Light-Loco-Parkour: Versatile Perceptive Whole-Body Locomotion via Multi-Skill Distillation

Hongming Chen, Zhuoran Li, Hongxi Wang, Jiangpeng Hu, Ziliang Li, Peize Liu, QingRui Zhao, Xuhao Liu, Liang Pan, Ximin Lyu, Yuntao Ma†, and Tingxiang Fan†
Light Origins

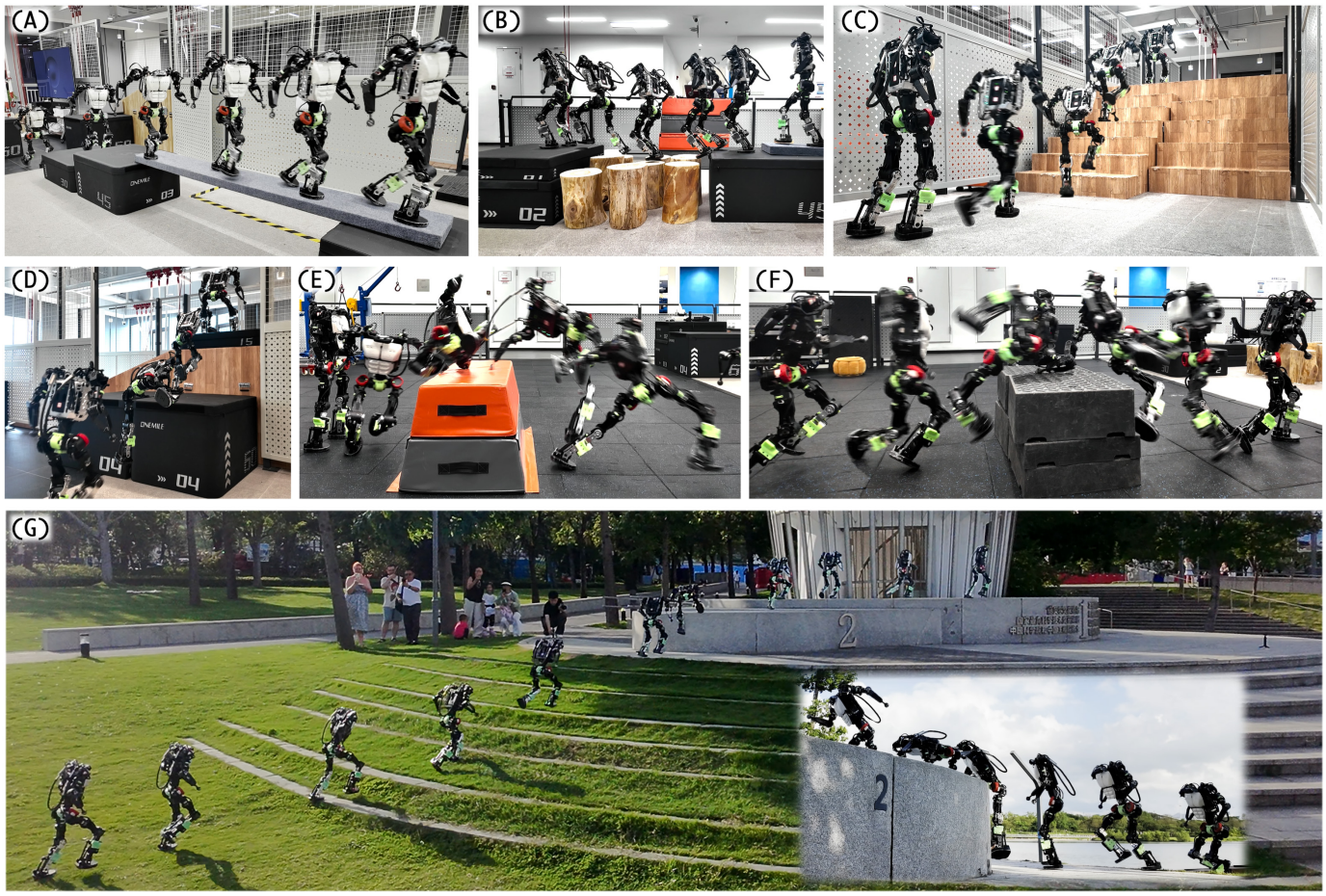


Fig. 1. **Light-Loco-Parkour** deployed on Lightbot 0, a custom-built 90 cm humanoid, and performs perceptive whole-body locomotion across indoor and outdoor terrains with onboard sensors and compute. **(A)** Crossing a narrow plank bridge, where precise foot placement is essential. **(B)** Traversing stepping stones over sparse footholds. **(C)** Climbing a staircase. **(D)** A double climb onto a tall platform beyond leg reach, bracing with the arms across two stages. **(E)** A speed-vault over a box, planting a hand and swinging the legs across. **(F)** A reverse-vault that clears a box back-first. **(G)** Outdoor experiments across natural terrain. Videos are available at **https://light-loco-parkour.github.io/**.

***Abstract*—Existing humanoid whole-body control systems still fall short of the way humans move through cluttered terrain: they either track expressive whole-body references without terrain generalization, or react to terrain online while leaving the arms, torso, and knees largely unused. We present `Light-Loco-Parkour` (LightLP), an end-to-end perceptive whole-body locomotion system that closes this gap with a single deployable policy. Conditioned only on onboard depth and a velocity command, the policy decides when to walk, balance, climb, step down, or vault, with no reference input, skill label, hand-coded gate, or runtime motion graph. Compared with prior humanoid systems, LightLP makes three contributions. First, it introduces a whole-body perceptive-control pipeline that extends an RL-trained, velocity-tracking locomotion policy with parkour skills learned from object-interacting motions, so the same policy tracks velocity in open terrain, executes whole-body traversal at obstacles, and resumes locomotion afterward. Second, it acquires terrain-conditioned skills from sparse seeds by expanding a single motion into dynamically feasible, terrain-paired references across obstacle geometry, rather than relying on a large motion corpus. Third, it learns autonomous skill transitions from reward, letting the policy decide when and which whole-body skill to invoke from depth and command alone, with no one-hot skill label, hand-coded state machine, or runtime motion generator. Simulation and real-world experiments show high success across both benchmarked terrains and unseen obstacle variations, and the same policy transfers zero-shot to indoor and outdoor hardware experiments. These results demonstrate autonomous perceptive whole-body locomotion on a humanoid in outdoor settings, using only onboard sensing and a single deployable policy.**

***Index Terms*—Humanoid and Bipedal Locomotion, Reinforcement Learning, Whole-Body Control, Perception-Action Coupling.**

†Light Origins robotics team Co-Lead

I. Introduction

HUMANOID robots are designed to operate in the cluttered, human-centric environments that wheeled and tracked machines cannot reach. Moving through such spaces is rarely a matter of the legs alone: traversing an obstacle often demands the *entire* body, with the hands vaulting, the knees bracing, and the torso leaning in continuous coordination with what the robot perceives. A humanoid that can sense the terrain ahead and recruit its whole body to negotiate it would unlock applications ranging from search-and-rescue over rubble, to inspection of cluttered industrial sites, to operation in the everyday human surroundings these machines are ultimately built for.

Recent learned controllers have demonstrated several humanoid capabilities: they can track whole-body reference motions [1, 2], walk on real hardware [3, 4], and support teleoperated whole-body tasks [5]. Most demonstrated capabilities, however, remain limited to flat or mildly uneven terrain. Traversal of rough, obstacle-rich environments remains less common, especially when success requires non-foot contacts. Progress toward this setting has mainly followed two learning paradigms: motion imitation and locomotion.

Mimic-based methods learn from a human-motion prior. A reference trajectory, retargeted from motion capture or video, is provided to the policy as part of its observation, and the policy is rewarded for tracking it [1, 6, 7]. This formulation has been used to reproduce dynamic whole-body skills, including cartwheels and backflips [1], and recent systems have scaled reference tracking to larger and more diverse motion sets [2, 8–10]. Its limitation is that the deployed behavior remains tied to the reference distribution. When the robot or environment deviates from the captured motion, the policy may enter out-of-distribution (OOD) states. Moreover, interaction data with objects and terrain is scarce, and without exteroception the policy cannot adapt its motion to scene geometry that differs from the demonstration [11].

Locomotion methods take a different approach. They optimize a policy directly with reinforcement learning to track a commanded planar velocity (v_x, v_y, ω_z). Without a reference trajectory, these policies can discover robust gaits for stairs, slopes, and unstructured terrain, and have transferred successfully to real robots [12–15]. However, reward-only learning has difficulty discovering contact-rich whole-body skills on humanoids. In practice, policies often converge to leg-dominant behaviors, while coordinated use of the arms, torso, and knees requires extensive reward shaping and does not scale easily across diverse parkour skills.

These complementary limitations motivate whole-body locomotion [16, 17], which combines reference-driven whole-body motion with terrain-conditioned perceptive control. Perception can condition a skill on the observed terrain, while whole-body contact extends locomotion beyond foot-only traversal. Existing approaches, however, still inherit important costs from reference-based learning, including offline motion generation and motion-capture datasets that scale poorly with the number of skills, contacts, and terrain variations.

Realizing whole-body locomotion that is at once capable, generalizable, and deployable raises three challenges. 1) The first is data scarcity. Whole-body parkour requires reference motions that are not only expressive, but also precisely paired with the terrain contacts that make the motion possible. Motion-capture clips must be retargeted across different embodiments and then aligned to objects, a process that easily produces floating contacts, scene penetration, or dynamically infeasible references; teleoperation cannot provide this data either, since current humanoids and teleoperation interfaces cannot reliably execute forceful whole-body parkour interactions in the first place. 2) The second challenge is learning a *generalizable* whole-body skill from such references. Most reference-based skills are demonstrated in a specific environment, with fixed object size, shape, and placement; the policy can succeed there by effectively replaying the trajectory, but it cannot adapt the contact timing, body posture, or force application when the terrain changes. Using references as a source of robust, terrain-conditioned skill learning, rather than as trajectories to memorize, is therefore itself a central difficulty. 3) Finally, reference-learned skills are usually short-horizon behaviors. Even if each individual skill works in isolation, a long-horizon task still requires the policy to decide from its own observations *which* skill to invoke, *when* to invoke it, and when to return to ordinary locomotion.

To address these challenges, we present `Light-Loco-Parkour`, an end-to-end whole-body locomotion system. On a humanoid robot it traverses demanding terrain, including stairs, narrow beams, boxes, and stepping stones, and performs agile parkour skills including climb-and-step, speed-vault, step-down, and reverse-vault (Fig. 1). At its foundation is a perceptive locomotion backbone, which we extend so that the same policy acquires a variety of whole-body skills rather than legged gaits alone. To overcome data scarcity, each skill begins from a single seed motion that is iteratively refined into a rich set of terrain-paired references covering different obstacle geometries, dynamically feasible by construction. To chain the skills together, LightLP learns a dedicated transition stage in which the handoffs emerge purely from RL reward, rather than being solved offline by trajectory optimization or a motion graph. The result is a single policy that, from onboard depth and a velocity command alone, infers and executes the most appropriate skill for the terrain ahead, without an explicit one-hot skill label.

The contributions of this paper are threefold.

- **Whole-body perceptive-control pipeline.** We introduce a whole-body perceptive-control pipeline that extends an RL-trained, velocity-tracking locomotion policy with parkour skills learned from object-interacting motions. Conditioned on onboard perception and a velocity command, the unified policy tracks velocity in open terrain, executes whole-body traversal at obstacles, and resumes locomotion afterward.
- **Terrain-conditioned skill acquisition from sparse seeds.** Terrain-conditioned skill acquisition grows a single seed motion into a continuum of dynamically feasible, terrain-paired references, for instance expanding one climb reference from a 45 cm obstacle to 75 cm,

rather than collecting a large motion corpus. Trained on this continuum, one policy per skill generalizes across obstacle geometry.

- **Autonomous skill transition from reward.** Reward-driven transition learning lets the decision of *when* to invoke *which* whole-body skill emerge from perception and task reward alone. A single policy switches between locomotion and skills purely from onboard depth and a velocity command, with no one-hot skill label, hand-coded state machine, or runtime motion generator.

II. Related Work

A. Locomotion

Classical locomotion is model-based, reducing the robot to a simplified template such as a linear inverted pendulum constrained by the zero-moment point, or a centroidal/rigid-body model solved by model-predictive control and trajectory optimization [18–22]. These models are precise but presume near-flat, coplanar foot contacts and hand-tuned gaits.

Deep reinforcement learning then recast locomotion as a data-driven problem. The recipe was proven first on quadrupeds. Hwangbo et al. [23] trained agile locomotion in simulation with an actuator network that closes the sim-to-real gap and transferred it to hardware, while RMA [24] added rapid online adaptation to changing dynamics. The same learning paradigm was then carried to bipeds and humanoids, from robust bipedal jumping [25] to real-world humanoid walking. Notably, Radosavovic et al. [3] trained a causal transformer over the history of proprioceptive observations with large-scale model-free RL and deployed it zero-shot. The resulting controller walks across varied outdoor terrain, stays robust to pushes, and adapts in context, all without a hand-designed model. More recent work even learns an internal denoising world model in place of direct exteroception [4]. These controllers, however, act from proprioception alone and remain blind, discovering terrain only on contact.

