol{q}_t$ to form the pose-based part of the query, namely the local foot state $\hat{\boldsymbol{f}}_t$ and the root velocity $\hat{\boldsymbol{h}}_t$. We then compute the short-horizon future root trajectory from the 2D velocity command to form the command-based part $\hat{\boldsymbol{t}}_t$.

Following [14], we convert the 2D velocity command into a future root trajectory using a critically damped spring model. We apply the spring to (i) the 2D root velocity and (ii) the root heading direction. The target 2D velocity is set to the commanded 2D velocity $\boldsymbol{u}_t^{\text{cmd}} \in \mathbb{R}^2$. The target heading ψ_t^{cmd} is set to $\operatorname{atan2}(u_{t,y}^{\text{cmd}}, u_{t,x}^{\text{cmd}})$.

Critically damped spring closed form. Let $\boldsymbol{s}$ denote the spring position and $\dot{\boldsymbol{s}}$ its velocity, with goal $\boldsymbol{s}_{\text{goal}}$ and damping parameter $y > 0$. Define $\boldsymbol{j}_0 = \boldsymbol{s}_0 - \boldsymbol{s}_{\text{goal}}$ and $\boldsymbol{j}_1 = \dot{\boldsymbol{s}}_0 + y\,\boldsymbol{j}_0$. Then the spring state at any future time τ admits the closed form

$$\boldsymbol{s}(\tau) = e^{-y\tau}\big(\boldsymbol{j}_0 + \tau \boldsymbol{j}_1\big) + \boldsymbol{s}_{\text{goal}}. \tag{4}$$

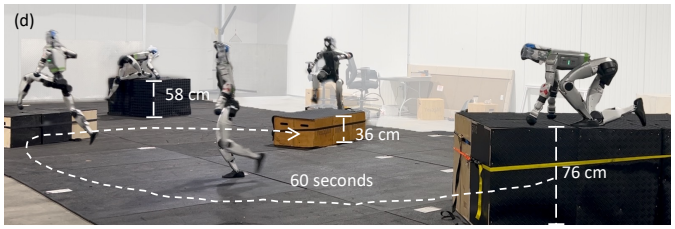

Fig. 6: Visualization of tasks evaluated.

Planar position from target velocity. For planar root translation, we use the spring in velocity space: the spring "position" corresponds to planar velocity (and its derivative to acceleration). We obtain future root positions by integrating the closed-form velocity:

$$\boldsymbol{p}(\tau) = \boldsymbol{p}_0 - \frac{\boldsymbol{j}_1}{y^2} e^{-y\tau} + \frac{-\boldsymbol{j}_0 - \tau \boldsymbol{j}_1}{y} e^{-y\tau} + \frac{\boldsymbol{j}_1}{y^2} + \frac{\boldsymbol{j}_0}{y} + \boldsymbol{u}_t^{\text{cmd}}\,\tau, \tag{5}$$

applied component-wise in the plane.

Heading direction from target heading. For rotation, we apply the spring directly to the heading angle ψ toward ψ_t^{cmd} and evaluate the resulting $\psi(\tau)$ at the same horizons via Eq. (4) (no integration is needed).

We evaluate at $\tau \in \{0.33, 0.67, 1.0\}$ s, and transform the resulting future positions $\boldsymbol{p}(\tau)$ and facing directions into the character's local coordinate frame to form the future trajectory feature $\hat{\boldsymbol{t}}_t$.

3) Transition Smoothing via Inertialization: To ensure smooth transitions when switching the playback index to a newly retrieved frame, we adopt inertialization [4]. The key idea is to compute an offset between the currently playing motion and the target motion at the transition instant, apply this offset after switching so the output remains continuous, and then gradually decay the offset to zero. We decay this offset using the same critically damped spring model as in Eq. (4), but with the goal set to zero.

B. Skill List and Training Implementation Details

1) Skill List: Our motion library includes locomotion and a set of atomic parkour skills. Locomotion provides a shared transition manifold for entering and exiting skills, avoiding explicit pairwise transitions between every pair of skills; as a result, our library scales *linearly* with the number of skills. Specifically, the locomotion library includes standing, walking, and running motions spanning commanded speeds from 0.8 to 3.5 m/s. Most parkour skills are instantiated at 1.0 m/s and 2.0 m/s. We additionally include a single 3.0 m/s cat-vault skill to cover extreme-speed vaulting behaviors. Table III summarizes the full skill list and the total duration of motion clips for each category. We use a different parkour skill or style for each combination of locomotion speed and obstacle height. For clarity, we visualize the heights of the common obstacles evaluated in this work in Fig. 6.

