Learning Whole-Body Humanoid Locomotion via Motion Generation and Motion Tracking

Anonymous Author(s)

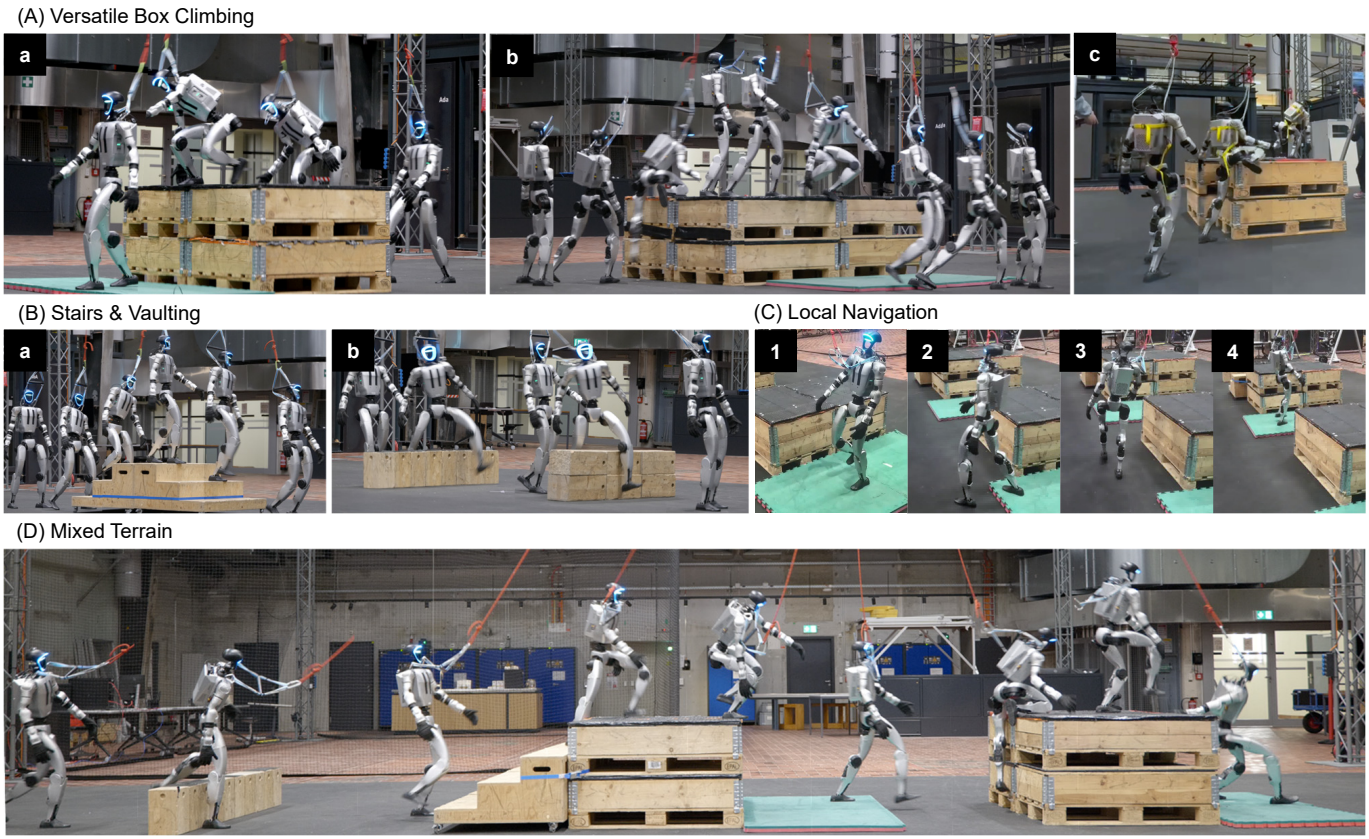

Fig. 1: Results of hardware experiments. (A) The robot climbs onto a 75 cm box and jumps down in three different ways: (a) straight ascent and descent; (b) straight ascent followed by a 90° right turn and side jump-down; and (c) ascent and descent from the box corner. (B) The robot traverses (a) a staircase and (b) a sequence of vaulting hurdles. (C) The robot shows local navigation behavior by bypassing the box to reach the target position. (D) The robot traverses a mixed terrain sequence consisting of a vaulting obstacle, stairs, and box climbing. https://wholebodylocomotion.github.io/.

***Abstract*—Whole-body humanoid locomotion is challenging due to high-dimensional control, morphological instability, and the need for real-time adaptation to various terrains using onboard perception. Directly applying reinforcement learning (RL) with reward shaping to humanoid locomotion often leads to lower-body-dominated behaviors, whereas imitation-based RL can learn more coordinated whole-body skills but is typically limited to replaying reference motions without a mechanism to adapt them online from perception for terrain-aware locomotion. To address this gap, we propose a whole-body humanoid locomotion framework that combines skills learned from reference motions with terrain-aware adaptation. We first train a diffusion model on retargeted human motions for real-time prediction of terrain-aware reference motions. Concurrently, we train a whole-body reference tracker with RL using this motion data. To improve robustness under imperfectly generated references, we further fine-tune the tracker with a frozen motion generator in a closed-loop setting. The resulting system supports directional goal-reaching control with terrain-aware whole-body adaptation, and can be deployed on a Unitree G1 humanoid robot with onboard perception and computation. The hardware experiments demonstrate successful traversal over boxes, hurdles, stairs, and mixed terrain combinations. Quantitative results further show the benefits of incorporating online motion generation and fine-tuning the motion tracker for improved generalization and robustness.**