Perceptive locomotion closes this gap by adding an exteroceptive observation that lets the policy choose its motion before contact, and two implementations dominate. The first builds an explicit elevation map from onboard LiDAR or depth fused with odometry and plans footholds on it [26–28]; it is accurate when the map is, but under mapping noise and drift the robot can step into empty space on sparse footholds such as stepping stones or balance beams. The second is end-to-end, acting directly on raw depth: it sidesteps the mapping pipeline and its latency, at the price of a harder perception problem. This end-to-end recipe first delivered agile parkour on quadrupeds [12, 29], and was then carried to humanoids for perceptive locomotion over challenging terrain [30]. The two efforts most related to ours combine whole-body skills with perception [16, 17]. PHP [16] obtains its whole-body locomotion from reference motion that must be synthesized offline by hand, so the robot appears to be magnetically drawn toward any obstacle ahead. Moreover, it relies on a one-hot command in its observation, which leaves it unable to accurately track the velocity command. Zhang et al. [17] instead pair a diffusion-based motion generator with a tracking policy; because both policies must be run at inference, onboard computation becomes slow and an additional compute unit is required, and the approach still inherits the same scarcity of whole-body data, which leaves its motions stiff.

In contrast to these two systems, LightLP differs on three counts. First, it learns perceptive locomotion from reward alone, with no reference in the loop, so its motion stays natural and robust. Second, it selects and executes whole-body skills from onboard depth and a velocity command, with no one-hot index, so it follows the commanded velocity faithfully rather than being drawn onto whatever obstacle lies ahead. Third, it runs as a single policy onboard, without the second network and dedicated compute unit that a runtime motion generator demands.

B. Whole-Body Skills from Human Motion

A humanoid shares almost the same morphology as a human, and human motion is among the easiest behavioral data to collect at scale, from motion capture to everyday Internet video. This precondition, a near-identical body paired with abundant, cheap demonstrations, has spurred a large body of work that leverages human motion data to endow humanoids with whole-body skills. These methods fall into two categories, separable at the observation level by whether the policy is given an explicit reference. *Mimic-based* methods [1, 2] feed a per-frame reference into the observation and reward the policy for tracking it. *Style-based* methods [6, 31] omit the explicit reference, instead matching the distribution of human motion through an adversarial style reward or encoding the reference into a latent token that the policy conditions on.

Peng et al. [1] train a physics-simulated character to track reference motion clips with reinforcement learning under a joint imitation-and-task objective; their reference state initialization (RSI), which resets each episode to a random reference frame, is what makes highly dynamic skills learnable. Recent systems carry this tracking recipe to real humanoids by closing the sim-to-real gap from different angles. BeyondMimic [2] models the actuators from first principles and deploys through a low-latency stack. ASAP [32] learns a residual delta-action model from real-world rollouts to align simulation with the measured dynamics. KungfuBot [33] pairs physically constrained motion processing with an adaptive tracking curriculum. Together, these systems transfer agile skills such as cartwheels, kicks, and kung-fu zero-shot to a Unitree G1. Scaling this idea up, SONIC [9] trains a single tracking policy on hundreds of hours of motion-capture data with adaptive sampling, yielding a *general* controller that tracks a wide range of motions and generalizes to unseen ones.

The complementary, style-based approach instead matches the *distribution* of reference motion rather than any specific frame: AMP [6] does so through a learned discriminator, and SMP [31] casts the prior as a reusable, task-agnostic score-matching module. These methods are demonstrated mostly on flat ground, however, and pushing them to acquire whole-body skills tends to deform the motion into unnatural shapes [8].

What unites both categories is a shared blind spot: these skills make almost no contact with objects off the ground,

fundamentally because human references rarely capture a motion together with the scene that shaped it. The community has therefore pushed to obtain more diverse references. One intuitive route recovers SMPL motion from video, e.g. via GVHMR [34]; without known camera intrinsics, however, the recovered motion is systematically biased and produces artifacts such as scene penetration. NMR [35] observes that optimization-based retargeting is non-convex and prone to such artifacts, and instead casts retargeting as learning a data distribution. It uses clustered reinforcement-learning experts to project noisy human demonstrations onto the robot's feasible manifold, thereby eliminating joint jumps and self-penetration; occlusion, however, remains difficult for video. Motion capture offers another source: OmniRetarget [11] jointly optimizes the motion together with the manipulated objects and terrain, preserving interactions and generalizing across configurations, but because it optimizes purely at the reference level it cannot guarantee the dynamic feasibility of the resulting motion.

Once references are available, the remaining question is how to learn a skill that generalizes across obstacle geometry rather than overfitting a single demonstration, and the answer turns on the observation. Deep Whole-body Parkour [15] places the reference in the observation and appends a height scan, expecting the scan to let the policy absorb how the obstacle deviates from the reference; in practice this generalizes poorly and its real-robot performance is limited. HIL [8] instead keeps the reference out of the observation, encoding the terrain through a PointNet latent while still training against a reference-tracking reward; lacking any reference input, the policy converges only with difficulty, and each parkour skill remains tied to one specific terrain, demonstrated without generalization or real-hardware deployment.

Our approach tackles both stages. For data, we augment a single seed motion, obtained from video or motion capture, into references that are dynamically feasible by construction and exactly matched to the terrain that produced them, rather than kinematic-only or scene-free. For skill learning, we keep the explicit reference out of the deployable observation, as in HIL, so that each skill generalizes across obstacle geometry; but to overcome the slow, unstable convergence this otherwise incurs, we adopt a teacher–student architecture in which a per-skill expert supervises the student, greatly accelerating convergence.

III. System Overview

The overview of LightLP is shown in Fig. 2 and comprises four modules: perceptive locomotion learning (Section IV) and three that together build the whole-body skills, namely data augmentation (Section V-A), skill learning and generalization (Section V-B), and transition learning (Section V-C). A final distillation pass then ports the resulting height-scan policies onto the robot's onboard depth camera for deployment (Section VI).

A. Problem Formulation

We train the whole-body perceptive locomotion policy with reinforcement learning, formulating the problem as a partially observable Markov decision process. At each control step the policy $\pi_\theta(a_t \mid o_t)$ maps an observation o_t to an action a_t, and is optimized with PPO [36] to maximize the expected discounted return $\mathbb{E}[\sum_t \gamma^t r_t]$ under a reward r_t specified per module. All training is carried out in simulation in IsaacLab [37].

B. Observation and Action Spaces

During teacher training, the observation includes privileged information, the velocity command (v_x, v_y, ω_z), and a local height scan. The deployable student is then distilled from this height-scan teacher and uses onboard depth in place of the privileged scan. The action a_t sets target joint positions tracked by a PD controller, and the policy runs onboard at 50 Hz.

IV. Perceptive Locomotion Learning

A. Policy Architecture

The general policy architecture uses an MLP encoder as the core feature extractor for the map input. A proprioception encoder embeds the proprioceptive observations, and the map encoder embeds the map observations conditioned on this proprioception embedding. We then feed both the proprioception embedding and the map embedding through a multilayer perceptron (MLP) to output the action.

We instantiate this architecture within a teacher–student scheme [26, 38, 39]: a teacher trained on privileged sensing supervises a student that is deployable from onboard observation alone. The teacher reads a clean height scan. For its map encoder, we compared a convolutional design with a fully-connected design. The convolutional encoder yielded no measurable gain yet noticeably lengthened training, so we use an MLP encoder in all experiments. Since the teacher's network is entirely MLP-based with no recurrence, a single observation carries no memory of the recent past. We therefore stack the last five observation frames to supply temporal context, following input-history designs that let a policy implicitly infer the latent dynamics and contact state it cannot sense directly [40]. The student, by contrast, is built for onboard deployment, where the privileged quantities available in simulation, most notably the base velocity, can no longer be measured, and where the only exteroception is a single depth image that is noisier and far narrower in field of view than the height scan. We therefore encode the depth with an MLP, fuse it with the proprioception embedding, pass the result through an RNN, and decode its hidden state with a final MLP into the action. The recurrence builds memory across frames so that the policy can implicitly infer the unmeasured state, such as the current velocity $(v_x^c, v_y^c, \omega_z^c)$, and retain terrain that has scrolled out of view.

B. Asymmetric Actor and Critic

We train the teacher with an asymmetric actor–critic [41, 42], in which the critic accesses privileged information that the actor does not, yielding more accurate value estimates that help the teacher learn better. Granting privileged input to the

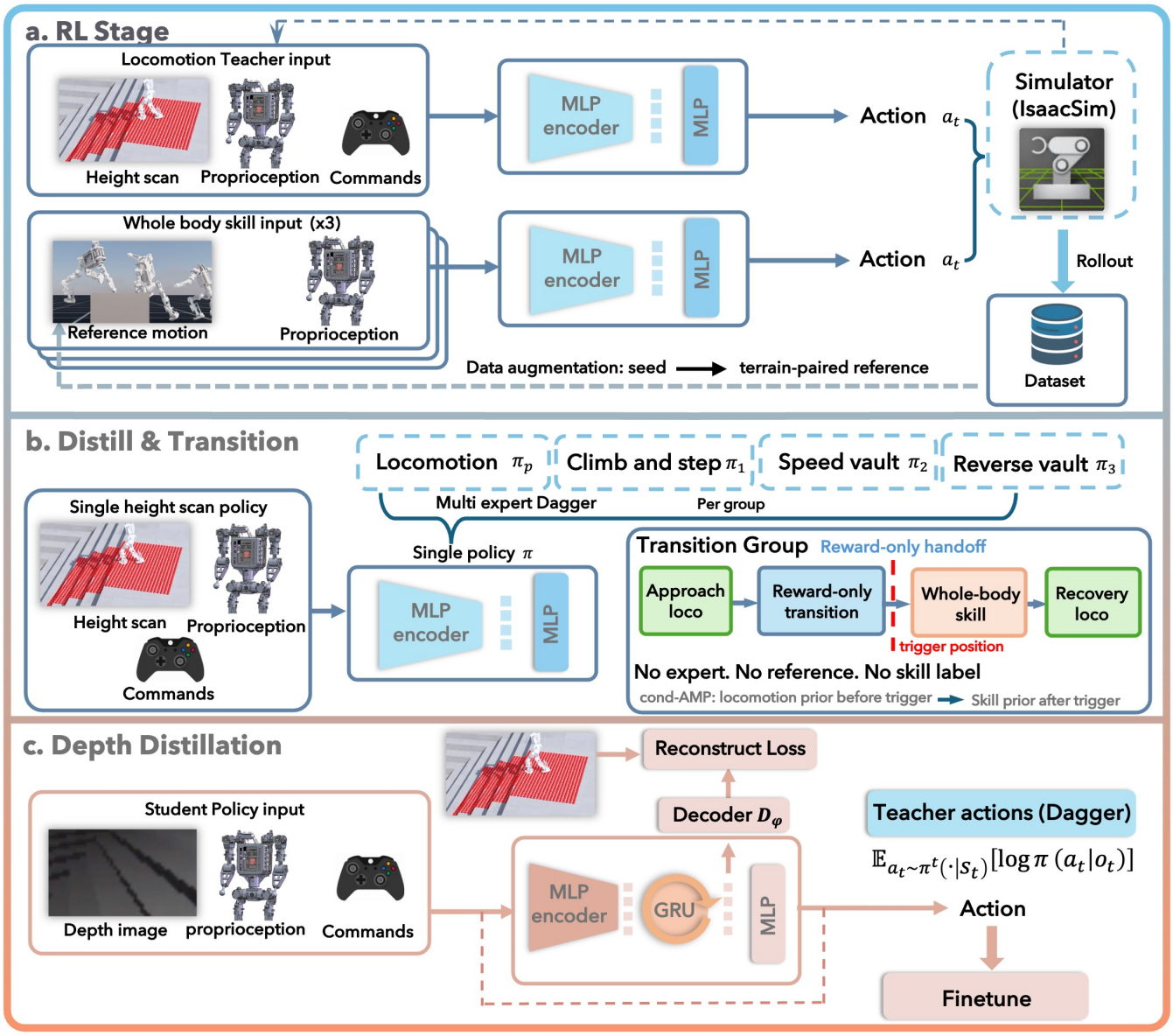


Fig. 2. **Overview of `Light-Loco-Parkour`. (a) RL stage:** using privileged information, we train one perceptive-locomotion teacher and one teacher for each whole-body skill, while data augmentation expands each seed motion into terrain-paired references. **(b) Distill & transition:** multi-expert DAgger distills the locomotion and skill teachers into a single height-scan policy; a reward-only transition group then fine-tunes the policy to hand off between locomotion and whole-body skills without an expert, reference, or skill label, while the AMP prior switches from locomotion to skill at the trigger position. **(c) Depth distillation:** the resulting height-scan policy is distilled into a recurrent depth policy using onboard-style depth, proprioception, and velocity commands, with teacher-action supervision, auxiliary height-scan reconstruction, and a final fine-tune.

teacher's *actor* would speed its learning and yield a stronger-performing teacher, yet it would leave the student, which never sees that input, unable to infer the teacher's observation at distillation and thus unable to reproduce its behavior. This information gap between an over-informed teacher and a partially observed student is well documented [43–45]. We therefore reason backward from the student when designing the teacher's observation, giving the actor only quantities the student can itself infer [46].