2) Motion Tracking Details: Specific reward formulations and domain randomization settings used for expert policy learning from [20] are summarized in Table VII and Table VIII for reference.

Skill	Duration (s)
Locomotion	
Locomotion	495.5
Parkour skills @ 1.0 m/s	
Step (36 cm)	2.2
Climb (58 cm)	12.1
Climb (76 cm)	8.8
Climb (94 cm)	10.3
Parkour skills @ 2.0 m/s	
Step (36 cm)	1.6
Climb (58 cm)	6.1
Climb (76 cm)	4.4
Climb (94 cm)	5.2
Climb (125 cm)	5.9
Dash Vault	5.0
Speed Vault	3.1
Parkour skills @ 3.0 m/s	
Cat Vault	1.5

TABLE III: Motion clips used in our motion library.

TABLE IV: Real-world benchmark results.

Task	Success	Failure: Topple	Failure: Bad Wrist Pose
Step 36 cm	9/10	1/10	0/10
Climb 58 cm	10/10	0/10	0/10
Climb 76 cm	8/10	2/10	0/10
Climb 125 cm	4/8	1/8	3/8
Cat Vault	6/8	1/8	1/8

3) Distillation Details: During student training, we relax the termination conditions relative to the expert to prevent premature termination of valid but mirrored executions. While this improves PPO stability, the student may visit states that are out-of-distribution for the expert policies, which were trained under the original termination thresholds and may not provide meaningful actions in these regimes. In particular, when the student violates the expert's original termination condition but remains within the relaxed one, querying the expert would yield unreliable supervision. To avoid introducing incorrect DAgger signals, we disable the DAgger loss at such timesteps and rely solely on the PPO objective.

For depth sensing, beyond the aforementioned camera noise, we also add a random depth offset within ± 3 cm and inject i.i.d. Gaussian noise with a standard deviation of 3 cm into the depth observations during training. The onboard depth camera operates at 30 Hz.

4) Training Hyperparameters: We include all hyperparameters for two-stage training in Table IX for reference.

C. Additional Experiment Details

1) Real-world Failure Modes: We further report hardware success rates for each real-world task and categorize failures in Table IV. Due to hardware constraints, we conduct only a limited number of trials; therefore, these results are not intended as a statistically significant benchmark. Across trials, failures most commonly fall into two modes: (1) the robot approaches too close to the lateral edge and topples over; and (2) the wrist fails to support the body or the climbing impact. The first failure mode can likely be mitigated by increasing motion and terrain coverage for edge-case approach states, so that the policy observes more near-boundary configurations during training and learns to recover from them. The second failure mode reflects the hardware limits of the current platform and may be addressed by improved wrist hardware or training objectives that explicitly discourage motor torque saturation during high-impact climbs. We exclude two hardware-specific failures unrelated to our framework: thermal stress and overcurrent protection, which occur approximately every five wall-climb trials. To mitigate these effects, we cool down the robot for 10 minutes after every five trials.

TABLE V: Robot and human wall-climb comparison. OH denotes obstacle height; CT denotes completion time; S denotes the number of participants who successfully complete the task.

Task	Robot OH	Robot CT	Human OH	Human CT	Human S
Mid Wall	76 cm	1.94 s	105 cm	**1.89** s	4/4
High Wall	125 cm	**2.63** s	160 cm	3.51 s	3/4

2) Quantitative Comparison with Human: To further assess whether the learned policy exhibits human-level agility, we conduct an informal, small-scale quantitative comparison with four adult participants on comparable wall-climbing tasks. Since G1 is substantially shorter than adults, to match the same obstacle-to-body-height ratio, we set the robot obstacle heights to approximately 70% of the corresponding human obstacle heights. As shown in Table V, the robot performs slightly slower than humans on the mid-wall task but substantially faster on the high-wall task. The faster high-wall completion is mainly because the robot uses a more dynamic wall-hop skill, whereas untrained adults generally cannot perform the same skill and need to use slower climbing strategies.

