Embrace Contacts: Humanoid Shadowing with Full Body Ground Contacts

Ziwen Zhuang
IIIS Tsinghua University,
Shanghai Qi Zhi Institute,
China
zhuangzw24@mails.tsinghua.edu.cn

Hang Zhao
IIIS Tsinghua University,
Shanghai Qi Zhi Institute,
China
zhaohang0124@gmail.com

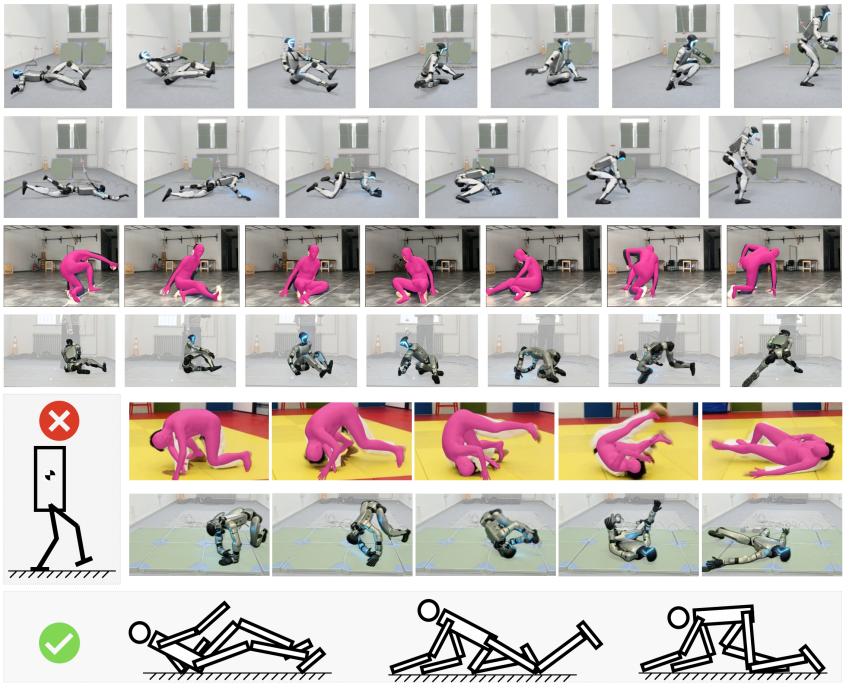


Figure 1: We present a unified humanoid motion interface and a zero-shot sim-to-real reinforcement learning framework, so that humanoid robots can successfully perform extreme contact-agnostic motion in the real world.

Abstract: Previous humanoid robot research works treat the robot as a bipedal mobile manipulation platform, where only the feet and hands contact the environment. However, we humans use all body parts to interact with the world, *e.g.*, we sit in chairs, get up from the ground, or roll on the floor. Contacting the environment using body parts other than feet and hands brings significant challenges in both model-predictive control and reinforcement learning-based methods: an unpredictable contact sequence makes it almost impossible for model-predictive control to plan ahead in real time; the success of sim-to-real reinforcement learning for humanoids heavily depends on the acceleration of the rigid-body physical simulator and the simplification of collision detection. On the other hand, lacking extreme torso movement of humanoid data makes all other components non-trivial to design, such as dataset distribution, motion commands, and task rewards. To address these challenges, we propose a general humanoid motion framework that takes discrete motion commands and controls the robot's motor actions in real time. Using a GPU-accelerated simulator, we train a humanoid whole-body control policy that follows the high-level motion command in the real world in real time, even with stochastic contacts and extremely large robot base rotation and not-so-feasible motion commands.

9th Conference on Robot Learning (CoRL 2025), Seoul, Korea.

1 Introduction

Humans do not only walk and manipulate objects—they sit, lie down, get up from the ground, and transition between various postures. However, contemporary humanoid research largely defines humanoid robots as bipedal mobile manipulation platforms, focusing on walking or dexterous hand use while neglecting the full range of human motions. This limited perspective overlooks the advantages of humanoid morphology in whole-body control. This work enables humanoid robots to execute motions that are difficult or impossible for other morphologies, such as getting up from lying down, recovering from a prone position, or transitioning through kneeling postures.

Achieving such a set of general humanoid motions presents significant challenges: (1) designing an effective motion command interface that supports a wide range of humanoid postures, and (2) constructing a general pipeline that trains a policy in simulation and deploys it in the real world.

1.1 Challenges in Motion Command Interface

Previous locomotion controllers define motion commands in terms of horizontal movement, typically using velocity commands (e.g., forward-lateral speed, heading angular velocity) or waypoint sequences that guide the robot over short time horizons. For example, when a quadruped robot stands from a crawling pose, the linear velocity command switches from the positive x-axis of the base frame to the negative z-axis of the base frame [1]. However, such representations become ambiguous when the robot undergoes significant roll or pitch rotations. When commanding a humanoid to switch between standing and crawling, defining movement as linear velocity in the base frame becomes impractical, requiring extensive manual coding.

Unlike locomotion commands [2, 3, 4, 5, 6, 7, 8], general motion commands lack a straightforward parameterization. The high-dimensional sampling space makes it difficult to generate feasible motions without extensive filtering. A more reasonable option is to use human motions as a reference, together with a re-targeting algorithm for a training humanoid policy. However, commonly used human motion datasets such as AMASS [9] predominantly contain standing postures, limiting their applicability to extreme motion scenarios. Additionally, kinematic differences between humans and humanoid robots cause many recorded motions to be infeasible. To address this issue, we construct an extreme-action dataset that includes motions with rich contact interactions.

1.2 Challenges in Producing a Deployable Humanoid Control Policy

Deploying complex humanoid motions on robot hardware faces multiple constraints. Most recent works on reinforcement learning (RL)-based locomotion [2, 3, 4, 5, 6, 7, 10, 11] rely on GPU-accelerated rigid-body simulation to train robust policies that can be transferred to the real world. However, they only focus on bipedal locomotion, assuming foot-only contact with the ground, where the upper-body interactions are typically ignored. To enable general contact interactions, determining the optimal level of collision simplification in simulation remains a non-trivial question.

Apart from reinforcement learning, model-based control methods also simplify the robot model when planning the contact sequences. Khatib et al. [12], Chignoli et al. [13], Nelson et al. [14] assume only feet are contacting the ground and optimize the control function with a manually built kinematic model. Even for quadruped locomotion, motion planners divide movement into swing and contact phases [15], significantly constraining possible motions. Introducing additional future contacts brings whole-body motion planning to the next level of difficulty, not to mention optimizing a collection of perception, state estimation, planning, and control systems that run accurately and in real time.

