GMT: General Motion Tracking for Humanoid Whole-Body Control

Zixuan Chen$*^{1,2}$ **Mazeyu Ji**$*^{1}$ **Xuxin Cheng**^{1} **Xuanbin Peng**^{1}
Xue Bin Peng$\dagger^{2}$ **Xiaolong Wang**$\dagger^{1}$
^{1}UC San Diego ^{2}Simon Fraser University $*$Equal Contribution $\dagger$Equal Advising

gmt-humanoid.github.io

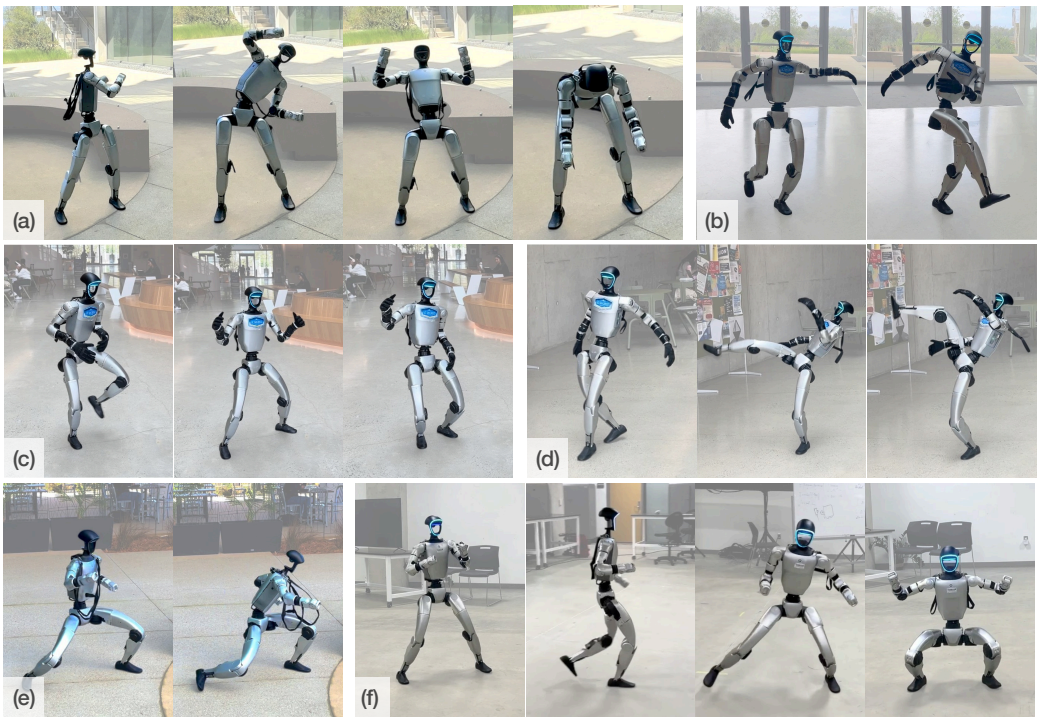


Figure 1: We deploy the *general* unified motion tracking policy on a medium-sized humanoid robot. GMT can perform a wide range of motion skills with good stability and generalizability, including **(a)** stretching, **(b)** kicking-ball, **(c)** dancing, **(d)** high kicking, **(e)** kungfu, and **(f)** other dynamic skills such as boxing, running, side stepping, and squatting.

Abstract: The ability to track general whole-body motions in the real world is a useful way to build general-purpose humanoid robots. However, achieving this can be challenging due to the temporal and kinematic diversity of the motions, the policy's capability, and the difficulty of coordination of the upper and lower bodies. To address these issues, we propose **GMT**, a general and scalable motion-tracking framework that trains a single unified policy to enable humanoid robots to track diverse motions in the real world. GMT is built upon two core components: an Adaptive Sampling strategy and a Motion Mixture-of-Experts (MoE) architecture. The Adaptive Sampling automatically balances easy and difficult motions during training. The MoE ensures better specialization of different regions of the motion manifold. We show through extensive experiments in both simulation and the real world the effectiveness of GMT, achieving state-of-the-art performance across a broad spectrum of motions using a *unified general* policy. Videos and additional information can be found at `gmt-humanoid.github.io`.