3) Ablation Failure Modes: To better understand how our design choices affect the proposed pipeline, we conduct a failure-mode analysis of the ablations, with results shown in Table VI. We can see that across 20 simulation trials, reduced data coverage primarily leads to stalling and incorrect skill selection under out-of-distribution entry states. This supports our hypothesis that dense data coverage is necessary for generalizing across diverse skill-entry conditions. In contrast, reducing the number of training environments or the model capacity leads to training failure, where the student policy fails to converge to correct and feasible parkour behaviors. This shows that scaling both RL training environments and policy capacity is important for convergence as the task distribution becomes more diverse and difficult. We also note that task success rates are not directly correlated with locomotion speed. In some tasks, higher approach speed can provide useful momentum and make climbing or vaulting easier.

Ablation	1.0 m/s, 76 cm			1.0 m/s, 94 cm			2.0 m/s, 58 cm			2.0 m/s, 76 cm		
	T	B	S	T	B	S	T	B	S	T	B	S
Extreme Distances	6	0	0	7	0	0	0	0	12	1	0	8
Half Density	10	0	0	7	0	0	0	0	3	3	0	0
1/4 training env	0	20	0	0	6	0	0	5	4	0	3	7
1/2 training env	0	1	0	0	0	0	0	1	2	0	4	3
3-MLP	0	20	0	0	20	0	0	5	0	0	2	0
4-MLP	0	0	0	0	19	0	0	0	3	0	2	1
Ours	0	0	0	0	0	0	0	1	1	0	0	2

TABLE VI: Failure modes over 20 simulation trials. T: incorrect timing or stalling; B: incorrect behavior; S: incorrect skill selection.

4) Velocity Tracking Baseline: To show the importance of human reference motion in our framework, we include a standard reward-shaping velocity-tracking baseline that learns locomotion purely from handcrafted rewards and a terrain curriculum, without any motion imitation or human reference trajectories. We follow IsaacLab's standard rough-terrain velocity-tracking recipe using its Unitree G1 rough-terrain configuration[1] , which is widely used for humanoid locomotion and terrain traversal and is largely consistent with state-of-the-art reward-shaping-based setups [13]. In alignment with prior works [53, 13], we also employ a terrain curriculum to ease learning. Specifically, we gradually increase terrain difficulty from 0.3 m to 1.0 m over 10 levels (with a 2 m run-up), providing a smooth progression of tasks that helps the policy bootstrap stable locomotion on easier terrain before tackling harder contact and balance challenges as the terrain becomes progressively harder. The curriculum advances the robot to a harder level once it achieves sufficient success on the current level, and moves it back to an easier level if its performance drops. Notably, different from our student policy which relies on an onboard depth camera for sensing, this baseline directly receives a local terrain height map (i.e., privileged height observations from simulation), which has been shown highly effective in prior parkour systems [13, 33].

5) AMP Baseline: Since AMP [31] is a popular algorithm for chaining skills with human reference data, we also implemented an AMP baseline by following the `MimicKit`[2] AMP implementation released by the original AMP authors. In our experiments, this baseline can walk stably and track the commanded velocity, but it does not perform well on obstacle traversal: it fails on most tasks, especially the harder ones, which is broadly consistent with prior reports that AMP can be difficult to extend to agile motions [50]. At the same time, we recognize that AMP performance can depend strongly on implementation details and careful tuning [38], and we did not have the bandwidth to fully explore this tuning space. For this reason, we do not include AMP in the formal comparison, and instead leave this result as a note for context.

[1]Code available at https://github.com/isaac-sim/IsaacLab/blob/main/source/isaaclab_tasks/isaaclab_tasks/manager_based/locomotion/velocity/config/g1/rough_env_cfg.py.

[2]Code available at https://github.com/xbpeng/MimicKit.

TABLE VII: Reward formulation using Gaussian-shaped tracking scores.