In this work, we propose a general framework that enables humanoid robots to execute a broad spectrum of extreme motions. Specifically, we express all motion targets in the base frame of the robot, no matter whether the robot is standing or lying down. We use a keyframe-based method to express future motion targets, so that the robot receives information about future motion expectations. To fully investigate the possibility of training a humanoid control policy that handles extreme motions, we curate an extreme-action dataset where the torsos are in extreme roll and pitch orientations. With

this training pipeline, we show that using simplified collision estimation and domain randomization techniques in the physical simulator, the control policy can be successfully deployed in the real world to follow real-time target motion commands.

Apart from the novel problem setting, dataset, and the whole pipeline design, we identify three additional technical contributions in this work:
• Designing a transformer-based motion encoder with key-frame-based future motion command.
• Using advantage mixing to bridge the gap between sparse motion reward and dense regularization reward.
• Verifying that removing the short-term odometry is tolerable for humanoid motion command, even for general motions with base movement.

2 Related Works

Humanoid Whole-Body Control Whole-body control for robots with multiple parts is a long-standing challenging problem. For humanoid robots, whole-body control with only feet contacting the ground is already a state-of-the-art research topic, due to the number of multiple rigid bodies attached to the system. Traditional methods depend on modeling the dynamics of the entire humanoid skeleton [13, 16, 17, 18, 19, 20]. However, these methods significantly limit the possibility of contacting the environments with components other than feet and potentially lead to failure when the contacting sequence becomes unpredictable.

In learning-based methods, whole-body control is still a challenging topic. Most of them can be viewed as a combination of lower-body and upper-body [21, 22, 4, 6]. More integrated humanoid algorithms control all the joints on the humanoid robot, but they only involve the contacts of a fixed set of body positions [2, 3, 11, 7, 23, 6, 24, 25, 26, 27]. Unexpected contact sequence situations are still not being explicitly investigated.

General Motion Interface for Humanoid To design a unified interface for large models to control the humanoid robot without real-time requirements, whole-body control algorithms for humanoid robots design the interface depending on how the general motion is defined. For mobile manipulation tasks, the general interface can be described as a locomotion goal and upper-body joint position goal, such as [28, 2, 4]. Specifically, locomotion can also be defined as a short-range navigation task [29, 30, 31]. However, all these interfaces intrinsically limit the potential of more complex behaviors, resulting in less flexibility to meet the fine-detailed motion targets.

In humanoid motion generation research, Peng et al. [32], Tessler et al. [33], Luo et al. [34] defines all joint orientation and base position sequences as the interface. The robot has to follow the motion target at each timestep, leading to less tolerance when the motion target is not physically feasible for the current robot model. Although lots of works apply them in the real world [23, 22, 25, 35, 36, 24, 37, 23, 25], they still use global odometry to provide feedback to the policy, which is expensive to acquire and deploy in the wild. To what extent the accumulated error in odometry can be tolerated remains unclear.

Sim-to-Real for Legged Robots Due to the over-complication of search and conditioning of traditional model-based planning, reinforcement learning and training in simulation have been widely used recently in control for legged robots. By simplifying the collision shapes and using efficient GPU-accelerated rigid-body physics simulations, quadruped robots [38, 39, 40] and humanoid robots [3, 41, 42, 4, 23, 10, 43] can perform various extremely difficult tasks, such as walking through rough terrain [44, 45, 29], overcoming extreme challenging obstacles [46, 28, 47, 2, 7]. Although humanoid motion with multiple contact points is being researched [24, 22, 26, 27], these motions involve fairly predictable contact sequences.

In this work, our aim is to investigate motions with stochastic and even unpredictable contact sequences and show that it is possible to train using simplified collision shapes while successfully deploying the policy on the real robot.

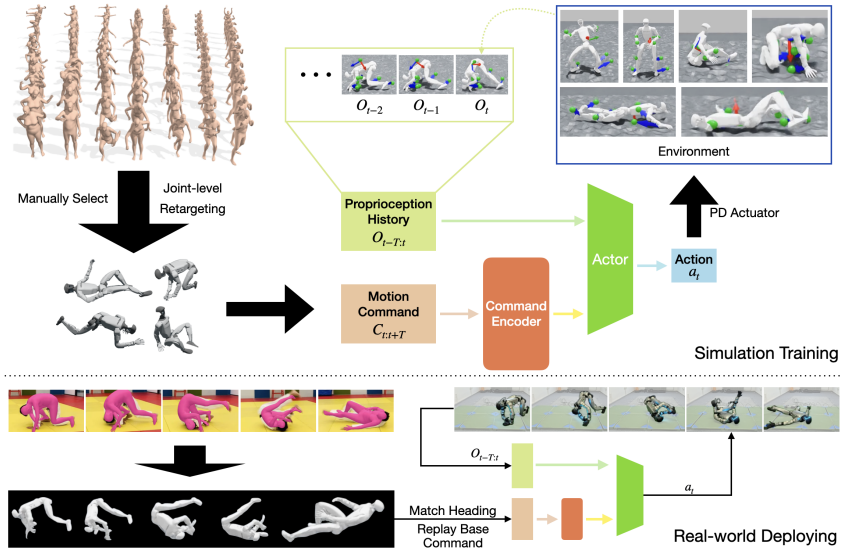


Figure 2: Overview of the framework. We curate an extreme-action dataset from the AMASS dataset and internet videos. We re-target the human motion to the joint-level target of the humanoid robot. We then feed the motion command as a streaming sequence to a Transformer-based encoder. Concatenated with a stack of history proprioceptive observation, an MLP actor network outputs the joint-level actions. A PD controller finally computes the torque for each joint motor.

3 Methods and Implementations

3.1 Pipeline Overview

We define the task as learning a humanoid control policy that reaches the target motion in a specified future time. Figure 2 gives an overview of the framework. First, we curate an extreme-action human motion dataset in the SMPL format for use as motion reference. We then re-target the human motion reference into the humanoid robot frame and use it as the motion command input of the policy network. The policy network consists of a command encoder and an actor. The command encoder consumes the motion command, which is represented in a sequence of concatenated joint positions, target link positions, and target base transform under the base when the motion target is refreshed, combined with the time passed from the motion target refreshing and the time to the specific frame. The actor takes the encoded feature, together with a stack of proprioception history as input, and outputs expected joint-level actions. Finally, a PD controller computes the torque of each joint motor. We train the whole-body control policy to reach the target motion positions as closely as possible in specific frames.