Keywords: Humanoid, Locomotion, Learning-based Control, Motion Imitation

1 Introduction

One of the primary goals for humanoid robots is to perform a wide range of tasks in everyday environments. Enabling humanoid robots to produce a broad repertoire of human-like movements is a promising approach towards achieving this objective. To generate such human-like movements, a general whole-body controller is required — one capable of leveraging a wide corpus of motor skills to perform both basic tasks, such as natural walking, and more dynamic and agile actions, such as kicking and running. With such a robust whole-body controller, it becomes possible to integrate a high-level planner that selects and sequences skills autonomously, paving the way towards general-purpose humanoid robots.

Manually designing such controllers can be challenging and labor-intensive due to the high degrees of freedom (DoFs) in humanoid systems and the complexity of real-world dynamics. Since humanoid robots are designed to closely resemble the human body, human motion data offers an ideal resource for equipping robots with a rich set of skills. In the field of computer graphics, researchers have leveraged human motion data and learning-based methods to develop a single unified controller for simulated characters that can reproduce a wide variety of motions and perform diverse skills with remarkably human-like behaviors [1, 2, 3, 4, 5].

Despite great progress in simulated domains, developing such a generic unified controller for humanoid robots can be challenging due to:

- **Partial Observability.** For real-world robots, full state information like linear velocities and global root positions is not accessible, which is quite important when learning motion tracking policies. The absence of this information makes the training process more challenging.
- **Hardware Limitations.** Existing human motion datasets often include movements such as back-flipping and rolling, which are infeasible for humanoid robots to execute due to hardware limitations. Moreover, even for more basic skills like walking and running, robots may struggle to produce enough torque to accurately match the speed and dynamics of human motion. These mismatches require extra special handling during the training of motion tracking controllers for robots.
- **Unbalanced Data Distribution.** Large mocap datasets like AMASS [6] often exhibit highly unbalanced distributions, as shown in Fig. 2, with a large proportion of motions involving walking or in-place activities. The scarcity of more complex or dynamic motions can cause the robot to struggle in mastering these less frequent but critical skills.
- **Model Expressiveness.** While a simple MLP-based network may be sufficient to develop a satisfying control policy when tracking a few motion clips, it often struggles when applied to large Mocap datasets. Such architectures typically lack the capacity to capture complex temporal dependencies and to distinguish between diverse motion categories. This limited expressiveness can result in sub-optimal tracking performance and poor generalization across a wide range of skills.

Although existing works have tried to solve some individual issues mentioned above, for example, teacher-student training framework is used to handle partial observability [7, 8]; several categories of small datasets are used to fine-tune separate specialist policies [7]; and a transformer model is used to increase the expressiveness of the model [9], developing a unified general motion tracking controller remains an open problem. This paper demonstrates that by addressing these issues jointly along with some other careful design decisions, we are able to create an effective system for training general motion tracking controllers for real humanoid robots. A brief comparison of our work with highly related existing works is shown in table 1. We measure if a policy is *diverse* based on its ability to perform a broad range of both upper-body and lower-body skills, including variations in walking styles, crouching, kicking, and other expressive whole-body movements.

In this paper, we propose GMT, a general and effective framework that trains a single unified, high-quality motion tracking policy for real-world humanoid robots from large mocap datasets. Central to GMT are two key innovations: a novel *Adaptive Sampling* strategy, designed to mitigate issues of struggling to learn less frequent motions arising from uneven motion category distributions, and a motion *Mixture-of-Experts (MoE)* architecture to enhance model expressiveness and generalizability. Fig. 1 showcases the real-world deployment of our policy, demonstrating its ability to reproduce a wide range of basic locomotion and agile skills with high fidelity, achieving state-of-the-art performance using a *single* unified policy. For more details, please refer to our project website.