In our implementation, the actor reads a noise-free height scan together with proprioception subjected to mild domain randomization. The sparse, narrow terrains we target, such as stepping stones and balance beams, make precise foot placement essential, so we additionally grant the actor a privileged contact flag. The critic receives clean, noise-free inputs as well, augmented with oracle information such as under-foot height scans.

C. Environments

1) Rewards: As a locomotion controller, the policy has two essential objectives: to follow the velocity command and to avoid terminating. Our reward accordingly comprises two groups, one that rewards tracking of the commanded velocity and one that penalizes the behaviors leading to termination, both summarized in Table I.

The tracking group comprises four terms: linear-velocity tracking, angular-velocity tracking, an upright-orientation reward, and a velocity-slack bonus. Both velocity-tracking terms use an exponential kernel on the tracking error,

$$r_{\text{vel}} = \exp\!\left(-\|e\|^2/\sigma^2\right), \tag{1}$$

where e is either the planar-velocity error $(v_x^c, v_y^c) - (v_x, v_y)$ or the yaw-rate error $\omega_z^c - \omega_z$, and σ is the kernel width. The upright term mixes two kernels on the projected-gravity error g_{xy},

$$r_{\text{ori}} = \exp\!\left(-2\|g_{xy}\|^2\right) + 0.1\,\exp(-\|g_{xy}\|)\,. \tag{2}$$

On the hardest terrains, such as stepping stones and balance beams, a policy trained from scratch tends to halt in front of an obstacle to avoid falling. Under the exponential tracking reward of Eq. (1) alone, this timidity is never overcome and the robot never crosses. A common remedy adds a position-based or goal-reaching reward that pulls the robot forward [12, 13], but our system carries no odometry, which makes a global position reward inapplicable. We therefore introduce a velocity-slack bonus that rewards the robot whenever its forward speed lies within a band of the command,

$$r_{\text{slack}} = \mathbf{1}\!\left[\, v_x^c/v_x \in [0.3,\ 1.5] \,\right], \tag{3}$$

with v_x the commanded forward velocity and v_x^c the current forward velocity. Unlike the exact tracking above, this loose band lets the policy modulate its speed while negotiating an obstacle, slowing to climb or vault without being penalized, as long as it keeps advancing roughly at the commanded pace. For the opposite-direction penalty in Table I, $\mathbf{v}^c = (v_x^c, v_y^c)$ denotes the current planar velocity and $\hat{\mathbf{v}} = (v_x, v_y)/\|(v_x, v_y)\|$ denotes the commanded planar direction.

With command tracking in place, the second group of terms shapes how well the robot negotiates the terrain. Most are standard regularization terms, including action-rate smoothing, a joint-limit safety penalty, and an undesired-contact penalty, and are listed in Table I. The remaining two penalties, by contrast, are central to our setting. The illegal-footstep penalty discourages placing a stance foot over a hole or an edge, which is essential on sparse footholds such as stepping stones and on narrow balance beams. Rather than building an elevation map [26–28], we mount a small downward height scanner on each foot and read its rays directly. For each foot in contact, the penalty is the fraction of its rays whose hit point lies more than $\delta = 0.1$ m below the foot (Fig. 3),

$$r_{\text{foot}} = \sum_i c_i \,\frac{1}{K}\sum_{k=1}^{K} \mathbf{1}\!\left[\, z_i^{\text{foot}} - z_{ik}^{\text{hit}} > \delta \,\right], \tag{4}$$

where each contacted foot contributes zero at the center of a stone and grows toward one near a hole or edge, with c_i flagging foot i in contact and K rays per foot.

The foot-acceleration penalty, in turn, shapes how the robot lands. Its raw signal is the excess linear acceleration of the feet above $\bar{a} = 30$ m/s^2, but instead of acting per step it is accumulated through a first-order exponential-decay filter,

$$\tilde{e}_t = \alpha\,\tilde{e}_{t-1} + \sum_{i\in\text{feet}} \max(\|a_i\| - \bar{a},\, 0)\,, \qquad \alpha = e^{-\Delta t/\tau}, \tag{5}$$

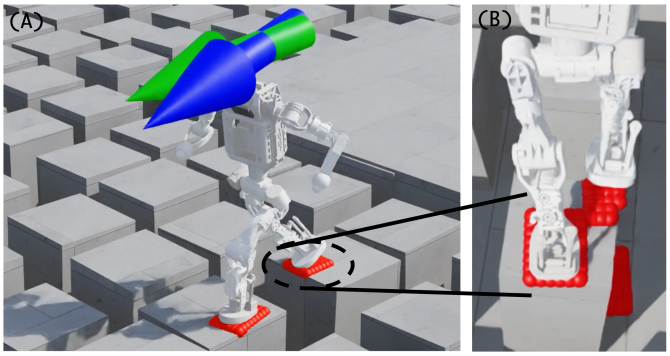


Fig. 3. Foot-mounted height scan to compute the illegal-footstep penalty. Each foot casts a small grid of downward rays, and a ray whose hit point lies more than δ below the foot signals that the foot overhangs a hole or an edge, as on the stepping stones shown.

with $\tau = 0.06$ s, and the term penalizes $\tilde{e}_t$. The decaying memory makes a hard impact felt over a short window rather than a single instant, which drives the policy to decelerate each foot before contact and to plan where and how firmly to land. This turns foothold selection into an anticipatory, perceptive behavior and yields softer, more deliberate steps.

2) Termination: We use the following termination terms, each with a distinct role.

- *Time-out.* The episode ends at the fixed horizon, marked as a time-out so the value function bootstraps from the final state rather than receiving a failure return.
- *Out of bounds.* The episode ends when the robot leaves its terrain patch, declared with a 2 m buffer before the true edge. It bootstraps rather than penalizes, signaling the end of usable terrain rather than a failure.
- *Joint-velocity guard.* A joint speed beyond 50 rad/s ends the episode, also treated as a time-out and as a numerical safeguard rather than a behavioral failure.
- *Torso contact.* A contact above 1 N on the torso resets the episode, penalizing the collisions and falls we want to avoid.
- *Excessive acceleration.* A base linear acceleration above 40 m/s^2, measured after a 1 s warm-up at reset, likewise resets the episode, catching violent impacts that the contact sensor alone may miss.
- *Fall-over.* When the base tilts past 63° from upright the episode terminates, but stochastically at probability 0.01 per step rather than instantly, leaving an expected hundred-step window in which the policy can attempt to recover its balance.

To improve robustness and gather more diverse experience, the two impact triggers, torso contact and excessive acceleration, share a degree of immunity: a random 10% of the environments, resampled every 200 steps, ignore both and continue through brief contacts or impacts rather than reset on the first frame. We expose this immunity flag to the critic (Section IV-B), so value estimation stays consistent with whether an environment can actually terminate on impact.

3) Terrain and Curriculum: We train on a single procedurally generated terrain that spans a broad range of obstacles,

TABLE I
REWARD TERMS FOR PERCEPTIVE LOCOMOTION TRAINING.

Term	Expression	Weight	Description
Tracking rewards			
Linear-velocity tracking	Eq. (1)	$+2.0$	Follow the commanded planar velocity.
Angular-velocity tracking	Eq. (1)	$+2.0$	Follow the commanded yaw rate.
Upright orientation	Eq. (2)	$+1.0$	Keep the torso upright.
Velocity-slack bonus	Eq. (3)	$+1.5$	Keep the forward speed near the command.
Penalties			
Undesired contact	$\sum_i \mathbf{1}[f_i > 1\,\mathrm{N}]$	-2.0	Avoid contact on links other than the feet.
Joint limit	$\sum_j \mathbf{1}[\,\text{near \& toward}\,]$	-10.0	Stay away from joint limits.
Illegal footstep	Eq. (4)	-1.0	Avoid stepping onto a hole or edge.
Heading error	$\lvert\Delta\psi\rvert$	-1.0	Follow the commanded heading.
Opposite direction	$\max(0, -\mathbf{v}^c \cdot \hat{\mathbf{v}})$	-1.0	Avoid moving against the command.
Action rate	$\lVert a_t - a_{t-1}\rVert^2$	-0.1	Encourage smooth actions.
Foot acceleration	Eq. (5)	-0.01	Land softly with anticipation.

arranged as a grid of 10 difficulty levels and 32 columns of terrain type (Fig. 4).

Among them, the sparse-foothold terrains, including stepping stones, plank bridges, and pillar fields, make precise foot placement essential, and they are where the illegal-footstep penalty and the foot-mounted height scan of Section IV-C1 are especially important.

Difficulty rises along the rows, and we move each robot through the levels with a terrain-level curriculum [47]. A robot that, under a velocity command, walks more than half a terrain cell while tracking that command well is promoted to a harder level. One whose tracking falls short is demoted to an easier one. Distance here is the episode-cumulative path length rather than straight-line displacement, so a command that resamples mid-episode does not corrupt the signal. A further 10% of resets are placed at a random level regardless of performance, which keeps every difficulty populated and prevents the early levels from emptying as the policy improves.

4) Events and Domain Randomization: The policy is trained entirely in simulation, so we randomize the quantities the simulator cannot match to the real robot and add disturbances the robot will meet in the field [48]. Table II lists the ranges, grouped by what they model. Ground-contact terms cover the friction and restitution of every body. Inertial terms perturb link masses, add an unmodeled torso payload, and shift the base center of mass. Actuation terms vary armature, joint friction, and the coupled stiffness–damping that stands in for motor-constant (K_t) variation. These last terms are resampled slowly, no more than once every 5×10^4 steps, so that each episode sees a temporally consistent actuator rather than gains that jump within a rollout.

We highlight two terms. We perturb the depth camera's mounting pose at every reset [49], since the student policy reads this camera at deployment and a fixed extrinsic calibration is never exactly right on hardware. We also push the base with a random velocity every few seconds, which forces the policy to recover its balance from disturbances rather than relying on an undisturbed gait. Finally, each episode begins from a randomized base pose and velocity and from perturbed joint positions, so the policy is exposed to a wide range of starting states rather than a single nominal stance.

TABLE II
DOMAIN RANDOMIZATION RANGES.

Parameter	Range
Ground contact (startup)	
Friction (static, dynamic)	$\times$ [0.2, 1.3]
Restitution	[0.0, 0.8]
Inertial properties (startup)	
Link mass	$\times$ [0.85, 1.15]
Torso payload	$\times$ [1.0, 1.2]
Base center of mass	$\pm$2.5 cm (x, y, z)
Actuation (slow resample)	
Joint armature	$\times$ [0.75, 1.25]
Coulomb friction	$\times$ [0.7, 1.3]
Viscous friction	$+$ [0.0, 0.05]
Stiffness–damping (K_t)	$\times$ [0.875, 1.075]
Perception (reset)	
Depth-camera position	$\pm$1 cm
Depth-camera orientation	$\pm$0.025 rad
Disturbance (interval)	
Push interval	3.7–4.2 s
Push velocity	$\pm$0.5 m/s
Initialization (reset)	
Base position, yaw	$\pm$0.5 m, $\pm\pi$
Base linear, angular vel.	$\pm$0.5
Joint position, velocity	$\pm$0.1 rad, $\pm$1.0 rad/s

V. WHOLE-BODY SKILLS

We build whole-body parkour skills that recruit not only the legs but also the arms and upper body to perform maneuvers such as climb-and-step, speed-vault, and reverse-vault, expanding the range of terrain the robot can reach.