3.2 Addressing the Data Challenges

Curating diverse and unpredictable motion references Human motions serve as an important reference for humanoid robots. We start by sampling motion commands from an existing human motion dataset, AMASS [9]. However, most of the trajectories in AMASS are collected when the subjects are standing instead of sitting or lying on the ground. So, there are not many contact-rich motions presented in the dataset. We provide statistics and a histogram in the Supplement. Therefore, we build an extreme-action dataset, which is a combination of extreme motions from the AMASS dataset and extreme human motions extracted from Internet videos using 4D-Human [48].

Dealing with physically infeasible motion commands While sampling from humanoid motion commands from a pre-collected human motion capture dataset is a good way to start the training pipeline, the motion trajectories from the dataset still suffer from inconsistencies after rescaling the size between the human subject and the real humanoid robot. In Figure 3, the height trajectory of the

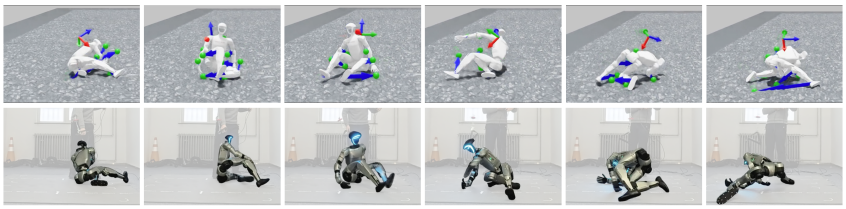


Figure 3: We show a challenging task where the robot is performing a breaking dance move on the ground. The hip, knee, thigh, elbow, and hands are contacting the ground, expected or unexpectedly. Note that the rubber hands on the robot cannot be simulated in the simulator, so we use the rigid-body collision shape by making a convex hull of the hands' mesh. The re-targeted base-pose reference is physically infeasible, as they are floating over the ground.

motion reference is inconsistent when the motion reference is in the standing position. To address this issue, we set a loose termination condition while training the policy in the simulator. Specifically, we terminate an episode only when the robot is far from the reference trajectory over 0.5m or the base rotation to the target rotation is greater than 1.0 rads.

3.3 Network Design and Model Training

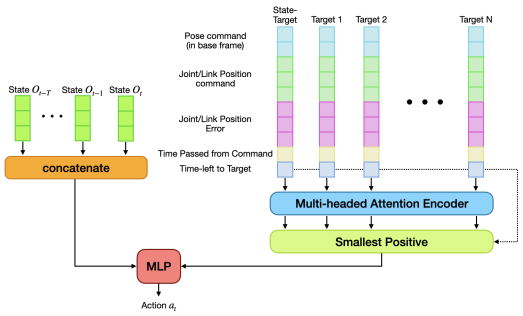


Figure 4: To handle variable lengths of input motion commands, the command encoder adopts the form of a Transformer encoder. We select the embedding whose source has the smallest positive t_{left}. Then, we concatenate the embedding with the stack of history proprioception and feed it to the actor MLP to acquire the action output.

Based on our extreme-action dataset, we re-target human motion data in the SMPL format to base position $\hat{p}_w$ and base axis-angle $\hat{\alpha}_w$ in the world frame and the motor joint positions $\hat{\theta}^i$ of the humanoid robot. For each motion command frame, we get the robot base position p_w and the base axis-angle α_w. We compute the motion target of the robot base ($\tilde{p}_b$ and $\tilde{\alpha}_b$) under the current robot base frame. We then use forward kinematics to compute the target link positions $\hat{l}_b^j$ under the robot's base frame. In summary, for each motion target frame, we concatenate all motion targets $[\tilde{p}_b, \tilde{\alpha_b}, \hat{\theta}^i, \hat{l}_b^j, (\hat{\theta}^i - \theta^i), (\hat{l}_b^j - l_b^j), t_{\text{passed}}, t_{\text{left}}]$. t_{passed} and t_{left} denote the time from the refresh of the motion reference and the time left to this motion target frame, respectively. Considering that link position errors and joint position errors are both in the local (robot's base) frame, we refresh these quantities in real-time in both simulation and real-world deployment.

To adapt to the needs of enabling an arbitrary number of motion targets being fed into the network, we use a Transformer-based motion reference encoder as shown in Figure 4. We select the frame that has the smallest positive t_{left} from the Transformer encoder's embedding as the input to the actor. Specifically, we concatenate the latent embedding with a stack of history proprioception and feed it into the actor MLP to obtain the joint action output.

To train the humanoid motion based on the motion command sequence, while leaving enough flexibility to control policy, we compute the motion target reward only when the frame of the motion target is expected to reach, $t_{\text{left}} == 0$. Therefore, some regularization reward terms must be used. For example the action rate, joint acceleration, energy, joint position out-of-limits, etc. However, these regularization terms are dense while the motion target reward is sparse. The robot's action spikes when the expected motion target is reached if all reward terms are combined. In this case, we use multiple critic networks and perform the advantage mixing technique [49] in addition to the PPO algorithm [50].

3.4 Advantage Mixing for Sparse Task Rewards

Different from conventional actor-critic architecture, we use one actor network as the policy and 3 critic networks for 3 different groups of rewards $\left(r^{(1)}, r^{(2)}, r^{(3)}\right)$. Each reward groups have multiple reward terms. Following Martinez-Piazuelo et al. [49], each critic network $V_{\Psi^{(i)}}(s_t)$ is supervised independently by their reward group $r^{(i)}$ with temporal difference error,

$$\mathcal{L}\left(\Psi^{(i)}\right) = \hat{\mathbb{E}}_t \left[\| r_t^{(1)} + \gamma V_{\Psi^{(i)}}(s_{t+1}) - V_{\Psi^{(i)}}(s_t) \|^2 \right] \tag{1}$$

where $\hat{\mathbb{E}}$ is the empirical average and γ is the discount factor. For the policy gradient part, the advantages are combined by weighted average after the general advantage estimation [51],

$$\tilde{A} = \sum_{i=0}^{n} w_i \frac{A_i - \mu_{A_i}}{\sigma_{A_i}} \tag{2}$$

where $\{A_i\}_{i=0}^n$ are advantages estimated by the 3 individual critic networks. μ_{A_i} and σ_{A_i} are the batch-wise statistics from each individual reward groups.

3.5 Real-world Deployment

One of the key statements of this work is to train a deployable sim-to-real controller to perform contact-agnostic motion. We deploy the policy onboard to test the entire pipeline. Our system is designed as a low-level controller commanded by a high-level motion generator. As shown in Figure 5, we use another laptop to record the testing data and visualize the robot motion. The low-level controller policy is running on the Nvidia Jetson Orin inside the Unitree G1's torso. We use ROS2 messages to communicate with the hardware onboard. We use ONNX [52] to accelerate the policy network.