I. Introduction

Developing whole-body perceptive humanoid locomotion remains a challenging yet important step toward expanding the operational range of humanoid robots. While deep reinforcement learning (RL) has achieved remarkable success in enabling highly dynamic behaviors on quadrupedal systems [1–6], extending these capabilities to humanoid robots remains

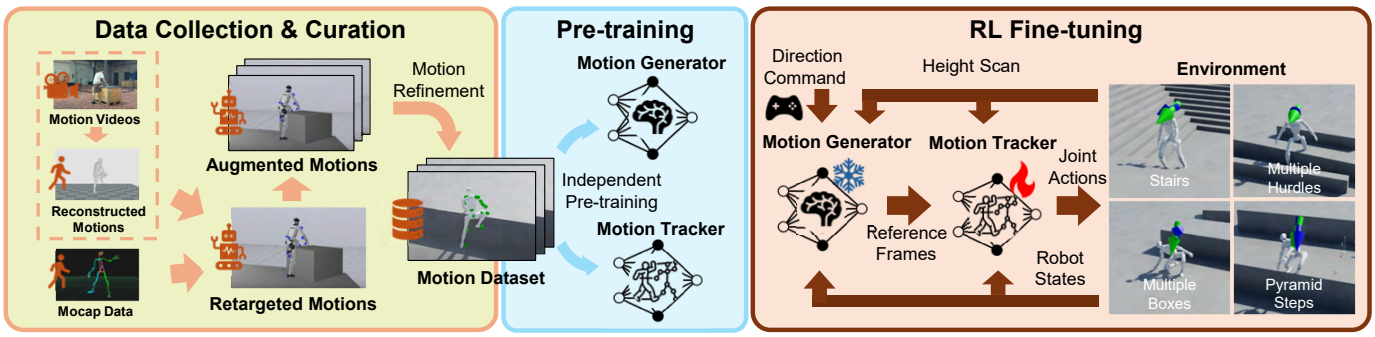

Fig. 2: Overview of the training framework. (a) Data Collection & Curation: whole-body robot motions are obtained from human motions through motion reconstruction and retargeting, followed by motion augmentation. (b) Pre-training: the diffusion-based motion generator and RL-based motion tracker are trained solely on the **offline motion dataset** from stage (a). (c) RL Fine-tuning: the motion tracker is **fine-tuned with RL** under randomized direction commands and on more diverse terrains with randomized configurations, including stairs, multiple vaulting hurdles, boxes, and pyramid-shaped high steps, while running the frozen motion generator.

considerably difficult, as humanoids possess higher degrees of freedom together with more complex morphological constraints. As a result, training an RL-based whole-body humanoid controller solely through reward shaping can be non-trivial and often suffers from inefficient exploration. Without explicit structural guidance, vanilla RL-based humanoid controllers frequently converge to locomotion strategies with limited whole-body coordination, making it difficult to acquire coordinated behaviors for challenging terrain traversal. This limitation becomes especially evident in tasks such as humanoid parkour, where coordinated whole-body movements are essential for traversing large and complex obstacles.

To alleviate the difficulty of reward shaping and inefficient exploration, motion imitation via RL has emerged as a promising alternative [7–12]. By leveraging reference data, motion tracking can efficiently transfer highly coordinated whole-body skills to the robot. However, pure motion trackers are inherently limited to replaying choreographed trajectories and generally lack the adaptability and reactivity required for operation in unstructured and diverse environments. Achieving human-level perceptive locomotion therefore requires a higher-level composition mechanism that can dynamically adjust locomotion style across varied obstacles based on real-time perceptive inputs.

One direct solution for skill composition is to distill multiple specialized teacher policies [13], or to compose multiple skills from separate reference trajectories within a single imitation-based teacher framework [14, 15]. By learning distinct expert skills for different terrain conditions, a unified distilled policy can learn to transition among them using exteroceptive observations. However, current methods have not yet demonstrated strong scalability to a diverse set of skills or robust generalization to new terrains.

Alternatively, generative models have recently shown strong potential for scalable motion control. Training generative models on large-scale motion datasets has proven effective in both character control [16–18] and robot control [9, 19], naturally reducing the engineering effort required as the motion repertoire expands. However, directly deploying expressive generated kinematic trajectories on real humanoid locomotion systems may introduce motion artifacts, such as foot sliding or temporal discontinuities, which can lead to catastrophic failure. To mitigate such issues, recent work [18, 20, 21] combines diffusion-based motion generation with RL-based motion tracking, allowing the tracker to output physically plausible actions instead of relying solely on kinematic generation. Despite this progress, their applicability to real humanoid robots in locomotion tasks requiring whole-body coordination remains insufficiently explored, and their deployment may potentially be hindered by slow inference and limited robustness to real-world disturbances.

To address these limitations, we propose a whole-body humanoid locomotion system that adapts a motion generation and motion tracking framework to real-time robot control for challenging humanoid locomotion tasks. In particular, we repurpose coordinated human whole-body skills learned from motion data for perceptive humanoid locomotion, and build a control framework that combines diffusion-based motion generation with RL-based motion tracking. The resulting system supports directional goal-reaching control together with terrain-aware whole-body adaptation. Our hardware experiments show that our approach enables efficient skill execution and generalization across diverse terrains, including terrains unseen during training. Our results further demonstrate the effectiveness of using a motion generator trained on human motion data to guide an RL-based controller through challenging terrains requiring diverse skills, without heavily engineered distillation strategies.