Reward Terms	Equation	Weight
Task (Tracking)		
Body Position	$\exp\Big(-\big(\frac{1}{\lvert\mathcal{B}_{\text{target}}\rvert}\sum_{b\in\mathcal{B}_{\text{target}}}\lVert\mathbf{p}_b^{\text{des}}-\mathbf{p}_b\rVert^2\big)/0.3^2\Big)$	1.0
Body Orientation	$\exp\Big(-\big(\frac{1}{\lvert\mathcal{B}_{\text{target}}\rvert}\sum_{b\in\mathcal{B}_{\text{target}}}\lVert\log(R_b^{\text{des}}R_b^{\top})\rVert^2\big)/0.4^2\Big)$	1.0
Body Linear velocity	$\exp\Big(-\big(\frac{1}{\lvert\mathcal{B}_{\text{target}}\rvert}\sum_{b\in\mathcal{B}_{\text{target}}}\lVert\mathbf{v}_b^{\text{des}}-\mathbf{v}_b\rVert^2\big)/1.0^2\Big)$	1.0
Body Angular velocity	$\exp\Big(-\big(\frac{1}{\lvert\mathcal{B}_{\text{target}}\rvert}\sum_{b\in\mathcal{B}_{\text{target}}}\lVert\boldsymbol{\omega}_b^{\text{des}}-\boldsymbol{\omega}_b\rVert^2\big)/3.14^2\Big)$	1.0
Anchor Position	$\exp\Big(-\lVert\mathbf{p}_{\text{anchor}}^{\text{des}}-\mathbf{p}_{\text{anchor}}\rVert^2/0.3^2\Big)$	1.0
Anchor Orientation	$\exp\Big(-\lVert\log(R_{\text{anchor}}^{\text{des}}R_{\text{anchor}}^{\top})\rVert^2/0.4^2\Big)$	1.0
Regularization		
Action smoothness	$\lVert\mathbf{a}_t-\mathbf{a}_{t-1}\rVert^2$	-0.1
Joint position limit	$\sum_{j=1}^{N}\big[\max(l_j-\theta_j,0)+\max(\theta_j-u_j,0)\big]$	-10.0
Undesired self-contacts	$\sum_{b\notin\mathcal{B}_{\text{ee}}}\mathbf{1}\big[\lVert f_b^{\text{self}}\rVert>1\,\text{N}\big]$	-0.5

TABLE VIII: Domain randomization parameters. ($\mathcal{U}[\cdot]$: uniform distribution)

Domain Randomization	Sampling Distribution
Physical parameters	
Static friction coefficients	$\mu_{\text{static}}\sim\mathcal{U}[0.4,\,1.3]$
Dynamic friction coefficients	$\mu_{\text{dynamic}}\sim\mathcal{U}[0.4,\,1.1]$
Restitution coefficient	$e_{\text{rest}}\sim\mathcal{U}[0,\,0.5]$
Default joint positions (except ankle) [rad]	$\Delta\theta_j^0\sim\mathcal{U}[-0.01,\,0.01]$
Default ankle joint positions [rad]	$\Delta\theta_j^0\sim\mathcal{U}[-0.03,\,0.03]$
Torso COM offset [m]	$\Delta x\sim\mathcal{U}[-0.025,0.025],\ \Delta y,\Delta z\sim\mathcal{U}[-0.05,0.05]$
Root velocity perturbations	
Root linear vel [m/s]	$v_x,v_y\sim\mathcal{U}[-0.1,0.1],\ v_z\sim\mathcal{U}[-0.05,0.05]$
Push duration [s]	$\Delta t\sim\mathcal{U}[1.0,\,3.0]$
Root angular vel [rad/s]	$\omega_x,\omega_y,\omega_z\sim\mathcal{U}[-0.1,0.1]$

TABLE IX: Training hyperparameters.

Hyperparameter	Motion Tracking	Distillation
Architecture		
Actor / Student MLP hidden dims	[512, 256, 128]	[2048, 1024, 512, 256, 128]
Critic MLP hidden dims	[512, 256, 128]	[512, 256, 128]
Activation function	ELU	ELU
Init noise std	1.0	0.01
Depth backbone	–	3-layer CNN + GAP
Depth input resolution	–	58×87
Depth output dim	–	32
Training		
Steps per environment	24	24
Max iterations	20,000	20,000
Learning rate	1×10^{-3}	3×10^{-4}
Schedule	adaptive	adaptive after 1000 iterations
Clip parameter	0.2	0.2
Entropy coefficient	0.005	0.001
Discount factor (γ)	0.99	0.99
GAE λ	0.95	0.95
Desired KL	0.01	0.01
Learning epochs	5	2
Mini-batches	4	96
Max grad norm	1.0	1.0
Distillation-Specific		
Curriculum end epoch	–	10,000
Distill loss type	–	mse
DAgger loss coefficient	–	10.0