A. Data Augmentation

To address the scarcity of whole-body references with environmental contact, we build a seed dataset from short Internet videos of a human performing each target skill. For each video, we extract the per-frame SMPL body model with GVHMR [34] and retarget it with GMR [50]. We then manually place a virtual obstacle at the contact points implied by the retargeted hands and feet, yielding a coarse motion-and-terrain seed pair. However, neither GVHMR nor GMR is aware of the obstacle, so the seed reference inevitably collides with it: the supporting hand sinks into the top face, and a

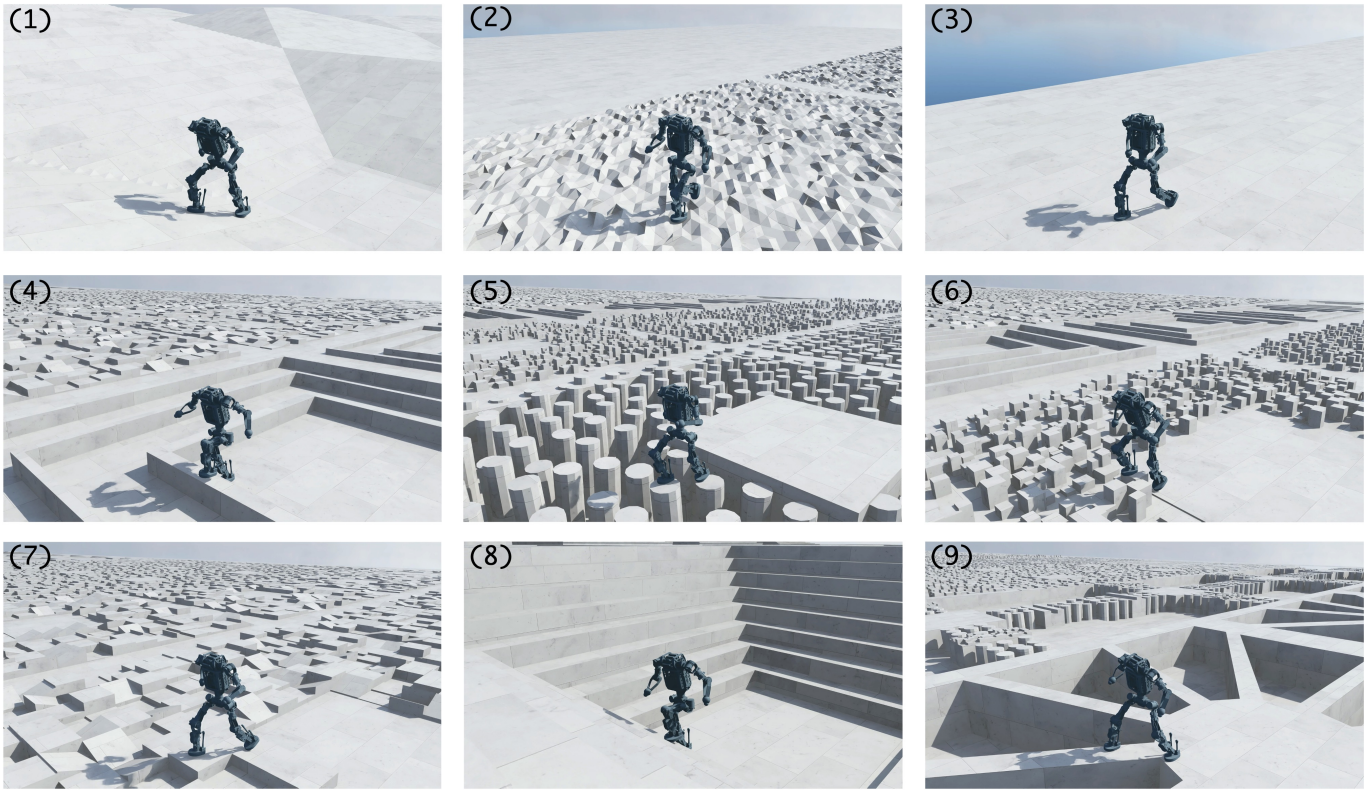


Fig. 4. The locomotion training terrain types. Each column is a distinct terrain type and difficulty increases along the rows: (1) slopes, (2) Perlin-noise rough ground, (3) flat ground, (4) rails, (5) raised pillars, (6) stepping stones, (7) box fields, (8) stairs, and (9) plank bridges.

knee clips through the front edge. The simulator, by contrast, is itself an effective retargeting resource: a reference tracked under physics comes out both in seamless contact with the obstacle and dynamically feasible [51, 52].

1) Object-Interaction Mimic: Our mimic stage builds on BeyondMimic [2], which we extend from free-space imitation to contact-rich, object-interacting skills. Since a parkour skill is defined by where the body meets the obstacle rather than by the pose alone, we track the global root position instead of only the root-relative pose, anchoring every contact, whether from the hand, foot, or pelvis, to a fixed location in the scene. We also add two privileged observations, the distance to the obstacle and its size, so the policy can tell how far the true surface lies from an imperfect seed and correct toward it even when that seed still penetrates the obstacle. We then adapt the reward: for the body part meant to make contact, a global-frame position tracking term insists the contact point actually reach the obstacle surface [33], turning a nominal contact into an enforced physical one. Reference State Initialization [1], which resets each episode to a random reference frame, is essential here: it spreads exploration across the whole motion and exposes the policy to the high-momentum mid-skill states, such as the apex of a vault and the push-off of a climb, without which these dynamic skills fail to emerge. In the end, this yields a reference that matches the obstacle exactly (Fig. 5).

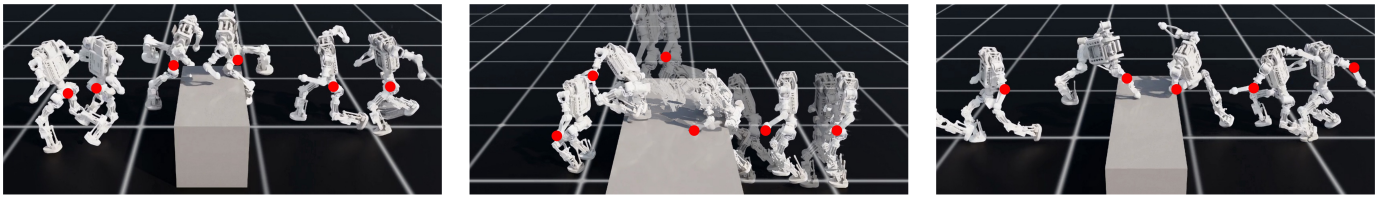
2) Iterative Self-Augmentation: Once the policy reliably traverses the seed obstacle, we roll out its trajectory in the simulator and adopt it as the next-iteration reference. Because the rollout is produced by the policy under the physics engine and the robot's actuator limits, it is dynamically and kinematically feasible by construction, even though its kinematic shape gradually drifts from the original SMPL trajectory across iterations. At each iteration, we lift both the obstacle and the motion by 5–10 cm so that the next round extracts a slightly harder version of the same skill. Formally, at iteration i the loop produces the next motion-and-terrain pair via

$$\begin{aligned} \pi_i &= \arg\min_{\pi} \mathcal{L}_{\text{mimic}}(\pi;\, r_i, d_i), \\ \hat{r}_i^{(k)} &\sim \text{Rollout}(\pi_i,\, d_i), \quad k \in [K], \\ \hat{r}_i^{\star} &= \arg\max_{k\in[K]} \text{Score}\big(\hat{r}_i^{(k)},\, r_i\big), \\ (r_{i+1},\, d_{i+1}) &= \big(\hat{r}_i^{\star} + m\,\mathbf{e}_z,\;\; d_i + m\,\mathbf{e}_z\big), \end{aligned} \tag{6}$$

where $\mathbf{e}_z$ is the gravity-axis unit vector, $m \in [5, 10]$ cm is the per-iteration lift, and $\mathcal{L}_{\text{mimic}}$ is a DeepMimic-style tracking loss optimized by the MIMICTRAIN step of Algorithm 1. We store each rollout reference together with its paired terrain, and the accumulated pairs form the augmented dataset used for skill learning.

B. Skill Learning and Generalization

1) Height-Scan Distillation: Although the Object-Interaction Mimic stage (Section V-A1) yields a feasible reference for each skill, the resulting experts are inherently neither general nor deployable. The mimic expert is fed the obstacle geometry directly, and, lacking odometry, it has no



(a) reverse-vault (b) climb and step (c) speed vault

Fig. 5. Red markers indicate the body where the global body position reward is applied: the wrist for climb-and-step, the elbow for speed-vault, and the pelvis for reverse-vault.

Algorithm 1 Iterative Reference Refinement

Definition: r_i: reference motion at iteration i; d_i: paired obstacle; π_i: policy trained at iteration i; m: per-iteration height increment; K: number of parallel rollouts; N: number of iterations.
Input: seed pair (r_0, d_0) recovered from an Internet video via GVHMR, with d_0 manually aligned to the human contacts in r_0.
$\mathcal{R} \leftarrow \{(r_0, d_0)\}$
for $i = 0, 1, \ldots, N-1$ **do**
  $\pi_i \leftarrow \textsc{MimicTrain}(r_i, d_i)$
  $\{\hat{r}_i^{(1)}, \ldots, \hat{r}_i^{(K)}\} \leftarrow \textsc{Rollout}(\pi_i, d_i, K)$
  $\hat{r}_i^{\star} \leftarrow \arg\max_k \ \textsc{Score}(\hat{r}_i^{(k)}, r_i)$
  $r_{i+1} \leftarrow \textsc{Lift}(\hat{r}_i^{\star}, m), \quad d_{i+1} \leftarrow \textsc{Lift}(d_i, m)$
  $\mathcal{R} \leftarrow \mathcal{R} \cup \{(r_{i+1}, d_{i+1})\}$
end for
Return: $\mathcal{R} = \{(r_0, d_0), \ldots, (r_N, d_N)\}$

way to anchor itself to a global reference frame. Each expert therefore reproduces a single motion tied to one obstacle, rather than a behavior that transfers across geometry or runs from onboard sensing alone. What we want instead is a whole-body skill that runs on the same observation as our perceptive locomotion policy: proprioception, a local height scan, and the velocity command. Exteroception and the velocity command then serve as a conditional input that tells the policy which task to perform at each moment, without any explicit skill label.

From the continuum of references produced by data augmentation we pick, for each skill, the iteration whose obstacle is the most demanding: furthest, tallest, and closest to the joint torque and angular-velocity limits of the deployment hardware. We distill this hardest expert directly onto the perceptive locomotion observation. The hardest expert is the most demanding case for distillation from exteroception alone, so using it lets the distilled policy inherit the full operating envelope rather than only the easy interior of the curriculum.

We distill each chosen expert with DAgger [53], collecting on-policy rollouts from the student and supervising them with the expert's actions at the matching states. Distilling on the student's own sampled states alone, however, yields a brittle policy: once a rollout drifts off the expert's distribution, imitation offers no signal for recovering, an instability also reported in HIL [8]. We therefore open a PPO loss alongside the imitation term, so the policy retains the ability to recover even when its tracking is imperfect. The final objective thus combines expert distillation with PPO,

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{DAgger}} + \lambda_{\text{RL}}\, \mathcal{L}_{\text{PPO}}(r), \qquad (7)$$
$$r = w_{\text{task}}\, r_{\text{task}} + w_{\text{goal}}\, r_{\text{goal}} + w_{\text{mimic}}\, r_{\text{mimic}},$$

where the imitation term

$$\mathcal{L}_{\text{DAgger}}(\pi_\theta) \;=\; \mathbb{E}_{s\sim d^{\pi_\theta}}\Big[\, \big\| \pi_E(s_{\text{priv}}) - \pi_\theta(s^{\text{stu}}) \big\|_2^2 \,\Big] \qquad (8)$$

penalizes the gap between the expert action under the privileged observation s_{priv} and the student action under the deployable observation s^{stu}, evaluated on the student-visited states d^{π_θ}, $\mathcal{L}_{\text{PPO}}$ is the standard clipped surrogate driven by the composite reward r, and λ_{RL} trades off the two.

TABLE III
REWARD AND TERMINATION TERMS INHERITED FROM THE OBJECT-INTERACTION MIMIC STAGE (SECTION V-A1).

Term	Expression
Goal reward r_{goal}	$-\lVert p_T - p^\star \rVert_2 - \alpha \lVert \theta_T - \theta^\star \rVert$
Mimic reward r_{mimic}	$\exp\big(-\beta \lVert q_t - \hat{q}_t \rVert_2^2\big)$
Reference-deviation termination	$\lVert p_t - \hat{p}_t \rVert_2 > 25$ cm

The reward, domain randomization, and terminations are inherited from the Object-Interaction Mimic stage (Section V-A1), and only the observation changes to the height scan and velocity command. Table III gives the three terms: a goal reward on the terminal base pose (p_T, θ_T) relative to its target $(p^\star, \theta^\star)$, a mimic reward on the joint angles q_t against the reference $\hat{q}_t$, and an early termination once the base p_t drifts from the reference $\hat{p}_t$.

2) Skill Generalization: After distillation, the policy π_i reproduces its skill only on the single obstacle its expert was trained on. The goal of this stage is to turn that single-configuration policy into one that handles the whole range of obstacles collected during data augmentation. We continue training π_i across that range, but now without an expert to imitate. Dropping the expert, and asking for generalization rather than reproduction, drives the changes below.

Terrain. Each skill trains on its own terrain, whose single obstacle is paired one-to-one with the corresponding motion reference recovered during data augmentation. We randomize the obstacle's placement in the ground plane and arrange the configurations into a curriculum of difficulty levels, so the

policy sees the full range of placements rather than the single one its expert was trained on.

Reward. The mimic reward now tracks the motion paired with the currently sampled obstacle, so as the curriculum varies the terrain the imitation target follows the matching reference rather than a single fixed motion. On its own, though, kinematic imitation no longer suffices once the expert supervision is gone: the policy can satisfy the mimic term yet still drift away from actually crossing the obstacle. We therefore add a task reward that pulls the robot toward a target $p^\star$ placed a fixed offset beyond the obstacle,

$$r_{\text{task}} = \exp\big(-\|p^\star - p_t\|_2/\sigma_c\big), \tag{9}$$

where p_t is the robot's anchor position. The target is defined relative to the obstacle, which the height scan perceives, so this reward needs no odometry. The continuous proximity term keeps the robot committed to crossing rather than stalling in front of the obstacle. We pair it with a terminal reward of the same form that switches on only once the reference motion has finished playing,

$$r_{\text{term}} = \exp\big(-\|p^\star - p_t\|_2/\sigma_t\big)\,\mathbf{1}[\text{done}], \qquad \sigma_t < \sigma_c, \tag{10}$$

where $\mathbf{1}[\text{done}]$ marks the end of the motion playback rather than a requirement to reach the target. Its sharper kernel then rewards being close to the target once the motion is over.

Initialization. Since the obstacle now changes from episode to episode, we disable Reference State Initialization and instead reset the robot at the first frame of the motion, just before the obstacle. We keep that first-frame base pose but draw the joint angles from the perceptive locomotion policy rather than from the reference. Starting each skill from a locomotion-consistent posture makes the policy more robust and eases the later handoff between locomotion and skills during transition training (Section V-C).