Before deploying in the real world, we play the well-trained low-level control policy in the simulator and record the target joint positions, target link positions, and target base positions into a rosbag file. Then we replay these commands in the real world in an open-loop manner.

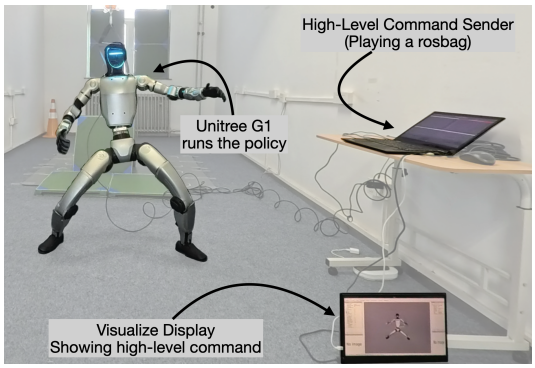


Figure 5: We run the control policy (including the Transformer-based encoder) using Nvidia Jetson NX inside of Unitree G1. We use an additional laptop to serve as another hardware that sends high-level motion commands, which are visualized in the bottom right of the figure. No motion capture system is used.

4 Experiments

We set up 3 experiments to verify the effectiveness of the designs and components of our method. We run quantitative results in the simulator. First, we show the effectiveness of the multi-critic, a.k.a advantage mixing technique, through a specific case study of the robot behavior. Then, we explain through the robot's behavior why we have to train using an extreme-action dataset to address the significance of training a sim-to-real-possible motion in the simulator.

To verify the successful deployment of our proposed pipeline, we build the extreme-action dataset with 3 types of motions. 2 of them are some extremely difficult motions because they introduce unexpected contacts between robot parts and the ground.

Getting up from the ground The first type of motion is getting up from the ground, such as subject 140 CMU mocap [53] and subject 3 in KIT [54].

	Deployable	Task		Smoothness	
		Success↑	mpjpe(m)↓	Action Jitter↓	Max Joint Acc(rad/s^2)↓
1. Getting up from the ground					
H2O [23]	✗	0%	0.48	8.23	548
w/o selected motions	✗	5.75%	0.42	5.25	357
w/o multi-critic	✓	83.50%	0.18	0.83	402
Embrace-Contacts	✓	**94.30%**	**0.21**	**0.61**	**86**
2. Ground interactions					
H2O [23]	✗	0%	0.46	9.10	601
w/o selected motions	✗	25.25%	0.38	6.91	483
w/o multi-critic	✓	92.50%	0.12	1.42	452
Embrace-Contacts	✓	**98.25%**	**0.16**	**0.63**	**120**
3. Standing dance					
H2O [23]	✓	**100%**	0.06	0.57	20
w/o selected motions	✓	98.50%	0.12	0.61	44
w/o multi-critic	✓	89.25%	0.07	0.62	36
Embrace-Contacts	✓	97.75%	0.09	0.56	24

Table 1: Comparison with the humanoid motion tracking baseline and ablations. In "w/o selected motions" option, we train the policy on the entire AMASS dataset. In "w/o multi-critic" option, we train the policy on the selected motion dataset but sum all the reward terms from the multi-critic setting to form a single reward.

Ground interactions Another type of motion is ground interaction motion. We collect the motion example from internet videos and track the human motion using 4D-Human [48]. We then retarget these human motion in SMPL format to the joint positions of the Unitree G1 robot.

Standing dance The last type of motion is mostly standing and dancing. We also test our pipeline by tracking a sequence of human motions from internet video using 4D-Human [48] and re-targeting the motion to the humanoid robot and playing in the simulator.

We test the success rate of these 3 types of motions by generating 1000 robots with uniformly sampled domain randomization parameters and run 10 times in the simulation. We count the number of trajectories being run and the number of trajectories being terminated because of the failure.

4.1 Importance of using an Extreme-Action dataset

Here we explain through a specific case study why we only select a handful of motions to train and achieve these extremely difficult motions in the real humanoid robot. As shown in Figure 6, if we train on the entire AMASS dataset, the lowest position the robot can reach is squatting. The data bias leads to the "safer" motion for the policy, which refuses to contact the ground using its hands. Since the main focus of this work is to train a deployable policy that performs extreme humanoid motion in the real world, generalization ability is not the primary concern. We trained on an extreme-action dataset, which is described in supplementary in detail. Also, shown in Table 1, the policy trained on the full dataset only masters the skills when the robot is standing. However, our method presents generalization ability when training on standing motions, even though we aim at training full-body ground interaction motions. The detailed humanoid behavior will be presented in the supplementary video.

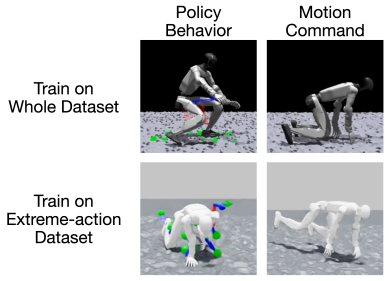


Figure 6: The comparison between to policy behaviors of crawling from a standing pose, using the AMASS dataset and the extreme-action dataset.

4.2 Effectiveness of Multi-Critic

We verify the effectiveness of training RL with multi-critic. In implementing the single-critic technique, we sum all rewards together with the same weights as for the multi-critic setting. Shown in Table 1, training without multi-critic leads to higher joint jitter and lower performance in massive tests in simulation. We hypothesize that the weights for the discrete task reward are hard to tune, which makes the policy focus more on meeting the motion target rather than the smoothness.

4.3 Importance of Adding Future Motion as Command

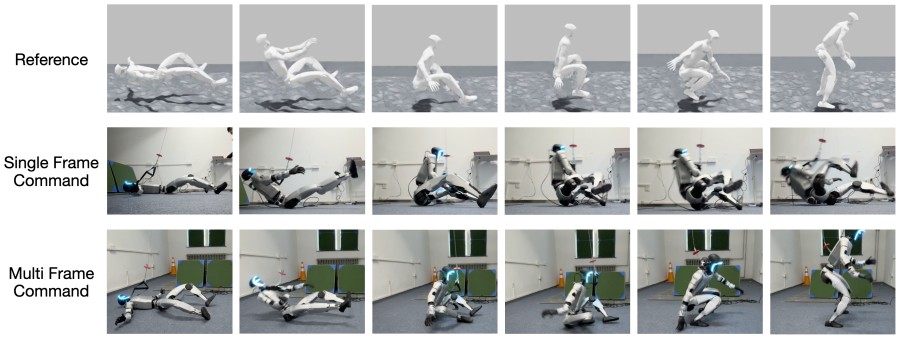


Figure 7: The effectiveness of feeding a sequence of motion commands rather than a single frame of motion command.