II. Method

The overall framework is illustrated in Fig. 2. Our method consists of three stages: (a) collecting whole-body motion data for different terrains and converting them into dynamically feasible robot trajectories; (b) pre-training the RL-based motion tracker and the diffusion-based motion generator on this dataset; and (c) fine-tuning the motion tracker on more diverse terrain configurations, while keeping the motion generator frozen and running it in a receding-horizon manner during deployment. We describe each stage in Sec. B.

TABLE I: Comparative Success Rates of Fixed-Reference Motion Tracking and The Proposed Full System Across Three Evaluation Settings.

(a) Box Climbing

Box Height [cm]	Yaw [deg]	Tracker Only	Tracker+Gen.
40	0	0.982	1.000
50	-30	0.988	0.980
	30	1.000	0.982
	-15	0.974	1.000
	15	0.970	0.998
	0	0.998	1.000
60	0	0.984	0.996
70	0	0.606	0.966
80	0	0.230	0.962
Mean ± SD		0.859 ± 0.252	**0.987 ± 0.014**

(b) Vaulting

Hurdle Height [cm]	Yaw [deg]	Tracker Only	Tracker+Gen.
25	0	0.978	1.000
35	-40	0.808	1.000
	40	0.492	1.000
	-20	0.922	1.000
	20	0.954	1.000
	0	0.988	1.000
45	0	0.950	1.000
55	0	0.348	0.920
Mean ± SD		0.805 ± 0.231	**0.990 ± 0.026**

(c) Ascending Stairs

Step Height [cm]	Yaw [deg]	Tracker Only	Tracker+Gen.
15	0	1.000	1.000
20	-30	0.948	1.000
	30	0.916	0.994
	-15	0.970	1.000
	15	0.990	0.996
	0	0.980	1.000
25	0	0.114	0.986
Mean ± SD		0.845 ± 0.300	**0.997 ± 0.005**

III. Experiments

In this section, we present the experimental results of the proposed humanoid locomotion system in simulation and on real hardware. We also provide quantitative evaluation to examine the contribution of motion generation to generalization, and the effect of RL fine-tuning of the motion tracker on robustness and success rates across terrain types.

A. Hardware Results

We evaluate the proposed pipeline on real hardware as shown in 1, and the supplementary video. The hardware experiments demonstrate both the effectiveness of our controller and its generalization ability across diverse terrains. In particular, we evaluate the controller on a range of terrain types, including box ascent and descent, stair ascent and descent, and continuous vaulting over obstacles of varying heights, as well as on more challenging compound terrain settings that combine vaulting, stairs, and box climbing within a single run. For box ascent and descent, our controller exhibits versatile climbing behaviors, including box traversal from different approach directions as well as the ability to traverse a box with on-top reorientation before descent. During ascent, the robot uses its knees and hands for support, while during descent it jumps off the box and uses the hands for cushioning, consistent with the motion style observed in the training data. For vaulting, the robot can traverse multiple hurdles of different heights, typically crossing over them directly rather than stepping onto the hurdle first. For stairs, the controller switches to stair ascent and descent behaviors. When these terrain types are combined, the controller can also transition dynamically among different motion styles according to the terrain, including sequences such as stair ascent following a jump-down motion, even though such terrain combinations are not seen during RL fine-tuning.

In addition, as shown in Fig. 1C and the supplementary video, we observe a form of local navigation behavior in the system after RL fine-tuning: when the robot is not in a suitable state to traverse an obstacle under the direction command computed from the given goal position using the motion generated by the planner, the tracking policy may partially override the reference motion and instead bypass the obstacle from the side to avoid failure and still reach the goal.

B. Quantitative Results

Beyond the hardware experiments, we further analyze two key design choices of our framework through quantitative simulation results: whether online motion generation improves adaptation and generalization across diverse terrains at deployment, and whether the additional fine-tuning stage is necessary after coupling the generator with the motion tracker. As shown in Table I and Fig. 3, both components are important for the final system.

1) Benefit of Online Motion Generation: We first evaluate whether online motion generation is beneficial for enabling the final system to adapt to diverse terrain variations by comparing it with fixed-reference tracking. We compare two settings on three representative tasks: box climbing, vaulting, and stair ascent. For box climbing, the reference trajectory is a motion for climbing onto a 50 cm box; for vaulting, the reference is a trajectory for traversing a 35 cm-high and 20 cm-wide hurdle; for stair ascent, the reference is a trajectory for ascending a staircase with 30 cm-wide and 20 cm-high steps. We then modify the terrain geometry at test time by changing the obstacle height or rotating the terrain mesh in yaw. **Tracker Only** uses the fine-tuned motion tracker with a fixed reference trajectory, and success is measured by whether the robot reaches the final XY base position of the reference trajectory. **Tracker + Gen.** uses the proposed full system, where the motion generator produces reference motions online from the target direction and terrain input, and success is measured by whether the robot reaches the goal position placed on the terrain. The goal position is chosen to match the final base position of the reference trajectory used in the *Tracker Only* setting, and the direction command is computed online accordingly. For each task, 500 robots are spawned in simulation with randomized initial poses and joint positions.