C. Transition Learning

At this point we have a perceptive locomotion policy π_p (Section IV) and a set of whole-body skills $\{\pi_i\}$, each trained in isolation on its own obstacle. What the system still lacks is any way to choose among them. Since every skill was trained separately, the locomotion policy cannot invoke them: confronted with an obstacle, it falls back on locomotion alone and tries to walk through, which on the harder terrains it usually cannot. The skills, for their part, know how to traverse an obstacle but not when to begin or end one, as the references behind them cover only the traversal and not the approach or exit. Closing this gap is the goal of the transition stage: deciding when to call a skill and when to keep walking, without a hand-coded state machine.

We close this gap in two steps, following the multi-expert distillation and RL fine-tuning of Parkour in the Wild [13]: we first merge the policies into one, then fine-tune that single policy to switch between them.

1) Multi-Expert Distillation: We build one combined environment partitioned into groups and assign a subset of robots to each: **(a)** a *locomotion group* carrying the same terrain used to train π_p (Section IV), and **(b)** one *skill group* per skill, carrying that skill's terrain (Section V-B). A single student is supervised by the matching expert, either π_p or the corresponding π_i, in each group through DAgger alone, with no PPO term, exactly as in the height-scan distillation above. The student is never given a skill label: to act correctly it must read the terrain from its exteroception and reproduce the behavior of whichever expert owns that group. The critic, by contrast, does receive a one-hot indicator of the current group, so that value estimation can account for the differing rewards across groups while the actor stays label-free and deployable. Unlike quadruped parkour [13], where the experts differ mainly in where the feet land, our skills are whole-body, so the student must also reproduce the arm and torso motion that drives each skill against the obstacle, not merely the foothold pattern.

2) Transition Fine-Tuning: Distillation yields a policy that executes each behavior on its own terrain, yet it still cannot *transition*. Nothing in the combined environment ever asks it to leave one behavior and enter another within a single rollout. We therefore fine-tune the distilled policy with RL, adding a third group to the two above. **(c)** *Transition group*: cluttered scenes that interleave the skill obstacles with the open locomotion terrain, with no per-frame reference motion. A hand-shaped reward that explicitly tells the robot when to vault, climb, or walk fails here because the right choice is highly state-dependent, and any miscalibration makes the policy commit to a skill too early or refuse to engage at all. We instead drop the mimic reward, since there is no reference to mimic across a transition, and give group (c) its own reward,

$$\begin{aligned} r_{\text{trans}} = {} & w_v r_v + w_\omega r_{\omega_z} + w_{\text{pos}} r_{\text{pos}} \\ & + w_{\text{prox}} r_{\text{prox}} + w_{\text{amp}} r_{\text{amp}}, \end{aligned} \tag{11}$$

made of four kinds of term. A *task* reward tracks the commanded linear velocity r_v and yaw rate r_{ω_z}. A *sparse position* reward r_{pos} pays out only once the robot reaches a target placed *behind* the obstacle, marking a completed crossing. A *dense proximity* reward r_{prox} grows as the robot closes on that target, supplying the gradient that the sparse term lacks. Finally, a *conditional* adversarial motion-prior reward [6],

$$r_{\text{amp}}(s_t, s_{t+1}) \;=\; -\log\big(1 - D_\phi(s_t, s_{t+1})\big), \tag{12}$$

keeps the motion close to the learned behaviors, with the discriminator D_ϕ switched by phase: before the robot reaches the skill's trigger position the prior is the *locomotion* AMP, and once it passes that position the prior switches to the *skill* AMP. This phase switch is what actively drives the handoff: the robot is pushed to walk up to the obstacle and then, at the trigger, to adopt the skill that carries it across (Fig. 6). Unlike Parkour in the Wild, which feeds a goal into the observation and lets transitions emerge as the robot drives toward it [13], our policy observes only a velocity command, so it is this conditional prior, rather than a goal, that elicits the switch.

Because the position reward is sparse and paid only behind the obstacle, a policy that has never reached the far side receives no useful learning signal to guide it there. We therefore reset robots not only in front of the obstacle but also behind it, so that some episodes begin already past the crossing and immediately collect the terminal position reward, seeding

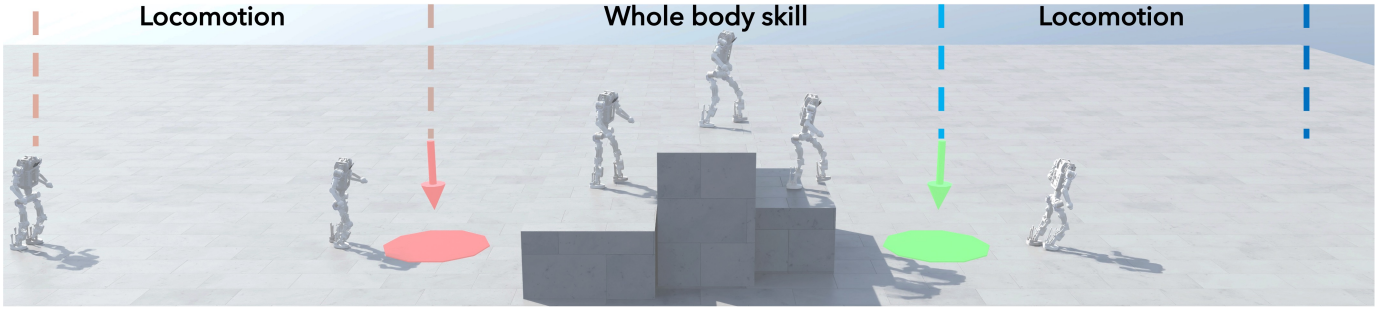


Fig. 6. Transition group. The robot is reset across all three regions, namely the locomotion terrain, the whole-body-skill zone, and the far side of the obstacle, so every phase of a crossing is trained. The conditional motion prior switches from the locomotion AMP to the skill AMP once the robot passes the trigger position.

the exploration that the front-reset episodes must otherwise discover on their own (Fig. 6).

Throughout the fine-tune, groups (a) and (b) act as *anchors*: their dense, well-shaped rewards keep the policy from drifting away from either competent locomotion or any of the learned skills, while group (c) is where the transition behavior is acquired under the sparse reward of Eq. (11). The policy is never told which skill to call: when the exteroception reports an obstacle that matches a known skill, its behavior gravitates toward the corresponding π_i, as that is the only way to satisfy both r_{amp} and r_{pos}. When the obstacle matches no skill, the policy falls back to π_p and either bypasses the obstacle or slows down. The resulting policy switches between locomotion and skills smoothly, without explicit gating, and recovers gracefully when an attempted skill aborts mid-execution.

VI. From Height Scan to Onboard Depth

Up to this point every policy operates on a height scan, which is convenient in simulation: it offers wide lateral coverage and is free of self-occlusion. The corresponding deployment-time observation, however, is a single chest-mounted depth image with a limited field of view, and a substantial fraction of the obstacle frequently leaves the camera frustum mid-skill, for instance when the torso pitches forward over a vault and the camera ends up looking at the sky or at the obstacle's edge. To bridge this gap, we run a final distillation pass from each height-scan policy onto a recurrent depth-conditioned policy under the same DAgger + PPO objective used to learn the skills. The recurrent stage feeds depth features into a GRU, so the policy can carry a short-term memory of terrain that has just fallen out of view rather than acting only on the current frame.

To make that memory encode terrain explicitly, we add an auxiliary reconstruction loss: a small decoder maps the GRU hidden state h_t back to the privileged height scan s_t^{scan} that the teacher observed but the student does not,

$$\mathcal{L}_{\text{recon}} = \left\| D_\psi(h_t) - s_t^{\text{scan}} \right\|_2^2, \tag{13}$$

where D_ψ is a lightweight reconstruction decoder with a bottleneck latent representation. Used as an auxiliary training loss, $\mathcal{L}_{\text{recon}}$ forces the recurrent state to accumulate a height-scan-like representation of the surrounding terrain from the stream of partial, self-occluded depth frames. The decoder is used only at training time and is dropped at deployment.

Distilling onto clean simulated depth would transfer poorly, as the hardware RealSense D435 returns far noisier and slower images. We therefore render the simulated depth through a noise-and-latency model calibrated to the physical camera. Each frame is corrupted by range-dependent Gaussian noise, with standard deviation $0.005 + 0.02\,d$ for a measured depth d (in meters), and a global depth-scale jitter of $\pm 5\%$. On top of this we apply persistent block dropout: small patches, refreshed every few frames, are set to the camera's maximum range to mimic the invalid regions the D435 produces, biased toward the top and bottom rows where they concentrate on the real sensor. We also replay the camera's measured timing, with a 30–60 ms processing latency at a 27–33 Hz frame rate, so the policy learns to act on stale, low-rate depth rather than the instantaneous frames available in simulation. Finally, a short fine-tune under this noise-and-latency model yields the policy we deploy on hardware.

VII. Experimental Results

A. Experimental Platform: Lightbot 0

All experiments are conducted on **Lightbot 0**, a custom-built compact humanoid developed at Light Origins (Fig. 7).

The platform stands 90 cm tall, weighs 18.9 kg, and exposes 21 actuated degrees of freedom distributed across the legs, torso, and arms, with parallel four-bar ankle linkages on both legs. Joint actuation is supplied by quasi-direct-drive motors with two torque tiers, 45 N·m for the waist and lower-body joints and 15 N·m for the arm joints, and a uniform peak angular velocity of 9.42 rad/s. These limits are substantially below those of the 1.3 m-class platforms used in concurrent humanoid parkour work [15–17]; in particular, the limited actuation is the reason the data-and-curriculum pipeline of Section V-A cannot rely on direct mocap retargeting from full-size humans.

The onboard sensor stack consists of a chest-mounted Intel RealSense D435 depth camera, tilted 30° downward to keep the terrain just ahead of the feet in view, together with a 6-axis IMU at the pelvis and joint encoders. All policies reported in this paper run from this onboard stack alone, with no LiDAR, motion capture, or external state estimation. The

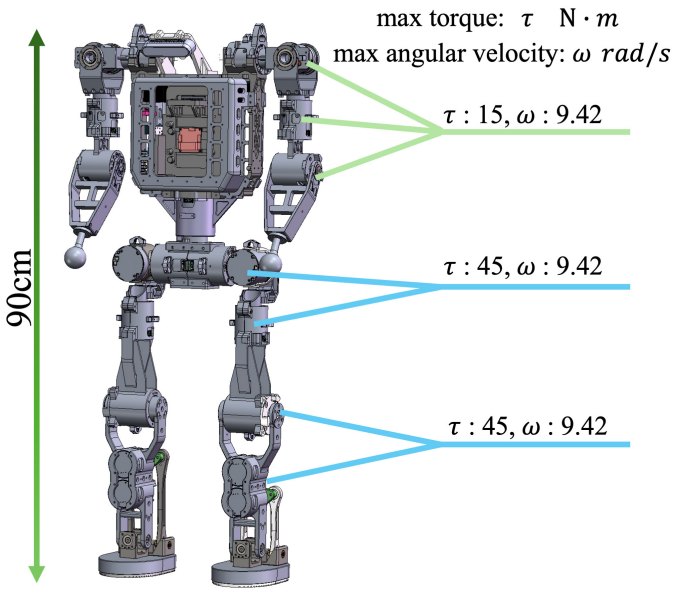


Fig. 7. **Lightbot 0**, the custom-built 90 cm humanoid used in all our experiments, with 21 actuated degrees of freedom, a chest-mounted depth camera, and a pelvis IMU.

deployed policy executes at 50 Hz on an onboard NVIDIA Jetson Orin Nano, a compact 7–25 W edge module rated at 67 INT8 TOPS, which is a far lighter compute budget than the dedicated units required by methods that run a motion generator at inference [17].

B. Real-World Experiments

We conduct real-world experiments, indoors and outdoors, to evaluate LightLP under deployment conditions. The policy tracks a velocity command from onboard depth and proprioception alone. At deployment, it needs no odometry, no motion-capture data, and no external mapping. Across these experiments it completes traversals across both structured indoor obstacles and unstructured outdoor terrain. These deployments demonstrate fully autonomous perceptive whole-body locomotion outdoors on a humanoid, with the policy switching between locomotion and whole-body contact-rich skills from onboard sensing alone. Fig. 8 shows the deployments, and we introduce each capability in turn.

reverse-vault. The robot executes a dynamic parkour maneuver by jumping toward an obstacle, performing a 360° aerial rotation, and using pelvis contact with the obstacle to redirect its motion before landing. The accumulated angular momentum during landing is then regulated through a second 360° rotation, allowing the robot to dissipate impact energy and achieve a stable final pose. This skill requires a rapid takeoff impulse, accurate obstacle-relative state estimation under temporary visual occlusion, and coordinated whole-body control for contact timing, momentum transfer, and impact absorption.

speed-vault. The speed-vault skill is a high-speed whole-body parkour maneuver that requires rapid momentum generation, transient single-arm support, and whole-body stabilization during flight and landing. Starting from a running approach, the robot reaches a peak velocity of 3.14 m/s, plants one hand on the obstacle, redirects its forward momentum, and enters a ballistic aerial phase. After hand release, the robot regulates its body orientation under inertial loading to reach a landing configuration. The maneuver requires coordination between unilateral arm support and bilateral leg swing, while the forward kinetic energy before takeoff and the landing impulse must be absorbed without destabilizing the robot.