In Section 3.3, we choose to use a sequence of motion commands as the general command input instead of the command history as Fu et al. [22]. For these full-body contact motions, the motion reference retargeted on the humanoid morphology deviates from the robot's kinematics too much, as stated in Section 1.2. Shown in Figure 7, when the policy receives only one future motion command, it consistently maintains tracking accuracy but fails to achieve the final goal, such as standing up.

5 Conclusion

In this work, we embrace full-body contacts that have rarely been discussed in recent humanoid research. We overcome the exponential searching issue in model-based control for humanoid robots using zero-shot sim-to-real reinforcement learning. We propose a general motion command for humanoids so that locomotion, manipulation and whole-body control tasks can be unified in a single interface. Based on this motion command, we adopt a transformer-based encoder to process the command input with a variable input length. By diving deep into the motions where humanoids contact with the environments with components not limited to hands and feet, we show the potential of training in simulation using reinforcement learning can make such complex and extremely difficult motion realizable in the real world, even if the motion command is not physically feasible for the given robot model.

Limitations: Even though this work shows the potential of training complex motions only in a simulator and making them possible in the real world, training a low-level controller that performs full humanoid motion still requires a high-quality motion dataset that is not only limited to standing motions. Or we need a way of bridging the gap between robot models and human subjects among these collected human motion datasets. For high-level commands, for example, large action models, they do not consider leg motions. We need to build an abstraction such as masking on the lower legs command to make the entire general humanoid motion system possible in the future. Addressing these limitations and training a general humanoid whole-body control system that allows a high-level large action model to reason and send general motion commands to the real humanoid robots will be our future work.

References

[1] Y. Li, J. Li, W. Fu, and Y. Wu. Learning agile bipedal motions on a quadrupedal robot. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9735–9742, 2024. doi:10.1109/ICRA57147.2024.10611442.

[2] Z. Zhuang, S. Yao, and H. Zhao. Humanoid parkour learning. In *8th Annual Conference on Robot Learning*, 2024. URL https://openreview.net/forum?id=fs7ia3FqUM.

[3] I. Radosavovic, B. Zhang, B. Shi, J. Rajasegaran, S. Kamat, T. Darrell, K. Sreenath, and J. Malik. Humanoid locomotion as next token prediction. *CoRR*, abs/2402.19469, 2024. URL https://doi.org/10.48550/arXiv.2402.19469.

[4] X. Cheng, Y. Ji, J. Chen, R. Yang, G. Yang, and X. Wang. Expressive whole-body control for humanoid robots. *arXiv preprint arXiv:2402.16796*, 2024.

[5] B. van Marum, A. Shrestha, H. Duan, P. Dugar, J. Dao, and A. Fern. Revisiting reward design and evaluation for robust human standing and walking. *(Under Submission)*, 2024.

[6] X. Gu, Y.-J. Wang, and J. Chen. Humanoid-gym: Reinforcement learning for humanoid robot with zero-shot sim2real transfer. *arXiv preprint arXiv:2404.05695*, 2024.

[7] J. Long, J. Ren, M. Shi, Z. Wang, T. Huang, P. Luo, and J. Pang. Learning humanoid locomotion with perceptive internal model, 2024. URL https://arxiv.org/abs/2411.14386.

[8] G. B. Margolis and P. Agrawal. Walk these ways: Tuning robot control for generalization with multiplicity of behavior. *Conference on Robot Learning*, 2022.

[9] 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*, pages 5442–5451, Oct. 2019.

[10] Q. Zhang, P. Cui, D. Yan, J. Sun, Y. Duan, A. Zhang, and R. Xu. Whole-body humanoid robot locomotion with human reference, 2024.

[11] A. Serifi, R. Grandia, E. Knoop, M. Gross, and M. Bächer. Vmp: Versatile motion priors for robustly tracking motion on physical characters. In *SCA '24: ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '24, page 1–11, Goslar, DEU, 2024. Eurographics Association. doi:10.1111/cgf.15175. URL https://doi.org/10.1111/cgf.15175.

[12] O. Khatib, L. Sentis, and J. Park. A unified framework for whole-body humanoid robot control with multiple constraints and contacts. In *European Robotics Symposium*, 2008. URL https://api.semanticscholar.org/CorpusID:8117859.

[13] M. Chignoli, D. Kim, E. Stanger-Jones, and S. Kim. The mit humanoid robot: Design, motion planning, and control for acrobatic behaviors. In *2020 IEEE-RAS 20th International Conference on Humanoid Robots (Humanoids)*, pages 1–8, 2021. doi:10.1109/HUMANOIDS47582.2021.9555782.

[14] G. Nelson, A. Saunders, and R. Playter. *The PETMAN and Atlas Robots at Boston Dynamics*, pages 169–186. Springer Netherlands, Dordrecht, 2019. ISBN 978-94-007-6046-2. doi:10.1007/978-94-007-6046-2_15. URL https://doi.org/10.1007/978-94-007-6046-2_15.

[15] G. Bledt, M. J. Powell, B. Katz, J. Di Carlo, P. M. Wensing, and S. Kim. Mit cheetah 3: Design and control of a robust, dynamic quadruped robot. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2245–2252, 2018. doi:10.1109/IROS.2018.8593885.

[16] B. Dariush, M. Gienger, B. Jian, C. Goerick, and K. Fujimura. Whole body humanoid control from human motion descriptors. In *2008 IEEE International Conference on Robotics and Automation, ICRA 2008, May 19-23, 2008, Pasadena, California, USA*, pages 2677–2684. IEEE, 2008. doi:10.1109/ROBOT.2008.4543616. URL https://doi.org/10.1109/ROBOT.2008.4543616.

[17] J. Grizzle, J. Hurst, B. Morris, H.-W. Park, and K. Sreenath. Mabel, a new robotic bipedal walker and runner. In *2009 American Control Conference*, pages 2030–2036, 2009. doi:10.1109/ACC.2009.5160550.

[18] F. L. Moro and L. Sentis. *Whole-Body Control of Humanoid Robots*, pages 1161–1183. Springer Netherlands, Dordrecht, 2019. ISBN 978-94-007-6046-2. doi:10.1007/978-94-007-6046-2_51. URL https://doi.org/10.1007/978-94-007-6046-2_51.

[19] A. Dallard, M. Benallegue, F. Kanehiro, and A. Kheddar. Synchronized human-humanoid motion imitation. *IEEE Robotics and Automation Letters*, 8(7):4155–4162, 2023. doi:10.1109/LRA.2023.3280807.