As shown in Table I, the full system with online motion generation achieves better generalization and higher average success rates across all three tasks than fixed-reference tracking on the corresponding terrain settings. For box climbing, the *Tracker Only* baseline performs well close to the nominal reference condition, but becomes brittle as the obstacle height increases. In contrast, *Tracker + Gen.* remains consistently robust across all tested settings. For vaulting, the fine-tuned tracker already exhibits some local generalization and can handle different hurdle heights, but its performance still degrades under larger changes in terrain height and orientation. The full system, by comparison, remains stable across all evaluated cases. A similar pattern appears in stair ascent: while the fixed-reference tracker can tolerate yaw perturbations, it deteriorates markedly under the height increase, whereas the

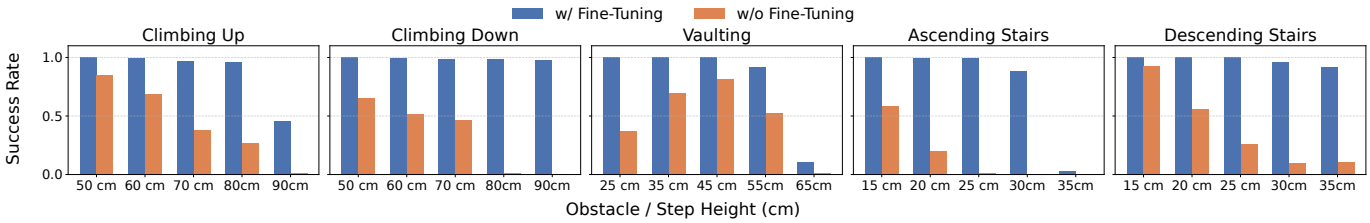

Fig. 3: Comparative results of the success rates of the motion tracker with and without fine-tuning across five terrain traversal tasks, including climbing up/down, vaulting, and stair ascent/descent. For each task, we spawn 500 robots in simulation with randomized initial poses and assign each of them a goal position that guides traversal of the corresponding terrain obstacle, from which the target direction is computed. The results consistently show that fine-tuning the tracker with a frozen motion generator improves the success rate of reaching the goal.

full system continues to achieve nearly perfect performance.

These results highlight two points. First, the fine-tuned tracker is not entirely rigid. Because it observes terrain height scans and is fine-tuned on diverse terrain configurations, it can retain a limited degree of local terrain generalization. This is particularly visible in vaulting and box climbing, where nearby obstacle scales can sometimes be handled even without adapting the reference motion. Second, this limited generalization is fundamentally insufficient for robust perceptive locomotion. Once the terrain geometry changes more substantially, especially in ways that require changing the timing or style of the motion rather than merely tolerating small mismatch, the fixed-reference tracker degrades rapidly. During deployment, terrain-aware reference motions produced online by the motion generator allow the full system to better adapt its traversal behavior to diverse obstacle geometries. Overall, the results in Table I highlight the importance of online motion generation for adaptation and generalization across diverse terrains in the final system.

2) Impact of Fine-Tuning the Motion Tracker: The above results show that online motion generation is important for generalization during deployment. We next ask whether the additional fine-tuning stage is also necessary, or whether the pretrained tracker can simply be combined with the motion generator directly. To answer this question, we compare the pretrained tracker and the fine-tuned tracker under the same motion generator, as shown in Fig. 3. We evaluate five terrain traversal tasks, namely climbing up, climbing down, vaulting, ascending stairs, and descending stairs. For each task, a goal position is placed behind the terrain obstacle, from which the direction command is computed, and 500 robots are spawned in simulation to measure the success rate.

Figure 3 shows that fine-tuning the tracker consistently improves success rates across all evaluated tasks, with the largest gains appearing on the more difficult terrain settings. For climbing up and climbing down, the improvement becomes more apparent as the box height increases, indicating that fine-tuning is especially important when the robot is required to react robustly to more demanding vertical transitions. For stair ascent and descent, the fine-tuned tracker also shows clear advantages on larger step heights. Vaulting is comparatively easier than the other tasks, but the fine-tuned tracker still remains more reliable overall. Taken together, the histogram suggests that fine-tuning the tracker with RL does not merely provide marginal refinement, but systematically improves robustness across a broad range of traversal tasks.

A likely reason is that the pretrained tracker is optimized on a fixed set of offline motion trajectories, which are relatively smooth and physically filtered. During deployment, however, the online references generated by the motion generator are conditioned on noisy robot state, target direction, and terrain observations, and may therefore contain temporal discontinuities or motion artifacts. This creates a distribution mismatch between the tracker training references and the actual reference stream encountered during control. Fine-tuning with the frozen generator is therefore crucial: it allows the tracker to adapt to the distribution and imperfections of generated references, learn how to make effective use of motion patterns for task completion, and suppress unsafe behaviors that could otherwise lead to failure. This observation is also consistent with the qualitative behavior discussed in Sec. B.III, where the fine-tuned tracker effectively acts as a motion filter over the generated references. Overall, Fig. 3 shows that the fine-tuning stage is an important part of making the combined generation-and-tracking framework robust in practice.

IV. Conclusion

Overall, our results demonstrate that combining motion generation with motion tracking is an effective way to realize perceptive whole-body humanoid locomotion. The proposed system enables efficient traversal over diverse terrain types and terrain combinations unseen during training, while the quantitative analysis highlights the importance of both online motion generation and tracker fine-tuning for adaptation and robustness.

One limitation of the current system is that it relies on a LiDAR-based elevation mapping pipeline to reconstruct the surrounding environment. When the mapping quality degrades due to sensing noises, the locomotion performance can deteriorate substantially. A promising direction for future work is to adopt perception architectures that do not depend as strongly on high-quality geometric sensing, such as neural mapping [6], or to incorporate a belief encoder [2] that compensates for sensing failures through proprioceptive observations. Another important direction is to extend the framework beyond pure locomotion to a broader range of loco-manipulation tasks and to more challenging outdoor environments.

References

[1] 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.

[2] 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, 2022.

[3] J. He, C. Zhang, F. Jenelten, R. Grandia, M. Bächer, and M. Hutter, "Attention-based map encoding for learning generalized legged locomotion," *Science Robotics*, vol. 10, no. 105, p. eadv3604, 2025.

[4] 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," 2025.