Climb-and-step. The robot establishes chest-height hand contacts with the obstacle and uses bilateral arm support to lift its body beyond the limits of leg-only actuation. It then extends its legs, transitions through a kneeling posture on the platform, and recovers to a standing pose before jumping down. This maneuver challenges the robot to overcome lower-limb torque limitations through sustained upper-body loading and precise whole-body coordination across multiple contact phases. Furthermore, the drop from the elevated platform requires impact recovery under limited control authority during the aerial phase, so the robot must dissipate the vertical landing impulse at touchdown and regain a stable posture. We further replace the box with a 50 cm pommel-horse-like trapezoidal obstacle never seen in training, and the same skill still succeeds on the sloped, non-box profile, showing that it keys on the perceived terrain geometry rather than memorizing a fixed obstacle shape.

Plank bridge. The robot traverses a narrow plank bridge while maintaining both feet on a highly constrained support surface, where even small lateral deviations can result in a loss of contact. This task requires precise foot placement, perception-aware locomotion, and dynamic balance control under narrow lateral margins. Unlike walking on flat terrain, the robot must continuously regulate its body motion and foot trajectories because little recovery space remains after a lateral deviation.

Stepping stones. The robot traverses discrete stepping stones with gaps in between, where successful locomotion requires selecting and reaching valid footholds rather than relying on continuous terrain support. Since the onboard camera provides limited visibility of the foot-placement region, the robot must maintain a memory of previously observed terrain geometry and use this spatial representation to guide future footsteps. This maneuver evaluates perception-driven foothold selection and memory-based locomotion under partial observability.

High platform. The robot mounts a 30 cm high platform, representing an extreme step height of approximately 33% of its standing height. To complete the maneuver, the robot must sustain a prolonged single-leg stance while lifting the swing leg and placing it onto the elevated surface with a large range of motion. The task presents a significant balance challenge due to the highly asymmetric posture and limited support during the stepping phase, requiring precise foot placement and robust whole-body stabilization.

Outdoor curbed stairs. The robot ascends and descends real outdoor curbed stairs and traverses natural terrains such as grass slopes in a zero-shot setting. Unlike regular stairs with predictable footholds, curbed stairs require accurate perception of step edges and proactive foot-trajectory adjustment; insufficient foot clearance can cause the swing leg to catch on the

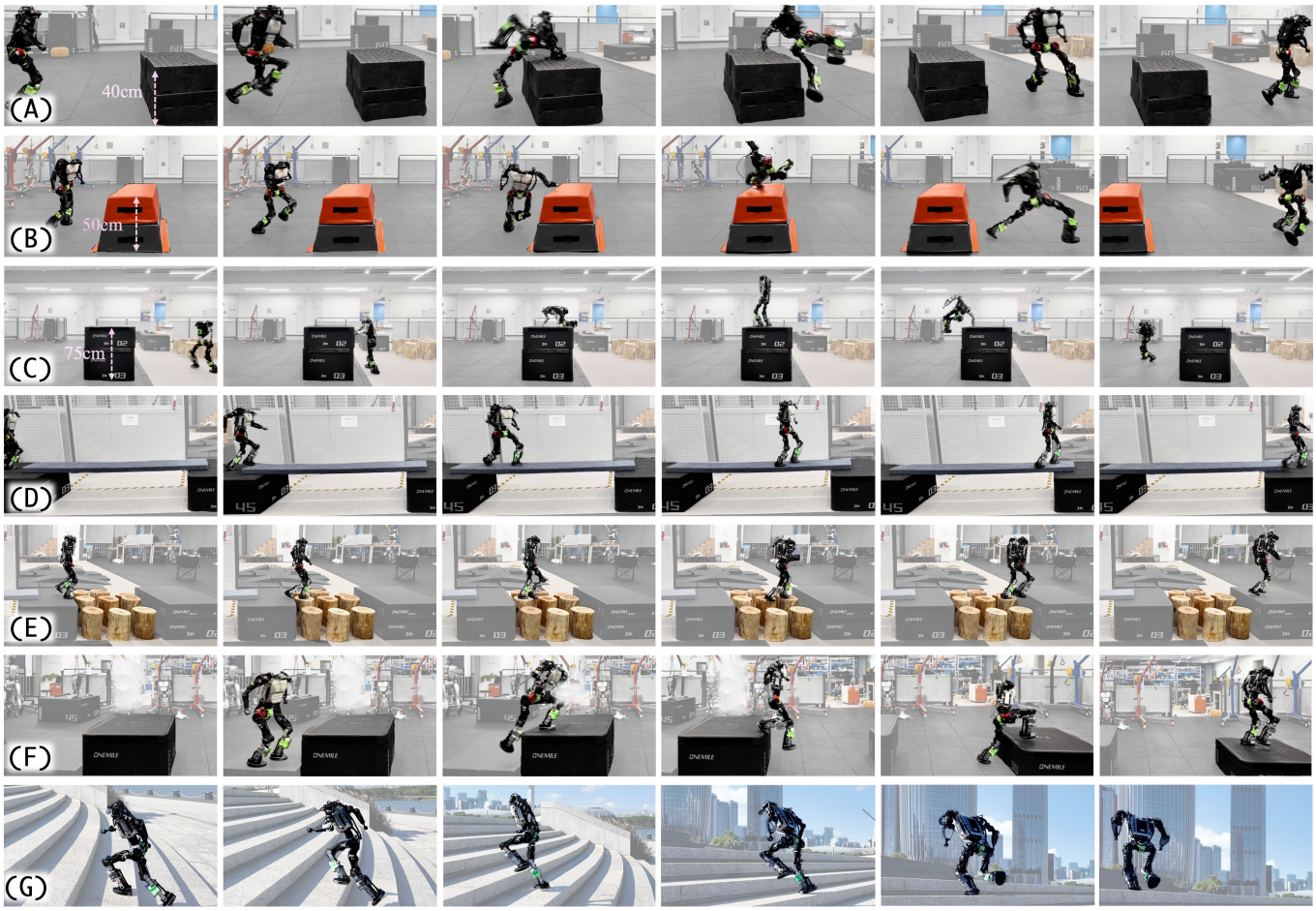


Fig. 8. **Real-world deployment on Lightbot 0**, each run zero-shot from onboard depth and proprioception. Rows follow the order of Section VII-B: reverse-vault, speed-vault, climb-and-step (including the unseen pommel horse), plank bridge, stepping stones, a high platform, and outdoor curbed stairs. The upper rows are whole-body skills, the lower rows the perceptive locomotion that carries the robot between them.

curb and result in failure. This experiment demonstrates the robustness of perceptive locomotion in real-world environments, where the policy must handle depth noise, terrain uncertainty, and previously unseen geometric variations.

Autonomous transitions. Finally, the policy chains locomotion and skills into continuous courses, deciding on its own and with no external cue when to switch (Fig. 9). This setting integrates the preceding capabilities, and we analyze it in detail in Section VII-D.

C. Simulation and Benchmark

We now turn to simulation, where we benchmark against prior work and ablate each component under controlled, repeatable conditions. We first evaluate the whole-body skills, namely reference generation and skill learning, and then the perceptive locomotion.

1) Reference Generation: Learning a *smooth* whole-body skill hinges on obtaining a good reference. Without one, the robot adopts strange poses while interacting with the object, because an obstacle-misaligned reference collides with the scene. Starting from the same seed motion, we compare our references against the two widely-used retargeting methods, GMR [50] and OmniRetarget [11], with the results reported in Table IV.

No penetration: the body must not pass through the obstacle. A direct GMR retarget ignores the obstacle entirely, so the hands and knees penetrate its surface. *No floating*: the body must actually touch the obstacle rather than hover above it. On climb-and-step, OmniRetarget avoids penetration only by keeping the body just off the surface, so the hands and feet never make contact. *Object-paired*: the motion is captured together with the obstacle it interacts with. *Dynamically feasible*: the motion is consistent with the robot's dynamics and actuation limits, with no teleporting jumps or other physically implausible transitions. As Table IV shows, only our references satisfy all four requirements at once: no penetration, no floating, object-paired, and dynamically feasible. OmniRetarget sometimes reports a larger nominal range, especially for vaulting, but this is a kinematic overestimate rather than an executable skill. On speed-vault the failure is sharper: to keep the body clear of the box, the retarget snaps it across in a discontinuous jump and teleports through the obstacle instead of clearing it. No torque-limited humanoid can execute such a reference, so it is unusable for policy learning. On climb-and-step, the retargeted motion instead hovers without load-

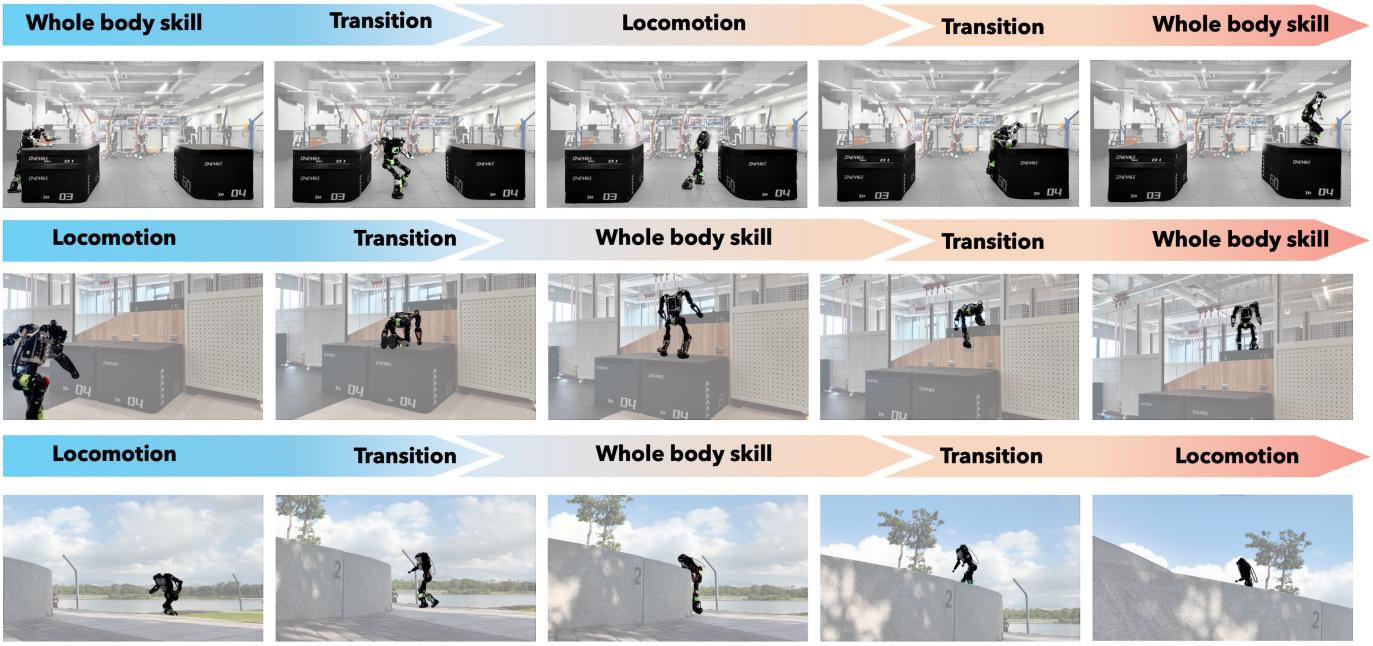


Fig. 9. **Real-world transition courses.** The policy switches between locomotion and whole-body skills on its own, with no external cue. Top: indoors, a climb-and-step followed by a staircase. Bottom: outdoors, crossing a grass field, ascending a staircase, and climbing a higher obstacle.

bearing contact, again leaving no continuous behavior to learn. The contrast with climb-and-step is instructive, since that skill depends directly on precise hand support and therefore exposes the missing contact detail.

TABLE IV
COMPARISON OF WHOLE-BODY REFERENCE-GENERATION METHODS.