[20] S. Kajita, F. Kanehiro, K. Kaneko, K. Yokoi, and H. Hirukawa. The 3d linear inverted pendulum mode: a simple modeling for a biped walking pattern generation. In *Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the the Next Millennium (Cat. No.01CH37180)*, volume 1, pages 239–246 vol.1, 2001. doi:10.1109/IROS.2001.973365.

[21] C. Lu, X. Cheng, J. Li, S. Yang, M. Ji, C. Yuan, G. Yang, S. Yi, and X. Wang. Mobile-television: Predictive motion priors for humanoid whole-body control. *arXiv preprint arXiv:2412.07773*, 2024.

[22] Z. Fu, Q. Zhao, Q. Wu, G. Wetzstein, and C. Finn. Humanplus: Humanoid shadowing and imitation from humans. In *Conference on Robot Learning (CoRL)*, 2024.

[23] T. He, Z. Luo, W. Xiao, C. Zhang, K. Kitani, C. Liu, and G. Shi. Learning human-to-humanoid real-time whole-body teleoperation. In *arXiv*, 2024.

[24] C. Zhang, W. Xiao, T. He, and G. Shi. Wococo: Learning whole-body humanoid control with sequential contacts. In *8th Annual Conference on Robot Learning*, 2024. URL https://openreview.net/forum?id=Czs2xH9114.

[25] T. He, Z. Luo, X. He, W. Xiao, C. Zhang, W. Zhang, K. Kitani, C. Liu, and G. Shi. Omnih2o: Universal and dexterous human-to-humanoid whole-body teleoperation and learning. *arXiv preprint arXiv:2406.08858*, 2024.

[26] X. He, R. Dong, Z. Chen, and S. Gupta. Learning getting-up policies for real-world humanoid robots. *arXiv preprint arXiv:2502.12152*, 2025.

[27] T. Huang, J. Ren, H. Wang, Z. Wang, Q. Ben, M. Wen, X. Chen, J. Li, and J. Pang. Learning humanoid standing-up control across diverse postures, 2025. URL https://arxiv.org/abs/2502.08378.

[28] Z. Zhuang, Z. Fu, J. Wang, C. G. Atkeson, S. Schwertfeger, C. Finn, and H. Zhao. Robot parkour learning. In *Conference on Robot Learning CoRL*, 2023.

[29] J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter. Learning quadrupedal locomotion over challenging terrain. *Science Robotics*, Oct. 2020.

[30] B. Yang, L. Wellhausen, T. Miki, M. Liu, and M. Hutter. Real-time optimal navigation planning using learned motion costs. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021.

[31] T. Miki, J. Lee, L. Wellhausen, and M. Hutter. Learning to walk in confined spaces using 3d representation. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 8649–8656, 2024. doi:10.1109/ICRA57147.2024.10610271.

[32] 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.*, 37(4):143:1–143:14, July 2018. ISSN 0730-0301. doi:10.1145/3197517.3201311. URL http://doi.acm.org/10.1145/3197517.3201311.

[33] C. Tessler, Y. Kasten, Y. Guo, S. Mannor, G. Chechik, and X. B. Peng. Calm: Conditional adversarial latent models for directable virtual characters. In *ACM SIGGRAPH 2023 Conference Proceedings*, SIGGRAPH '23, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9798400701597. doi:10.1145/3588432.3591541. URL https://doi.org/10.1145/3588432.3591541.

[34] Z. Luo, J. Cao, A. W. Winkler, K. Kitani, and W. Xu. Perpetual humanoid control for real-time simulated avatars. In *International Conference on Computer Vision (ICCV)*, 2023.

[35] F. Zargarbashi, J. Cheng, D. Kang, R. Sumner, and S. Coros. Robotkeyframing: Learning locomotion with high-level objectives via mixture of dense and sparse rewards. In *8th Annual Conference on Robot Learning*, 2024. URL https://openreview.net/forum?id=wcbrhPnOei.

[36] C. Tessler, Y. Guo, O. Nabati, G. Chechik, and X. B. Peng. Maskedmimic: Unified physics-based character control through masked motion inpainting. *ACM Transactions on Graphics (TOG)*, 2024.

[37] S. Liu, L. Wu, B. Li, H. Tan, H. Chen, Z. Wang, K. Xu, H. Su, and J. Zhu. Rdt-1b: a diffusion foundation model for bimanual manipulation. *arXiv preprint arXiv:2410.07864*, 2024.

[38] 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*, 2021.

[39] A. Kumar, Z. Fu, D. Pathak, and J. Malik. Rma: Rapid motor adaptation for legged robots. 2021.

[40] A. Escontrela, X. B. Peng, W. Yu, T. Zhang, A. Iscen, K. Goldberg, and P. Abbeel. Adversarial motion priors make good substitutes for complex reward functions. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 25–32, 2022. doi:10.1109/IROS47612.2022.9981973.

[41] Q. Liao, B. Zhang, X. Huang, X. Huang, Z. Li, and K. Sreenath. Berkeley humanoid: A research platform for learning-based control, 2024.

[42] B. Xia, B. Li, J. Lee, M. Scutari, and B. Chen. The duke humanoid: Design and control for energy efficient bipedal locomotion using passive dynamics, 2024. URL https://arxiv.org/abs/2409.19795.

[43] G. A. Castillo, B. Weng, W. Zhang, and A. Hereid. Robust feedback motion policy design using reinforcement learning on a 3d digit bipedal robot. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5136–5143, 2021. doi:10.1109/IROS51168.2021.9636467.

[44] I. M. Aswin Nahrendra, B. Yu, and H. Myung. Dreamwaq: Learning robust quadrupedal locomotion with implicit terrain imagination via deep reinforcement learning. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5078–5084, 2023. doi:10.1109/ICRA48891.2023.10161144.

[45] A. Agarwal, A. Kumar, J. Malik, and D. Pathak. Legged locomotion in challenging terrains using egocentric vision. In *6th Annual Conference on Robot Learning*, 2022. URL https://openreview.net/forum?id=Re3NjSwf0WF.

[46] D. Hoeller, N. Rudin, D. Sako, and M. Hutter. Anymal parkour: Learning agile navigation for quadrupedal robots. *Science Robotics*, 9(88):eadi7566, 2024. doi:10.1126/scirobotics.adi7566. URL https://www.science.org/doi/abs/10.1126/scirobotics.adi7566.

[47] X. Cheng, K. Shi, A. Agarwal, and D. Pathak. Extreme parkour with legged robots. *arXiv preprint arXiv:2309.14341*, 2023.