[5] Z. Zhang, C. Li, T. Miki, and M. Hutter, "Motion priors reimagined: Adapting flat-terrain skills for complex quadruped mobility," 2025.

[6] C. Zhang, V. Klemm, F. Yang, and M. Hutter, "Ame-2: Agile and generalized legged locomotion via attention-based neural map encoding," 2026.

[7] X. B. Peng, P. Abbeel, S. Levine, and M. van de Panne, "Deepmimic: Example-guided deep reinforcement learning of physics-based character skills," *ACM Trans. Graph.*, vol. 37, no. 4, pp. 143:1–143:14, Jul. 2018.

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

[9] 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," 2025.

[10] M. Ji, X. Peng, F. Liu, J. Li, G. Yang, X. Cheng, and X. Wang, "Exbody2: Advanced expressive humanoid whole-body control," *arXiv preprint arXiv:2412.13196*, 2024.

[11] Y. Ze, Z. Chen, J. P. Araújo, Z. ang Cao, X. B. Peng, J. Wu, and C. K. Liu, "Twist: Teleoperated whole-body imitation system," 2025.

[12] 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," 2025.

[13] Y. Wang, T. Leng, C. Lin, S. Liu, S. Simon, B. Chen, J. Francis, and D. Zhao, "Apex: Learning adaptive high-platform traversal for humanoid robots," 2026.

[14] Z. Wu, X. Huang, L. Yang, Y. Zhang, K. Sreenath, X. Chen, P. Abbeel, R. Duan, A. Kanazawa, C. Sferrazza, G. Shi, and C. K. Liu, "Perceptive humanoid parkour: Chaining dynamic human skills via motion matching," 2026.

[15] D. Wang, X. Wang, C. Zhang, J. Shi, Y. Zhao, C. Bai, and X. Li, "X-loco: Towards generalist humanoid locomotion control via synergetic policy distillation," 2026.

[16] X. Huang, T. Truong, Y. Zhang, F. Yu, J. P. Sleiman, J. Hodgins, K. Sreenath, and F. Farshidian, "Diffuse-cloc: Guided diffusion for physics-based character look-ahead control," *ACM Transactions on Graphics*, vol. 44, no. 4, p. 1–12, Jul. 2025.

[17] G. Tevet, S. Raab, B. Gordon, Y. Shafir, D. Cohen-or, and A. H. Bermano, "Human motion diffusion model," in *The Eleventh International Conference on Learning Representations*, 2023.

[18] G. Tevet, S. Raab, S. Cohan, D. Reda, Z. Luo, X. B. Peng, A. H. Bermano, and M. van de Panne, "Closd: Closing the loop between simulation and diffusion for multi-task character control," 2024.

[19] C. Chi, S. Feng, Y. Du, Z. Xu, E. Cousineau, B. Burchfiel, and S. Song, "Diffusion policy: Visuomotor policy learning via action diffusion," in *Proceedings of Robotics: Science and Systems (RSS)*, 2023.

[20] A. Serifi, R. Grandia, E. Knoop, M. Gross, and M. Bächer, "Robot motion diffusion model: Motion generation for robotic characters," in *SIGGRAPH Asia 2024 Conference Papers*, ser. SA '24. New York, NY, USA: Association for Computing Machinery, 2024.

[21] M. Xu, Y. Shi, K. Yin, and X. B. Peng, "Parc: Physics-based augmentation with reinforcement learning for character controllers," in *SIGGRAPH 2025 Conference Papers (SIGGRAPH '25 Conference Papers)*, 2025.

[22] N. Chentanez, M. Müller, M. Macklin, V. Makoviychuk, and S. Jeschke, "Physics-based motion capture imitation with deep reinforcement learning," in *Proceedings of the 11th ACM SIGGRAPH Conference on Motion, Interaction and Games*, ser. MIG '18. New York, NY, USA: Association for Computing Machinery, 2018.

[23] Z. Luo, Y. Yuan, T. Wang, C. Li, S. Chen, F. Castañeda, Z.-A. Cao, J. Li, D. Minor, Q. Ben, X. Da, R. Ding, C. Hogg, L. Song, E. Lim, E. Jeong, T. He, H. Xue, W. Xiao, Z. Wang, S. Yuen, J. Kautz, Y. Chang, U. Iqbal, L. Fan, and Y. Zhu, "Sonic: Supersizing motion tracking for natural humanoid whole-body control," *arXiv preprint arXiv:2511.07820*, 2025.

[24] C. Li, M. Vlastelica, S. Blaes, J. Frey, F. Grimminger, and G. Martius, "Learning agile skills via adversarial imitation of rough partial demonstrations," in *Proceedings of The 6th Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, K. Liu, D. Kulic, and J. Ichnowski, Eds., vol. 205. PMLR, 14–18 Dec 2023, pp. 342–352.

[25] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," 2017.

[26] Y. Ze, S. Zhao, W. Wang, A. Kanazawa, R. Duan, P. Abbeel, G. Shi, J. Wu, and C. K. Liu, "Twist2: Scalable, portable, and holistic humanoid data collection system," 2025.