Method	Range	Paired	No pen.	No float.	Feasible
Climb-and-step					
GMR [50]	single	✗	✗	✗	✗
OmniRetarget [11]	50–70 cm	✓	✓	✗	✗
Ours	50–75 cm	✓	✓	✓	✓
speed-vault					
GMR	single	✗	✗	✗	✗
OmniRetarget	40–55 cm	✗	✗	✗	✗
Ours	40–50 cm	✓	✓	✓	✓
reverse-vault					
GMR	single	✗	✗	✗	✗
OmniRetarget	40–55 cm	✓	✓	✗	✗
Ours	40–50 cm	✓	✓	✓	✓

2) Ablation Study: Table V reports a set of ablation experiments across all skills and perceptive-locomotion terrains. We evaluate *Ours*, *Ours w/o FT*, *Ours w/o GRU*, *Teacher*, *Teacher w/o ED*, and *Teacher w/o RA*. The student-side ablations isolate the final fine-tune and recurrent memory, while *Teacher w/o ED* tests whether the expert whole-body skill can be learned without expert distillation and *Teacher w/o RA* tests the effect of reference augmentation. The simulation evaluation covers both whole-body parkour skills and perceptive locomotion, with representative scenes shown in Fig. 10. In the whole-body-skill rows, each policy is tested across obstacle heights; within each height, the box width and depth are

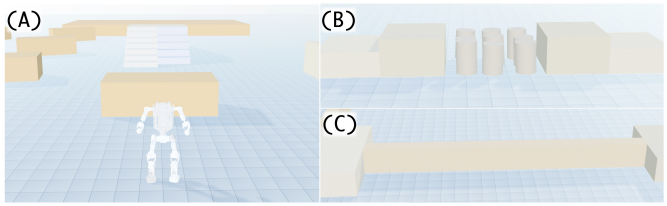


Fig. 10. Representative simulation evaluation scenes for whole-body parkour and perceptive locomotion, including obstacle skills, high stairs, stepping stones, and a balance beam.

randomized as $x \in [0.7, 1.0]$ m and $y \in [0.5, 1.5]$ m. Each reported ablation success rate in Table V is computed over 500 randomized trials under the corresponding setting. To keep the comparison controlled, every variant is trained independently under the same environment and a 15,000-iteration budget, so that differences in success rate reflect the removed component rather than training compute. The depth policy is distilled for 14,000 iterations and then fine-tuned for a further 1,000; the *w/o fine-tune* variant stops after distillation.

The most pronounced ablation effect comes from the final fine-tune. Distillation from the expert already gives the depth policy a usable version of each behavior, but the policy still suffers from residual errors after replacing the teacher's height scan with onboard depth. This drop is especially clear on foothold-sensitive tasks such as stepping stones, where small perception or timing errors directly lead to missed contacts. The final RL fine-tune largely closes this gap, bringing the depth policy back toward teacher-level performance by optimizing task success under the student's own observations.

We also compare feed-forward MLP and recurrent GRU architectures. The non-recurrent policy suffers a cliff-like drop in success, because the teacher observes privileged informa-

TABLE V
ABLATION AND BENCHMARK EXPERIMENTS.

Configuration	**Ours**	**Ours** w/o FT	**Ours** w/o GRU	Teacher	Teacher w/o ED	Teacher w/o RA	PHP [16]	MGMT [17]	BeamDojo [27]	CReF [54]
Whole-body skills: success rate (%)										
climb-and-step 60 cm / $0.66H$	99.2	88.4	54.0	99.9	✗	99.9	95	96.2	/	/
climb-and-step 65 cm / $0.72H$	98.8	89.8	56.6	99.2	✗	✗	/	/	/	/
climb-and-step 70 cm / $0.77H$	90.0	76.4	34.2	99.2	✗	✗	/	/	/	/
climb-and-step 75 cm / $0.83H$	33.4	17.0	0.0	98.6	✗	✗	/	/	/	/
reverse-vault 40 cm / $0.44H$	99.6	74.4	26.4	99.9	✗	99.9	/	/	/	/
reverse-vault 45 cm / $0.50H$	99.2	75.4	26.8	99.9	✗	✗	/	/	/	/
reverse-vault 50 cm / $0.55H$	96.8	72.8	25.4	99.4	✗	✗	/	/	/	/
speed-vault 40 cm / $0.44H$	99.0	65.2	23.4	99.9	✗	91	99	/	/	/
speed-vault 45 cm / $0.50H$	95.0	68.4	22.6	99.8	✗	✗	/	/	/	/
speed-vault 50 cm / $0.55H$	93.4	61.8	23.2	99.6	✗	✗	/	/	/	/
Perceptive locomotion: success rate (%)										
Stepping stones	99.9	34.6	0	99.9	/	/	/	/	91.7	/
Balance beam	99.9	99.9	53.8	99.9	/	/	/	/	94.3	/
Stairs (low)	99.9	99.9	47.6	99.9	/	/	/	99.9	/	/
Stairs (middle)	99.9	89.4	19.0	99.9	/	/	99.9	99.9	/	97.9
Stairs (high)	83.4	80.8	12.4	83.0	/	/	/	98.7	/	/
Stones (+height)	95.6	24.4	0	98.6	/	/	/	/	/	/
Beam (+yaw)	99.9	99.9	42.2	99.9	/	/	/	/	/	/

Skill rows use randomized boxes with $x \in [0.7, 1.0]$ m and $y \in [0.5, 1.5]$ m. / indicates settings not reported or outside the method's scope; ✗indicates failure to converge.

tion such as the body's local linear velocity whereas the deployable student does not. A memory-based network can partially recover this missing state in its latent representation by integrating recent proprioception and depth observations; we observed the same conclusion when replacing the GRU with an LSTM. Comparing *Ours* with the *Teacher* also shows the limit of memory: it mitigates the missing velocity and height-scan information, but cannot fully replace them. For example, in the 75 cm climb-and-step setting, the camera motion makes the obstacle height difficult to infer reliably during the approach, leading to a sharp drop in success.

The teacher-side ablations ask two related questions. *Teacher w/o ED* tests whether an expert whole-body skill can be learned directly from sparse reward without an expert-distillation stage, while *Teacher w/o RA* tests whether a single reference trajectory can provide generalization across obstacle variations. Together, they point to the difficulty of learning high-dynamic whole-body skills from sparse reward and a single seed motion alone. Following the HIL-style setting and removing expert distillation, the teacher fails to converge even though it still has privileged observations. The reason is not merely observation quality: to learn whether and how to traverse an obstacle, the robot must discover a sequence of forceful contacts with the object, and the exploration space becomes too large for reward-only learning. The policy often collides with the obstacle and settles into a poor local minimum instead of finding the coordinated contact sequence needed for vaulting or climbing. Using only a single reference without augmentation also overfits the teacher to that one trajectory and obstacle height. It may succeed at the seed configuration, but fails to cover the range in Table V; even on speed-vault, the success drops, suggesting that multiple terrain-paired references provide not only height coverage but also a broader exploration space for refining contact timing and placement.

TABLE VI
SUCCESS RATE (%) ON A TRAINED BOX VS. AN UNSEEN POMMEL-HORSE (TRAPEZOIDAL) OBSTACLE.

	reverse-vault		speed-vault	
Method	Box	Horse	Box	Horse
Ours (Student)	99.9	93.4	99.9	95.3

Beyond scale, we test generalization to obstacle *shape*. For reverse-vault and speed-vault we replace the training box with a pommel-horse-like trapezoidal obstacle never seen in training. Table VI shows that both skills transfer to the new shape, confirming that they key on the terrain the robot perceives rather than on a fixed obstacle profile.

3) Benchmark: We evaluate our student policy against state-of-the-art methods and, on the same terrains, ablate the components that produce it. We compare with four published humanoid controllers as policy-level configurations: motion generation plus motion tracking (MGMT) [17], PHP [16], BeamDojo [27] for sparse footholds, and CReF [54] for stairs. These methods cover complementary terrains: MGMT and PHP focus on reference-motion whole-body locomotion,

BeamDojo on sparse footholds, and CReF on stairs. Our student policy is evaluated across all of them, including whole-body skills such as climb-and-step and speed-vault, stepping stones with varying height, and balance beams with yawed approaches. In Table V, "/" indicates that the method does not report that experiment.

Normalized obstacle height. To compare robots of different sizes, we report each obstacle height as a ratio h/H, where h is the obstacle height and H is the robot's standing height. This ratio gives a simple scale-normalized measure of how large the obstacle is relative to the robot. Under this metric, PHP reports whole-body obstacle traversal mainly around $0.45H$ and $0.60H$, and does not demonstrate the higher normalized heights tested here. MGMT similarly reports whole-body climbing around $0.60H$. By contrast, our student policy is evaluated up to $0.83H$ for climb-and-step and shows the same advantage on speed-vault, while maintaining higher success at the overlapping settings. This indicates stronger generalization across obstacle scales and pushes the behavior closer to the robot's physical limits. We attribute this to the high-precision, dynamically feasible references produced by our method; combined with the final task-reward fine-tune, these references allow the student policy to retain high success even at larger normalized obstacle heights.

Coverage across terrains. Beyond whole-body skills, Table V also compares perceptive locomotion settings. BeamDojo is designed specifically for sparse footholds such as stepping stones and balance beams, yet its reported success remains lower than ours on the overlapping settings. For stairs, we divide the task into low (15 cm), middle (25 cm), and high (35 cm) levels; CReF reaches only the middle level and still reports lower success than our student policy. We attribute this broader terrain coverage to the reward design used during training and to the final fine-tune, which optimizes task success under the deployable depth observations. We further test stepping stones with height variation and balance beams with yawed approaches, where the student policy still maintains high success. Taken together, these results show that the proposed pipeline realizes a broad set of perceptive locomotion and whole-body parkour skills within a single deployable policy. This moves humanoid whole-body control beyond isolated terrain-specific behaviors toward a unified controller that can coordinate stepping, balancing, climbing, and vaulting within one learned policy.

D. Transitions

Fig. 9 demonstrates the transition courses in the real world, covering skill-to-locomotion, skill-to-skill, and locomotion-to-skill switches. These three settings jointly verify that the transition mechanism is effective across the full set of behavior changes required by the proposed pipeline. The decision is made from the policy's depth observation together with the velocity command: in the transition group, the policy effectively learns the composition of skills and selects, at each moment, which behavior is needed to follow the command through the current terrain.

TABLE VII
TRANSITION ABLATION AND COMMAND-ADHERENCE TEST: TRANSITION-AND-SKILL SUCCESS RATE (%).

Setting	Result
Transition-group ablation	
Plane locomotion + skills, w/o transition group	0
Rough perceptive locomotion + skills, w/o transition group	33
w/o AMP	51
Ours (with transition group)	98
Command adherence near an obstacle	
Reverse-away command in front of obstacle	True

1) Ablation Study: We count a trial as successful only when the policy both switches to the behavior required by the terrain and completes the resulting skill or locomotion segment. Evaluated over 100 randomized trials per setting, Table VII shows that the transition group is essential for this end-to-end behavior. Without it, a policy trained only on flat locomotion and isolated whole-body skills never learns how to connect them, and the success rate remains 0%. Even when rough-terrain perceptive locomotion is added, removing the transition group leaves the policy without a learned composition strategy: in front of an obstacle, it often continues the locomotion behavior, collides with the wall, or repeatedly steps in place instead of invoking the required whole-body skill. This raises success only to 33%. With the transition group, the policy identifies from depth and the velocity command which skill is needed in the current scene and completes the obstacle traversal, raising success to 98%. To examine how this switch is represented internally, we visualize the RNN hidden states with t-SNE in Fig. 11. With the transition group, the hidden state can leave the perceptive-locomotion region and enter the parkour-skill region when the terrain requires a skill. Without the transition group, the hidden state remains in the perceptive-locomotion region, explaining why the policy keeps walking or stepping in place instead of executing the required transition.

We next ablate the adversarial motion prior (AMP), which is needed not only for switching but also for deployable motion quality. Without AMP, the policy can still react to terrain cues, but the unconstrained objective produces unnatural motions and awkward contacts that are difficult to transfer directly to hardware. With AMP, the policy remains close to the learned skill manifold, yielding both a reliable transition and a physically plausible whole-body motion (Table VII).

We further probe whether the policy truly acts on the command rather than being drawn to any obstacle in view. This failure can arise in reference-bound whole-body policies [16]: because both locomotion and obstacle traversal are learned from a limited set of reference motions, the policy may not become robust to velocity commands that contradict the reference, and an observed obstacle can trigger forward traversal even when the command asks the robot to retreat. Standing in front of an obstacle, we therefore command the robot to reverse away from it. Our policy retreats as commanded (Table VII), confirming that its behavior is decided jointly by the velocity command and the depth observation rather than by the mere presence of an obstacle.

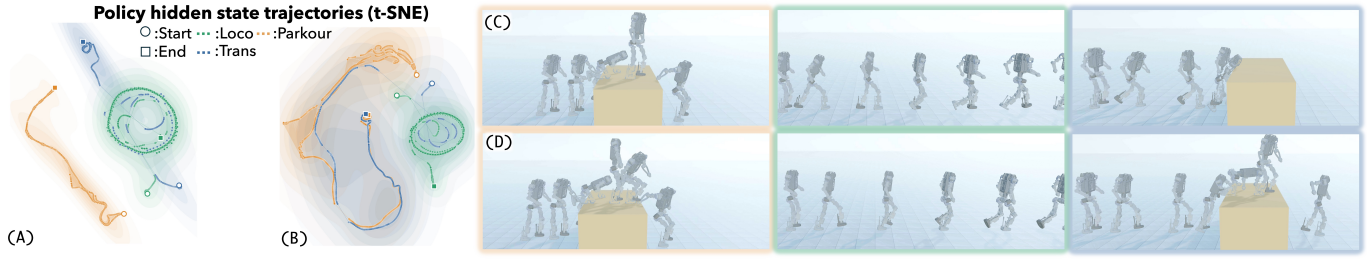


Fig. 11. t-SNE visualization of the RNN hidden states during transition courses, colored by behavior type: flat locomotion, transition, and parkour; rollout snapshots are paired with the corresponding embeddings. **(A,C)** Without the transition group, the hidden state stays confined to the perceptive-locomotion region even when the terrain ahead demands a skill, so the policy keeps walking or steps in place and never enters skill mode. **(B,D)** With the transition group, the hidden state leaves the locomotion region and moves into the parkour-skill region as the terrain requires, while the transition states form a bridge between the two clusters. The paired contrast shows that the recurrent policy learns to encode which mode to enter from the observed terrain and velocity command alone.