[48] S. Goel, G. Pavlakos, J. Rajasegaran, A. Kanazawa, and J. Malik. Humans in 4D: Reconstructing and tracking humans with transformers. In *ICCV*, 2023.

[49] J. Martinez-Piazuelo, D. E. Ochoa, N. Quijano, and L. F. Giraldo. A multi-critic reinforcement learning method: An application to multi-tank water systems. *IEEE Access*, 8:173227–173238, 2020. doi:10.1109/ACCESS.2020.3025194.

[50] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal policy optimization algorithms. *CoRR*, abs/1707.06347, 2017. URL http://arxiv.org/abs/1707.06347.

[51] J. Schulman, P. Moritz, S. Levine, M. I. Jordan, and P. Abbeel. High-dimensional continuous control using generalized advantage estimation. *CoRR*, abs/1506.02438, 2015. URL https://api.semanticscholar.org/CorpusID:3075448.

[52] O. R. developers. Onnx runtime. https://onnxruntime.ai/, 2021.

[53] URL http://mocap.cs.cmu.edu/.

[54] C. Mandery, O. Terlemez, M. Do, N. Vahrenkamp, and T. Asfour. Unifying representations and large-scale whole-body motion databases for studying human motion. *IEEE Transactions on Robotics*, 32(4):796–809, 2016.

[55] S. Zhong, T. Power, A. Gupta, and P. Mitrano. PyTorch Kinematics, Feb. 2024.

6 Appendix

We attach a video describing the main idea and the real-world experiments in real-time with no accelerations. We perform real-world tests on all three types of motions. We encourage readers to see the video for a more comprehensive understanding of this work.

A More Results

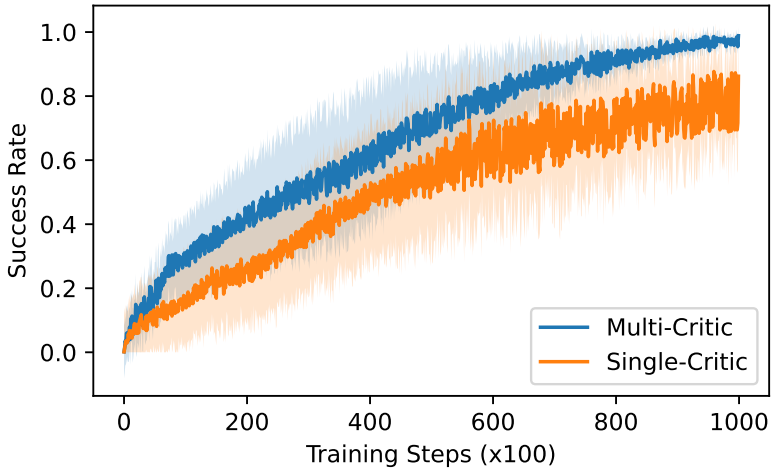


Figure 8: The success rate reported during training when trained on all 4 extreme difficult contact-agnostic motion.

Figure 8 shows the training progress comparing multi-critic technique and single-critic technique. Using multi-critic setting leads to faster training speed as well as faster convergence rate.

B Simulation Details

Considering these extreme difficult humanoid motion reference data may lead to physically infeasible situations, such as body parts penetrating the ground or robot base floating in the air, we set the robot's joint positions as the retarget robot pose in the first motion reference frame. We initialize the robot's base position by first adding a positive height to all motion reference frame so that no motion reference frame is penetrating the ground. Then we add a tiny height offset to spawn the robot, typically 0.06m. Also, to help the policy experience more states if the policy stuck as some place, for example if it does not get up from the ground, we sample the initial pose of the robot not only from the first frame of the motion reference trajectory, but from the start of the motion reference to the 60% of it.

During each rollout, we select one motion reference to generate the motion command. At time t, we use a pre-sampled time interval t_{int} and sample the motion reference at $t + t_{\text{int}}, t + 2t_{\text{int}}, t + 3t_{\text{int}}, t + 4t_{\text{int}}, t + 5t_{\text{int}}$ respectively. We also compute the base reference position and orientation in the base frame at time t.

C Training Details

As described in the main text, we select a handful of motion commands to train the humanoid whole-body controller to overcome the unbalanced issue of the precollected dataset. Shown in Table 3, we train our policy in simulation by extracting motion command from these motion references.

	Names	Value
Environments	Number of robots	4096
Domain randomization	Scaling body mass	0.8 ∼1.2
	Center of mass position	-0.02m ∼0.02m
	Scaling motor stiffness	0.9 ∼1.1
	Scaling motor damping	0.9 ∼1.1
	Motor delays	0.0s ∼0.03s
Initialize pose	Height offset	0.04m
	Sampling frame ratio from the trajectory	0.0 ∼0.6

Table 2: Detailed parameters for running the system in the simulator

Dataset	Subject	Motion names
CMU	140	Get up from ground
KIT	3	Crawling
Internet Video	Bilibili BV1L34y1t71x	Popping dance movement
Internet Video	BiliBili BV1Nm4y1k7wP	Breaking dance movement
Internet Video	Bilibili BV1mv411G7WM	Jiu-jitsu movement

Table 3: The motion reference we select to build tine motion command dataset.

Reward group	Reward term	Expression
Task	Base position tracking	$\Psi(\Delta(p_b), 0.4)$
	Base orientation tracking	$\Psi(\Delta(q_b), 0.8)$
	Joint position tracking	$\Psi(\|\theta^j - \hat{\theta}^j\|, 0.3)$
Regularization	Action rate	$\Psi(\|a_t^j - a_{t-1}^j\|, 1.0)$
	Joint acceleration	$\Psi(\|\ddot{\theta}^j\|, 500)$
	Joint velocity	$\Psi(\|\dot{\theta}^j\|, 15)$
Safety	Joint position limit	$\Psi(\max(\theta^j - \theta^j_{\text{max}}, \theta^j_{\text{min}} - \theta^j), 0.1)$
	Joint torque limit	$\Psi(\max(\|\tau^j\| - 0.9\,\tau^j_{\text{max}}, 0), 0.1)$

Table 4: Reward terms and their expressions

In Table 4, the function Ψ is a Gaussian kernel where,

$$\Psi(a, b) = \exp(-a/b^2) \tag{3}$$

Shown in Table 4, we build these reward functions in the range of $0\ 1$ so that everything is positive, potentially preventing active termination behavior. Then we **multiply** all reward terms in each reward group so that the algorithm will not completely ignore any of these terms. For the experiment variant using single critic, the reward terms within each group are multiplied and the reward groups are sum together weighted by the same weight parameters of the advantage mixing to get the scalar reward.