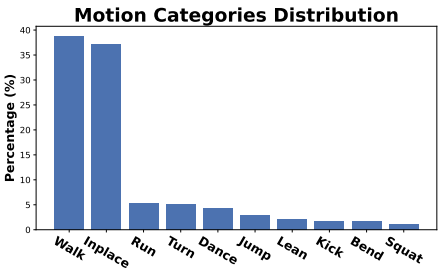


Figure 2: Distribution of motion categories in the AMASS dataset. The figure shows the proportion of the total motion duration corresponding to each category.

Table 1: Comparison of closely related works.

Method	Single Policy	Whole-Body	Diverse	Dataset Size
ExBody [10]	✓	✗	✗	780 clips
ExBody2 [7]	✓	✓	✓	1050 clips(filtered)
HumanPlus [9]	✓	✓	✗	10k clips
OmniH2O [8]	✓	✓	✗	14k clips (augmented)
ASAP [11]	✗	✓	✓	– (internet clips)
GMT (Ours)	✓	✓	✓	8925 clips (filtered)

2 Related Works

Developing a general whole-body controller that enables humanoid robots to perform a wide range of skills has long been a fundamental yet challenging problem, primarily due to the systems' high dimensionality and inherent instability. Traditional model-based methods have achieved robust whole-body locomotion controllers for bipeds and humanoids with gait planning and dynamics modeling [12, 13, 14]. However, designing such controllers can be labor-intensive and requires careful handling of complex dynamics. In recent years, learning-based approaches have made significant progress in building whole-body controllers. They either develop the controller with careful hand-designed task rewards [15, 16, 17, 18, 19, 20, 21], or with human motions as reference [11, 8, 9, 7, 10, 22, 23].

2.1 Learning-based Humanoid Whole-Body Control

This work focuses on whole-body control for humanoid robots. Previous studies have demonstrated that whole-body control policies, trained with manually designed rewards, can enable humanoid robots to perform locomotion skills such as walking [15, 16, 17, 18, 19], jumping [24, 25, 26], and fall-recovery behaviors [27, 28]. However, these controllers are typically task-specific, requiring training separate policies with customized reward functions for each task. For instance, policies developed for walking are not easily transferable to tasks such as jumping or manipulation. In contrast, human motion data offers a promising way to develop general purpose controllers without having to design task-specific reward functions for each skill we want the agent to perform.

2.2 Humanoid Motion Imitation

Leveraging human motion data to develop human-like whole-body behaviors for robots has been extensively studied in character animation. Previous works have achieved high-quality and general motion tracking on simulated characters [1, 3, 5, 4], and diverse motion skills for various downstream

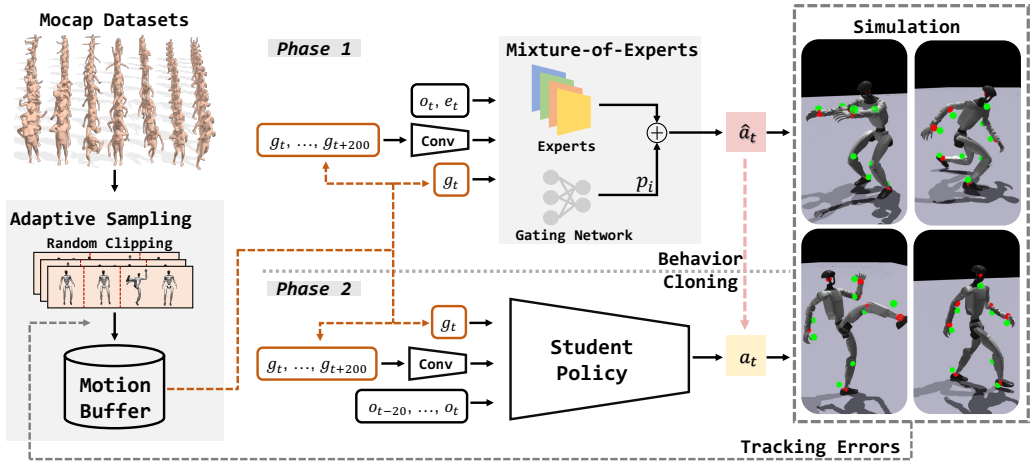


Figure 3: An overview of GMT. Here $\boldsymbol{g}_t$ denotes the motion target frame, $\boldsymbol{o}_t$ denotes proprioceptive observation, and $\boldsymbol{e}_t$ denotes privileged information.