[27] T. He, Z. Luo, W. Xiao, C. Zhang, K. Kitani, C. Liu, and

G. Shi, "Learning human-to-humanoid real-time whole-body teleoperation," *arXiv preprint arXiv:2403.04436*, 2024.
[28] D. Hoeller, N. Rudin, D. Sako, and M. Hutter, "Anymal parkour: Learning agile navigation for quadrupedal robots," *Science Robotics*, vol. 9, no. 88, p. eadi7566, 2024.
[29] S. Ross, G. J. Gordon, and J. A. Bagnell, "A reduction of imitation learning and structured prediction to no-regret online learning," 2011.
[30] N. Mahmood, N. Ghorbani, N. F. Troje, G. Pons-Moll, and M. J. Black, "AMASS: Archive of motion capture as surface shapes," in *International Conference on Computer Vision*, Oct. 2019, pp. 5442–5451.
[31] 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 Conference Proceedings*, 2024.
[32] D. D. Team, "Drake: Model-based design and verification for robotics," 2019. [Online]. Available: https://drake.mit.edu
[33] NVIDIA, :, M. Mittal *et al.*, "Isaac lab: A gpu-accelerated simulation framework for multi-modal robot learning," 2025.
[34] K. Chen, R. Nemiroff, and B. T. Lopez, "Direct lidar-inertial odometry: Lightweight lio with continuous-time motion correction," *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3983–3989, 2023.
[35] T. Miki, L. Wellhausen, R. Grandia, F. Jenelten, T. Homberger, and M. Hutter, "Elevation mapping for locomotion and navigation using gpu," 2022.

APPENDIX

A. RELATED WORK

I. Whole-body Motion Tracking for Humanoids

In simulated whole-body character control, prior works [7, 22] have explored RL-based frameworks in which characters learn to imitate reference motion clips. By training a control policy to minimize the discrepancy between the reference motion and the executed motion, the controller is able to reproduce a wide range of dynamic maneuvers while remaining consistent with the physical constraints of the controlled system.

More recently, this paradigm has been widely adopted in quadruped and humanoid robot control [9, 12, 23, 24]. By designing imitation rewards and regularization terms, and optimizing the policy with the PPO algorithm [25], recent works have demonstrated strong capability in imitating large-scale motion datasets and even enabling real-time teleoperation [11, 26, 27]. However, for general locomotion tasks, pure motion tracking remains fundamentally limited in its adaptability to varying terrains.

II. Motion Generation for Motion Control

Motion generation with generative models provides an effective way to sample feasible motion trajectories from curated motion datasets. Prior work [17] has demonstrated that diffusion models can generate smooth kinematic motion trajectories conditioned on text descriptions by incorporating additional geometric losses. Building on this line of work, Tevet et al. [18] further extend it to physical character control by coupling the generated motion with an RL-based motion tracker, which helps correct artifacts introduced by the generative model. Serifi et al. [20] also combine motion generation with RL-based motion tracking, and further fine-tune the generator using value information from the critic network of the tracking policy. Xu et al. [21] adopt a similar design philosophy to [18] and extend it to humanoid parkour in simulation, demonstrating impressive whole-body motion control conditioned on surrounding terrain observations. However, these methods mainly target simulated character control, and may still be constrained by the inference latency of diffusion models and lack of robustness for real robotic systems. As a result, relatively few real-time whole-body humanoid locomotion systems with integrated exteroceptive sensing have been demonstrated.

Another line of work [9, 16] generates joint state-action pairs without relying on a separate motion tracker. These methods have been demonstrated on humanoid robots for tasks such as waypoint navigation and motion inpainting. However, because control is produced directly from the co-diffusion process, they may still suffer from motion artifacts that can lead to unstable locomotion. Moreover, more dynamic and demanding tasks, such as humanoid whole-body locomotion, are not well-studied in this setting. In our work, we follow the general framework of [21], and adapt it to real-world humanoid locomotion tasks that require both whole-body coordination and real-time reaction to terrain conditions.

III. Skill Composition for Perceptive Locomotion Control

Composing multiple skills into a unified policy is often regarded as an effective shortcut toward whole-body visuomotor locomotion control, compared with training a single policy entirely from scratch. Hoeller et al. [28] first train multiple expert skills for quadrupedal locomotion across different terrain types, and then learn a high-level planner that switches categorically among the expert policies. In contrast, Rudin et al. [4] employ DAgger [29] to first distill several expert policies into a single policy, and subsequently fine-tune the distilled policy with RL on new terrains.

For humanoid whole-body locomotion, concurrent work [13] adopts a similar strategy to [4]: it first trains terrain-specific expert policies from scratch with RL and then distills them into a unified student policy. Wu et al. [14] likewise apply a similar distillation framework, but construct the expert skills from imitation-based RL policies trained with whole-body motion references. These methods have shown promising results for terrain traversal requiring coordinated whole-body control. However, both approaches may rely on carefully designed distillation procedures, such as expert assignment and data distribution design, to efficiently learn transitions and combinations across different skills.

In our work, instead of explicitly engineering a distillation pipeline over multiple expert skills, we use a diffusion-based motion generator trained on retargeted human motion data as a skill composition module that produces corresponding motion references for a low-level tracking policy based on the environment conditions. In addition, to reduce the impact of motion artifacts introduced by the generator and improve control performance, we further fine-tune the motion tracking policy with RL under randomized direction commands and more diverse terrain configurations while using a frozen motion generator pretrained on the offline motion dataset. More details are provided in Sec. B.III.

B. Details for Methods

I. Data Collection & Curation

Our initial motion data consist of approximately 5 minutes of motion clips collected from two sources: motion videos captured by ourselves and public motion datasets [12, 30]. The dataset contains one representative motion type for each terrain skill, namely climbing onto a 50 cm box, vaulting over a 35 cm hurdle, jumping down from a 50 cm box, and stair ascent and descent on steps of approximately 20 cm height, together with omnidirectional walking motions such as straight walking and turning. To enlarge this limited initial dataset and improve terrain diversity, we further apply motion augmentation. In total, the resulting dataset contains approximately one hour of motion data for pre-training the motion generator and motion tracker.