The policy network consist of actor and multiple critics with the same structure. We use a transformer-based encoder block to encoder all motion command. The encoder outputs a sequence of embedding, which we select the embedding whose 'time-to-target' attribute is the smallest positive value. We then concatenate this embedding with a stacked history proprioception observation and feed them to a Multi-Layer Perceptron. The MLP layers outputs the 29-dof action as the target position to the robot motors. Detailed parameters for the network are shown in Table 6.

We train our algorithm on a Nvidia 4090D GPU with 4096 robots in parallel for about 72 hours from scratch. We build the simulation environment using IsaacLab and modify the reinforcement learning framework based on rsl_rl.

D Deployment and Real-World Experiment Details

To run the trained policy on the real robot, we deploy the entire system on an Nvidia Jetson Orin NX and a laptop running Intel i5 CPU. We export the policy (including the transformer-based encoder) as

Hyperparameters	Value
Optimizer	AdamW
β_1, β_2	0.9, 0.999
Learning rate	$1e-4$
Batch size	4096
Clip param	0.2
Entropy coefficient	0
min_std clip	0.2
Desired KL	0.01
Maximum gradient norm	1
Num minibatches	4
γ	0.99
λ	0.95
Advantage mixing weights	[0.7, 0.1, 0.2]

Table 5: Parameters in Algorithm implementation

Hyperparameters	Value
Encoder Activation	GELU
Encoder Project Activation	ReLU
Encoder num heads	1
Encoder num layers	2
Encoder d_model	128
Encoder feedforward dimension	128
Encoder output size	128
MLP hidden sizes	[512, 256, 256]
MLP Activation	ELU

Table 6: The detailed network parameters for the low-level policy, which runs onboard

Joint name	**Stiffness (kp)**	**Damping (kd)**
Left/right shoulder pitch/roll/yaw	25	1.0
Left/right elbow	25	1.0
Left/right wrist roll	25	1.0
Left/right wrist pitch/yaw	5	0.5
Waist roll/pitch	60	2.5
Waist yaw	90	2.5
Left/right hip pitch/roll/yaw	90	2.0
Left/right knee	140	2.5
Left/right ankle pitch/roll	20	1.0

Table 7: Parameters that runs on the hardware

an ONNX program. All components communicate using ROS2 in the network of Unitree G1 robot. We then run the policy on the Jetson board at 50Hz. Since the policy outputs the action as the target joint position of each motor on the robot, we use the built-in PD controller on the Unitree G1 robot, which runs at 1000Hz, with the kp/kd setting as shown in Table 7. These kp/kd parameters are also the same when training in simulator.

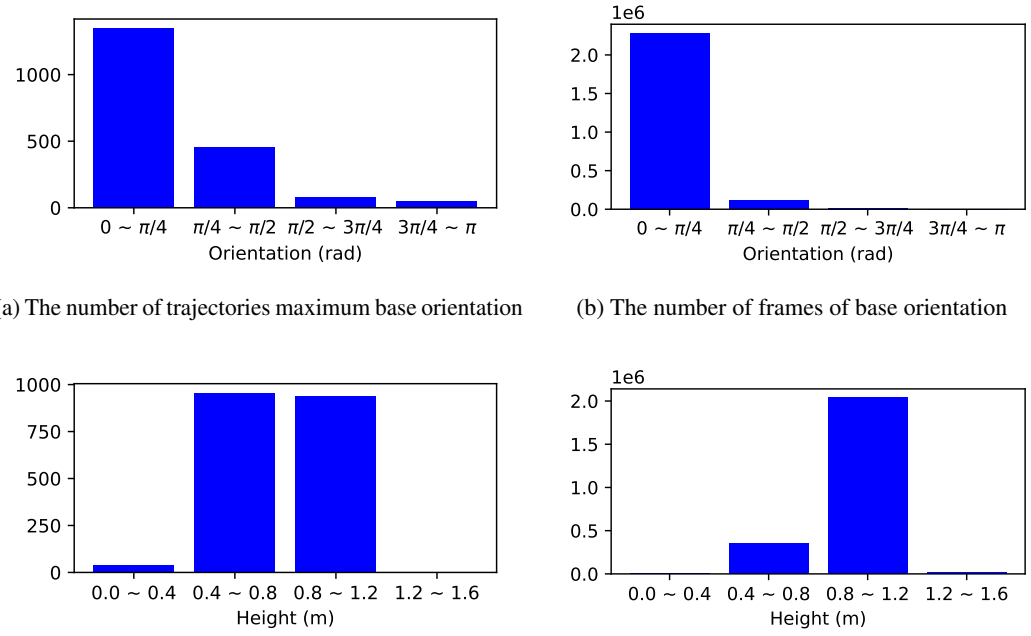
To acquire the target link position and their error respectively, we use Pytorch_Kinematics [55] and ONNX [52] to export the forward kinematics computation as an ONNX program. The exported ONNX program gets the joint positions and outputs the target link positions in the robot's base frame, which runs in real time on Nvidia Jetson Orin NX.

Since this work is also a proof-of-concept for building a hierarchical general humanoid controller, with a low-level whole-body control policy and a high-level command sender, we use another laptop to send the high-level command which simultaneously test the communication latency. Considering our high-level motion command is defined with base pose sequence under the robot frame when the command is generated, the high-level motion command for the real-world testing cannot be played

directly from SMPL-based motion file. We play each motion in simulation using the well-trained low-level policy and record the motion command, as well as the base pose command under the robot's base frame in simulation. We then play this base pose command in the real world and ignore the difference between the robot trajectory in the simulator and in the real world.

In the real-world testing, it is important to determine whether the testing motion succeeded, while we don't install additional motion capture system. For each extreme motion, we determine the success of each motion as finishing the entire motion command sequence with no unexpected head contacting the ground. For getting-up-from-ground task, we terminate the test when the robot's torso orientation significantly deviate from the motion command. In our real-world experiment, we also visualize the motion command in the laptop that sends the motion command sequence.

E Data distribution analysis



(a) The number of trajectories maximum base orientation

(b) The number of frames of base orientation

(c) The number of trajectories smallest base height

(d) The number of frames of base height range

Figure 9: Histogram of the number of motions in terms of their maximum roll/pitch and the minimum base height.

As shown in Figure 9, we count the number of frames and the number of files whose maximum base orientation and minimum base heights. Figure 9c counts the number of motion files whose minimum base height reaches a certain range. As shown in Figure 9a and Figure 9b, most of the motion files are performed when the base position is standing straight. As shown in Figure 9c and Figure 9d, most of the motions are performed when the robot base is over 0.4m. In this case, not many contact-rich motion is presented in the dataset.