tasks [29, 2, 30, 31, 32]. However, due to the partial observability in the real world, developing such whole-body controllers for real robots [10, 33, 9, 8, 7, 11, 34, 35, 36] can be challenging. For developing a unified general whole-body motion tracking controller, some works decouple upper-body and lower-body control to compromise between expressiveness and balance [10, 33]. HumanPlus [9] and OmniH2O [8] then successfully enabled whole-body motion imitation on a full-sized robot, but with unnatural movements in the lower body. ExBody2 [7] achieved better whole-body tracking performance but with several separate specialist policies. VMP [34] demonstrated high-fidelity reproduction of a wide range of skills on the real robot, but its dependency on the mocap system during deployment limits its applicability in the wild.

3 Learning General Motion Tracking Controllers

In this section, we introduce GMT, a framework for developing general and high-quality motion tracking controllers for real-world humanoid robots. GMT follows the two-stage teacher-student training framework similar to prior works [7, 8], where a privileged teacher policy is first trained using PPO [37], and a student policy is then trained using DAgger[38] by imitating the output of the teacher policy. For the following part, we first introduce the core components of our method, which are *Adaptive Sampling strategy* and *Motion MoE architecture*. Then we present some key designs that are also crucial for policy learning, including motion input designs and the dataset curation process.

3.1 Adaptive Sampling

As illustrated in Fig. 2, large motion datasets like AMASS [6] exhibit significant category unbalance. This unbalance can substantially hinder the learning of less frequent and more complex motions. Moreover, longer sequences within such datasets are often composite motions, consisting of a series of skills that include both basic and challenging segments. Using previous sampling strategies [5, 3], when a policy fails on harder segments, it is still trained on the entire motion sequence — often dominated by easier parts — which results in a low effective sampling rate for the difficult portions.

To address these issues, we introduce *Adaptive Sampling*, which consists of two key components:

- **Random Clipping:** Motions longer than 10 seconds are clipped into several sub-clips, each with a maximum length of 10 seconds. To prevent artifacts at clip transitions, we apply randomized clipping by introducing a random offset of up to 2 seconds. Additionally,

all motions are re-clipped periodically during training to further diversify the sampled sub-clips;

- **Tracking Performance-based Probabilities:** During training, we log the completion level c_i of each motion, and terminate episodes when the tracking error exceeds E_i. When a motion is successfully completed, both c_i and E_i are gradually decreased, with minimum values of $c_{\text{min}} = 1.0$, $E_{\text{min}} = 0.25$. Therefore, the sampling probability of each motion is defined as:

$$p_i = \begin{cases} \left(\min\left(E_{\text{max\_key\_body\_error}}/0.15, 1\right)\right)^5, & c_i = 1, \\ c_i, & c_i > 1. \end{cases} \tag{1}$$

By applying Adaptive Sampling, we avoid repeatedly sampling easy segments of long motions and focus training efforts on refining performance on harder motions with higher tracking errors.

3.2 Motion Mixture-of-Experts

To enhance the expressiveness of the model, we incorporate a soft MoE module during the training of the teacher policy. An illustration of our model is shown in Fig. 3. The policy network consists of a group of expert networks and a gating network. The expert networks take both of the robot state observation and motion targets as input, which output the final action $\boldsymbol{a}_t$. The gating network also takes the same input observations and outputs a probability distribution over all experts. The final action output is a combination of actions sampled from each expert's individual action distributions: $\boldsymbol{a} = \sum_{i=1}^{n} p_i \boldsymbol{a}_i$, where p_i is the probability of each expert output by the gating network and $\boldsymbol{a}_i$ is the output of each expert policy.