(1) Initial Dataset Construction: For motion skills recorded from videos, we first reconstruct human motions from raw videos using GVHMR [31]. We then retarget the reconstructed motions both from videos and motion datasets to the humanoid robot using a contact-constrained IK solver [32], similar to [12]. After obtaining the retargeted robot motions, we manually place terrain objects to align with the original motions and preserve the physical interaction between the robot and the environment. To ensure that the final training data are physically plausible for both the motion tracker and the motion generator, we do not directly use the raw retargeted trajectories; instead, we refine training motions from a DeepMimic-style tracking policy trained on those retargeted motions.

(2) Motion Augmentation: To further improve the generalization of both the motion tracker and the motion generator, we enlarge the dataset through kinematics-based augmentation. Specifically, we augment existing motions by varying terrain geometry, including scaling obstacle heights and inserting random small boxes along the motion path, and then optimizing the motions to remain consistent with the modified terrain. The optimization includes losses such as terrain penetration and motion smoothness losses to reduce collisions and discontinuities [21]. As in the initial dataset construction, we do not use the optimized trajectories directly; instead, we train a tracking policy on them and record the resulting physically feasible trajectories. This allows us to extend motions defined for one obstacle scale to motions over terrains with different heights. As a result, the augmented dataset covers box climbing and jump-down motions on 35–75 cm boxes, vaulting motions over 25–45 cm hurdles, stair motions on steps of 15–20 cm height, and omnidirectional walking motions with randomly placed small boxes.

II. Pre-training Stage

The pre-training stage equips both the motion generator and the motion tracker with the basic ability to react to different environments and to output appropriate motion references and joint actions.

(1) Whole-body Motion Tracker: We first train a single whole-body motion tracker using a DeepMimic-style RL framework with PPO in IsaacLab [33], based on the motion data collected in Sec. B.I. We design imitation rewards r_{mimic} and regularization terms r_{reg} (see Table II) so that the policy can imitate all reference motions while respecting the physical constraints of the real hardware:

$$R_{\text{pre}} = r_{\text{mimic}} + r_{\text{reg}}. \tag{1}$$

The tracker observation consists of the reference state, including linear and angular velocity, joint position and velocity, and key body positions in the base frame; proprioceptive terms, including five consecutive frames of base angular velocity, projected gravity, joint position and velocity, and the previous action; together with terrain height scans. The policy outputs 23-dimensional target joint positions as actions. Although Yang et al. [12] show that terrain information is not strictly required when the demonstrations are sufficiently accurate, we find that it can be beneficial to include exteroceptive input already at this stage, as it helps the later fine-tuning stage.

(2) Diffusion-based Motion Generator: Our motion generator is based on a diffusion model that predicts whole-body reference motion sequences, following the architecture proposed in MDM [17, 21]. The model predicts future motion features over a 0.5-second horizon (25 frames), including root position in $\mathbb{R}^3$, root orientation in $\mathbb{R}^4$, joint positions in $\mathbb{R}^{23}$, and body link positions in $\mathbb{R}^{23\times 3}$, conditioned on the target heading vector, terrain height scan, and the past two frames of motion features.

During training, we randomly sample motion sequences from the dataset, compute the heading vector from the base pose difference, and use the first two frames as conditions to predict the remaining frames. The reconstruction loss is then computed by comparing the predicted motion with the ground-truth sequence. In addition to the reconstruction loss, we incorporate geometric losses including velocity, joint consistency, and terrain penetration losses, similar to [21]. The generator is trained on the collected motion dataset and later serves as the motion planner that provides reference motions for the tracking policy over different terrains. To improve robustness against noisy robot states, we additionally perturb the height scans and previous-state conditions during training.

However, as discussed in Sec. A.II, generated motions may still contain artifacts, which can lead to failure when directly paired with a tracker trained mainly on high-quality references.

Simply combining the pretrained motion generator and motion tracker can therefore result in limited robustness and reliability.

III. RL Fine-tuning Stage

In this stage, we improve the robustness and reliability of the pretrained motion generator and motion tracker, and adapt the full control system for real-time closed-loop deployment on a real humanoid robot. Directly combining the pretrained generator and tracker may still suffer from motion artifacts, tracking failures caused by imperfect references, and limited generalization to terrains and target directions beyond those covered by the training dataset. Therefore, we introduce an additional fine-tuning stage in which the motion tracker is fine-tuned with RL on more diverse terrains and under varied target direction conditions, while the motion generator is kept frozen.

During this stage, the motion generator produces reference frames conditioned on the past two frames of robot states to form a closed-loop motion prediction process, rather than relying on autoregressive feedback of its own previous predictions. To reduce deployment latency under limited onboard computation, we use only 2 denoising steps in the motion generator. We also inject additional noise into both the motion generator inference process and the tracker observations to improve robustness against real-world disturbances.

In addition, we randomize the target heading direction and introduce a heading tracking reward as a task reward r_{task} (see Table II), together with the imitation r_{mimic} and regularization terms r_{reg} , to encourage the full system to follow the desired direction while leveraging the motion priors acquired during pre-training:

$$R_{\text{post}} = r_{\text{mimic}} + r_{\text{reg}} + r_{\text{task}}. \tag{2}$$

We further introduce more diverse training terrains, including stairs with 15–25 cm step heights, continuous vaulting hurdles of 25–55 cm height, and multiple climbing boxes and pyramid steps ranging from 30–85 cm in height with varying widths as well as yaw and pitch angles.