VIII. Conclusion

We presented `Light-Loco-Parkour`, an end-to-end perceptive whole-body locomotion system for a humanoid robot. The central result is a single deployable policy that reads onboard depth and a velocity command, then selects and executes the appropriate behavior across both perceptive locomotion and whole-body parkour skills, without a reference input, skill label, hand-coded gate, or runtime motion graph. This capability is enabled by three components: high-precision reference–object pairs that align whole-body motion with the terrain contacts required for each skill, RL fine-tuning that turns each skill into a terrain-conditioned behavior, and transition-group training that lets switching between locomotion and skills emerge from sparse reward. Deployed zero-shot from simulation onto Lightbot 0, the policy traverses climb-and-step, reverse-vault, and speed-vault obstacles in both indoor and outdoor environments, showing that whole-body contact and perceptive locomotion can be unified within one learned controller.

Several limitations are worth stating plainly. The seed-collection step of our data pipeline still requires a human-in-the-loop to align each video with a candidate obstacle, and we expect that this manual step is a real ceiling on how fast the skill set can be expanded. The transition policy currently couples a small, discrete set of skills, and we observe degraded behavior when obstacles overlap or appear in close succession, a problem the current sparse-reward setup does not fully resolve. Finally, the fixed chest-mounted depth camera limits both whole-body manipulation of the scene and the perceptive coverage available to the locomotion controller, particularly when the upper body occludes the camera mid-skill.

These limitations suggest the directions we are most interested in next. We plan to scale the pipeline to much larger reference sets and study whether richer transition behavior emerges as data grows, rather than having to be hand-shaped. We also believe the perceptual bottleneck can be addressed through active perception (camera or head gaze controlled by the policy itself) or by giving the policy an explicit short-term memory of the scene, so that mid-skill occlusion does not erase what the robot already saw on approach.

Acknowledgment

The authors would like to thank colleagues at Light Origins for helpful discussions and hardware support.

References

[1] X. B. Peng, P. Abbeel, S. Levine, and M. Van de Panne, "Deepmimic: Example-guided deep reinforcement learning of physics-based character skills," *ACM Transactions On Graphics (TOG)*, vol. 37, no. 4, pp. 1–14, 2018.

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

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

[4] X. Gu, Y.-J. Wang, X. Zhu, C. Shi, Y. Guo, Y. Liu, and J. Chen, "Advancing humanoid locomotion: Mastering challenging terrains with denoising world model learning," *arXiv preprint arXiv:2408.14472*, 2024.

[5] K. Darvish, L. Penco, J. Ramos, R. Cisneros, J. Pratt, E. Yoshida, S. Ivaldi, and D. Pucci, "Teleoperation of humanoid robots: A survey," *IEEE Transactions on Robotics*, vol. 39, no. 3, pp. 1706–1727, 2023.

[6] X. B. Peng, Z. Ma, P. Abbeel, S. Levine, and A. Kanazawa, "Amp: Adversarial motion priors for stylized physics-based character control," *ACM Transactions on Graphics (TOG)*, vol. 40, no. 4, pp. 1–20, 2021.

[7] G. B. Margolis, M. Wang, N. Fey, and P. Agrawal, "Softmimic: Learning compliant whole-body control from examples," *arXiv preprint arXiv:2510.17792*, 2025.

[8] J. Wang, Y. Jiang, H. Zhang, C. Tessler, D. Rempe, J. Hodgins, and X. B. Peng, "Hil: Hybrid imitation learning of diverse parkour skills from videos," *arXiv preprint arXiv:2505.12619*, 2025.

[9] Z. Luo, Y. Yuan, T. Wang, C. Li, S. Chen, F. Castañeda *et al.*, "Sonic: Supersizing motion tracking for natural humanoid whole-body control," *arXiv preprint arXiv:2511.07820*, 2025.

[10] Z. Luo, J. Cao, A. Winkler, K. Kitani, and W. Xu, "Perpetual humanoid control for real-time simulated avatars," *arXiv preprint arXiv:2305.06456*, 2023.

[11] L. Yang, X. Huang, Z. Wu, A. Kanazawa, P. Abbeel, C. Sferrazza, C. K. Liu, R. Duan, and G. Shi, "Omniretarget: Interaction-preserving data generation for humanoid whole-body loco-manipulation and scene interaction," *arXiv preprint arXiv:2509.26633*, 2025.

[12] Z. Zhuang, Z. Fu, J. Wang, C. Atkeson, S. Schwertfeger, C. Finn, and H. Zhao, "Robot parkour learning," *arXiv preprint arXiv:2309.05665*, 2023.

[13] N. Rudin, J. He, J. Aurand, and M. 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.

[14] H. Kim, H. Oh, J. Park, Y. Kim, D. Youm, M. Jung, M. Lee, and J. Hwangbo, "High-speed control and navigation for quadrupedal robots on complex and discrete terrain," *Science Robotics*, vol. 10, no. 102, p. eads6192, 2025.

[15] Z. Zhuang, S. Zhu, M. Zhao, and H. Zhao, "Deep whole-body parkour," *arXiv preprint arXiv:2601.07701*, 2026.

[16] Z. Wu, X. Huang, L. Yang, Y. Zhang, K. Sreenath, X. Chen, P. Abbeel, R. Duan, A. Kanazawa, C. Sferrazza *et al.*, "Perceptive humanoid parkour: Chaining dynamic human skills via motion matching," *arXiv preprint arXiv:2602.15827*, 2026.

[17] Z. Zhang, K. Wen, M. Xu, J. He, C. Li, T. Miki, C. Schwarke, C. Zhang, X. B. Peng, and M. Hutter, "Learning whole-body humanoid locomotion via motion generation and motion tracking," *arXiv preprint arXiv:2604.17335*, 2026.

[18] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, and H. Hirukawa, "Biped walking pattern generation by using preview control of zero-moment point," in *IEEE International Conference on Robotics and Automation (ICRA)*, vol. 2. IEEE, 2003, pp. 1620–1626.

[19] D. E. Orin, A. Goswami, and S.-H. Lee, "Centroidal dynamics of a humanoid robot," *Autonomous Robots*, vol. 35, no. 2-3, pp. 161–176, 2013.

[20] S. Kuindersma, R. Deits, M. Fallon, A. Valenzuela, H. Dai, F. Permenter, T. Koolen, P. Marion, and R. Tedrake, "Optimization-based locomotion planning, estimation, and control design for the atlas humanoid robot," *Autonomous Robots*, vol. 40, no. 3, pp. 429–455, 2016.

[21] J. Di Carlo, P. M. Wensing, B. Katz, G. Bledt, and S. Kim, "Dynamic locomotion in the MIT cheetah 3 through convex model-predictive control," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 1–9.

[22] M. Posa, C. Cantu, and R. Tedrake, "A direct method for trajectory optimization of rigid bodies through contact," *The International Journal of Robotics Research*, vol. 33, no. 1, pp. 69–81, 2014.

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

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

[25] Z. Li, X. Peng, P. Abbeel, S. Levine, G. Berseth, and K. Sreenath, "Robust and versatile bipedal jumping control through multi-task reinforcement learning. arxiv 2023," *arXiv preprint arXiv:2302.09450*, 2023.

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

[27] H. Wang, Z. Wang, J. Ren, Q. Ben, T. Huang, W. Zhang, and J. Pang, "Beamdojo: Learning agile humanoid locomotion on sparse footholds," *arXiv preprint arXiv:2502.10363*, 2025.

[28] J. Long, J. Ren, M. Shi, Z. Wang, T. Huang, P. Luo, and J. Pang, "Learning humanoid locomotion with perceptive internal model," in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2025.

[29] S. Luo, S. Li, R. Yu, Z. Wang, J. Wu, and Q. Zhu, "Pie: Parkour with implicit-explicit learning framework for legged robots," *IEEE Robotics and Automation Letters*, 2024.

[30] J. Sun, G. Han, P. Sun, W. Zhao, J. Cao, J. Wang, Y. Guo, and Q. Zhang, "Dpl: Depth-only perceptive humanoid locomotion via realistic depth synthesis and cross-attention terrain reconstruction," *arXiv preprint arXiv:2510.07152*, 2025.

[31] Y. Mu, Z. Zhang, Y. Shi, D. Yang, M. Matsumoto, K. Imamura, G. Tevet, C. Guo, M. Taylor, C. Shu, P. Xi, and X. B. Peng, "Smp: Reusable score-matching motion priors for physics-based character control," *arXiv preprint arXiv:2512.03028*, 2025.

[32] T. He, J. Gao, W. Xiao, Y. Zhang, Z. Wang, J. Wang *et al.*, "Asap: Aligning simulation and real-world physics for learning agile humanoid whole-body skills," *arXiv preprint arXiv:2502.01143*, 2025.

[33] W. Xie, J. Han, J. Zheng, H. Li, X. Liu, J. Shi, C. Bai, X. Li, and W. Zhang, "Kungfubot: Physics-based humanoid whole-body control for learning highly-dynamic skills," *arXiv preprint arXiv:2506.12851*, 2025.

[34] Z. Shen, H. Pi, Y. Xia, Z. Cen, S. Peng, Z. Hu, H. Bao, R. Hu, and X. Zhou, "World-grounded human motion recovery via gravity-view coordinates," in *SIGGRAPH Asia 2024 Conference Papers*, 2024, pp. 1–11.

[35] Q. Zhao, K. Yang, X. Wang, S. Zhao, Y. Lu, X. Zhang, Q. Shen, X.-X. Long, and X. Cao, "Make tracking easy: Neural motion retargeting for humanoid whole-body control," *arXiv preprint arXiv:2603.22201*, 2026.

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

[37] M. Mittal, C. Yu, Q. Yu, J. Liu, N. Rudin, D. Hoeller, J. L. Yuan, R. Singh, Y. Guo, H. Mazhar *et al.*, "Orbit: A unified simulation framework for interactive robot learning environments," *IEEE Robotics and Automation*

Letters, vol. 8, no. 6, pp. 3740–3747, 2023.

[38] J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter, "Learning quadrupedal locomotion over challenging terrain," *Science Robotics*, vol. 5, no. 47, p. eabc5986, 2020.

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

[40] Z. Li, X. B. Peng, P. Abbeel, S. Levine, G. Berseth, and K. Sreenath, "Reinforcement learning for versatile, dynamic, and robust bipedal locomotion control," *The International Journal of Robotics Research*, 2024.

[41] L. Pinto, M. Andrychowicz, P. Welinder, W. Zaremba, and P. Abbeel, "Asymmetric actor critic for image-based robot learning," in *Robotics: Science and Systems (RSS)*, 2018.

[42] M. Andrychowicz, B. Baker, M. Chociej, R. Józefowicz, B. McGrew, J. Pachocki, A. Petron, M. Plappert, G. Powell, A. Ray *et al.*, "Learning dexterous in-hand manipulation," *The International Journal of Robotics Research*, vol. 39, no. 1, pp. 3–20, 2020.

[43] A. Walsman, M. Zhang, S. Choudhury, D. Fox, and A. Farhadi, "Impossibly good experts and how to follow them," in *International Conference on Learning Representations (ICLR)*, 2022.

[44] H. Nguyen, A. Baisero, D. Wang, C. Amato, and R. Platt, "Leveraging fully observable policies for learning under partial observability," in *Conference on Robot Learning (CoRL)*. PMLR, 2023, pp. 1673–1683.

[45] A. Warrington, J. W. Lavington, A. Scibior, M. Schmidt, and F. Wood, "Robust asymmetric learning in POMDPs," in *International Conference on Machine Learning (ICML)*. PMLR, 2021, pp. 11 013–11 023.

[46] N. Messikommer, J. Xing, E. Aljalbout, and D. Scaramuzza, "Student-informed teacher training," *arXiv preprint arXiv:2412.09149*, 2025.

[47] N. Rudin, D. Hoeller, P. Reist, and M. Hutter, "Learning to walk in minutes using massively parallel deep reinforcement learning," in *Conference on Robot Learning (CoRL)*. PMLR, 2022, pp. 91–100.

[48] X. B. Peng, M. Andrychowicz, W. Zaremba, and P. Abbeel, "Sim-to-real transfer of robotic control with dynamics randomization," in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 3803–3810.

[49] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, "Domain randomization for transferring deep neural networks from simulation to the real world," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 23–30.

[50] J. P. Araujo, Y. Ze, P. Xu, J. Wu, and C. K. Liu, "Retargeting matters: General motion retargeting for humanoid motion tracking," *arXiv preprint arXiv:2510.02252*, 2025.

[51] M. Xu, Y. Shi, K. Yin, and X. B. 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*, 2025, pp. 1–11.

[52] J. Kim, S. Fahmi, S. Rho, S. Ha, and G. Nelson, "Flip stunts on bicycle robots using iterative motion imitation," *arXiv preprint arXiv:2603.27944*, 2026.

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

[54] Y. Hao, R. Yu, S. Luo, G. Zhang, J. Wu, and Q. Zhu, "Cref: Cross-modal and recurrent fusion for depth-conditioned humanoid locomotion," *arXiv preprint arXiv:2603.29452*, 2026.