After fine-tuning, the tracking policy effectively acts as a motion filter: it tracks the reference produced by the generator, while adjusting its execution to suppress unsafe behaviors using exteroceptive terrain observations. Despite the reduced number of denoising steps, which may further degrade raw generation quality, the fine-tuned tracker still robustly tracks the motion patterns produced by the generator and successfully traverses diverse terrains. Randomization of target headings and terrain configurations further enables behaviors beyond those present in the offline data, such as corner box climbing, reorientation on the box before descent, and continuous vaulting.

IV. Details for Hardware Deployment

For hardware experiments, the entire pipeline is deployed fully onboard the Unitree G1 robot. We use DLIO [34] with the onboard Livox MID360 LiDAR and IMU to estimate the robot base pose, which is used as an input to motion generation. For terrain perception, we employ Elevation Mapping CuPy [35] together with the DLIO base pose to reconstruct terrain height information over different real-world environments (See Fig. 4).

Since the neck joint of the G1 robot is passive, aggressive motions can induce head movement that degrades base pose estimation. To improve state-estimation stability, we fuse the LiDAR IMU and torso IMU measurements by subtracting the orientation estimated by DLIO from the moving-average pitch measured by the torso IMU, which allows us to estimate the head pitch variation and compensate for its effect.

To accelerate diffusion-based motion generation, we deploy the generator with TensorRT, reducing inference time to approximately 0.02 s. Although the motion generator operates with a 0.5-second prediction horizon (2 Hz) during fine-tuning, we update the reference per 0.25 seconds in a receding-horizon way during deployment to improve responsiveness to the surrounding environment. The motion generator runs on a Jetson Thor mounted on the back of the robot, while the motion tracking controller and the remaining onboard modules run on the Jetson Orin integrated into the Unitree G1 platform.

TABLE II: Reward Equations for RL-based Motion Tracker

Group	Name	Equation
r_{mimic}	Base Pose Tracking	$3\exp\left(-4\left\|\mathbf{p}_b^w - \mathbf{p}_b^{w*}\right\|^2 - \left\|\mathbf{q}_b \otimes \mathbf{q}_b^{*-1}\right\|^2\right)$
	Base Velocity Tracking	$\exp\left(-\left\|\mathbf{v}^b - \mathbf{v}^{b*}\right\|^2 - 0.1\left\|\boldsymbol{\omega}^b - \boldsymbol{\omega}^{b*}\right\|^2\right)$
	Joint Position Tracking	$2.5\exp\left(-2\sum_{i=1}^{23} 0.25(q_i - q_i^*)^2\right)$
	Joint Velocity Tracking	$0.5\exp\left(-2\sum_{i=1}^{23} 0.01(\dot{q}_i - \dot{q}_i^*)^2\right)$
	Body Pos Tracking (Base)	$2.0\exp\left(-\sum_{i=1}^{9}\left\|\mathbf{p}_{l,i}^b - \mathbf{p}_{l,i}^{b*}\right\|^2\right)$
	Body Pos Tracking (Global)	$\exp\left(-\sum_{i=1}^{9}\left\|\mathbf{p}_{l,i}^w - \mathbf{p}_{l,i}^{w*}\right\|^2\right)$
r_{reg}	Action Rate (First-Order)	$-0.1\sum_{i=1}^{23}(a_{t,i} - a_{t-1,i})^2$
	Action Rate (Second-Order)	$-0.05\sum_{i=1}^{23}(a_{t,i} - 2a_{t-1,i} + a_{t-2,i})^2$
	Joint Position Limits	$-10.0\sum_{i=1}^{23}\max(q_i - q_{lim}, 0)$
	Joint Velocity Limits	$-\sum_{i=1}^{23}\max(\dot{q}_i - \dot{q}_{lim}, 0)$
	Torque Limits	$-0.001\sum_{i=1}^{23}\max(\tau_i - \tau_{lim}, 0)$
	Joint Torques	$-0.00001\sum_{i=1}^{23}\tau_i^2$
	Body Linear Acceleration	$-0.01\sum_{i=1}^{9}(\ddot{p}_{l,i}^w)^2$
r_{task}	Target Head Tracking	$\langle \mathbf{v}^b, \mathbf{d}\rangle / \left\|\mathbf{v}^b\right\|$

TABLE III: Symbols used in Table II

Symbol	Description
$(\cdot)^*$	Reference quantity
$(\cdot)^b$	Expressed in the base frame
$(\cdot)^w$	Expressed in the world frame
$\mathbf{p}_b$	Base position
$\mathbf{q}_b$	Base orientation quaternion
$\mathbf{v}_b$	Base linear velocity
$\boldsymbol{\omega}_b$	Base angular velocity
$\mathbf{p}_{l,i}$	Position of key body i
q_i	Position of joint i
$\dot{q}_i$	Velocity of joint i
$\ddot{q}_i$	Acceleration of joint i
τ_i	Torque of joint i
q_{lim}	Joint position limit
$\dot{q}_{lim}$	Joint velocity limit
τ_{lim}	Joint torque limit

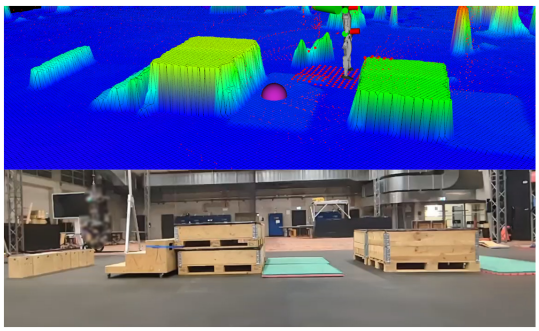

Fig. 4: Terrain reconstruction on the parkour course using Elevation Mapping CuPy [35]. Top: reconstructed elevation map. Bottom: real-world testing terrain.