3.3 Dataset Curation

We use a combination of AMASS [6] and LAFAN1 [39] to train the motion tracking policy. Since raw datasets contain infeasible motions like crawling, fallen states, and extremely dynamic ones due to hardware constraints. As infeasible motions represent noise and can hamper learning, we adopt a two-stage data curation process similar to previous works [5]. In the first stage, we apply rule-based filtering to eliminate infeasible motions — for example, motions where the root's roll or pitch angles exceed specified thresholds, or where the root height is abnormally high or low. In the second stage, we train a preliminary policy on previously filtered dataset with approximately 5 billion samples. Based on the completion rates achieved by this policy, we further filter out failed motions. This results in a curated version of the training dataset - a subset of AMASS and LAFAN1 with 8925 clips totaling 33.12 hours.

3.4 Motion Inputs

The goal of motion tracking is to let the robot track specific targets in each motion frame. We represent the motion tracking target of each frame as: $\boldsymbol{g}_t = [\boldsymbol{q}_t, \boldsymbol{v}_t^{\text{base}}, \boldsymbol{r}_t^{\text{base}}, \boldsymbol{p}_t^{\text{key}}, h_t^{\text{root}}]$, where $\boldsymbol{q}_t \in \mathbb{R}^{23}$ represents joint positions, $\boldsymbol{v}_t \in \mathbb{R}^6$ denotes the base linear and angular velocities, $\boldsymbol{r}_t \in \mathbb{R}^2$ represents the base roll and pitch angles, h_t^{root} represents root height, and $\boldsymbol{p}_t^{\text{key}} \in \mathbb{R}^{3\times \text{num keybody}}$ corresponds to the local key body positions. Unlike prior works that use global key body positions [8, 11], we adopt local key body positions similar to ExBody2 [7], with the further refinement of aligning the local key bodies relative to the robot's heading direction.

Furthermore, to improve tracking performance, we move beyond using only the immediate next motion frame as input [7, 10, 8]. Instead, we stack multiple consecutive frames $[\boldsymbol{g}_t, \ldots, \boldsymbol{g}_{t+100}]$ covering approximately two seconds of future motion. These stacked frames are then compressed by a convolutional [40] encoder into a latent vector $\boldsymbol{z}_t \in \mathbb{R}^{128}$, which is then combined with the immediate next frame $\boldsymbol{g}_t$ and fed into the policy network. This design enables the policy to capture both the long-term trends of the motion sequence and explicitly recognize the immediate tracking target. We show in experiments that this design is essential to high-quality tracking.

Table 2: Simulation evaluation of trained policies on AMASS-test and LAFAN dataset. All policies are trained on our filtered AMASS+LAFAN dataset. For all the baseline comparison and ablation studies, we only compare the performance of privileged policies.

	AMASS-Test				LAFAN1			
Method	$E_{\text{mpkpe}} \downarrow$	$E_{\text{mpjpe}} \downarrow$	$E_{\text{vel}} \downarrow$	$E_{\text{yaw vel}} \downarrow$	$E_{\text{mpkpe}} \downarrow$	$E_{\text{mpjpe}} \downarrow$	$E_{\text{vel}} \downarrow$	$E_{\text{yaw vel}} \downarrow$
Teacher & Student								
Privileged Policy	42.07	0.0834	0.1747	0.2238	45.16	0.0975	0.2837	0.3314
Student Policy	42.01	0.0807	0.1943	0.2211	46.14	0.1060	0.3009	0.3489
Baseline								
ExBody2 [7]	50.28±0.28	0.0925±0.001	0.1875±0.001	0.3402±0.004	58.36±0.48	0.1378±0.002	0.3461±0.005	0.4260±0.006
GMT (ours)	**42.07**±0.17	**0.0834**±0.001	**0.1747**±0.001	**0.2238**±0.002	**45.16**±0.35	**0.0975**±0.001	**0.2837**±0.004	**0.3314**±0.003
(a) Ablations								
GMT w.o. MoE	42.53±0.19	0.0874±0.00	0.1902±0.002	0.2483±0.001	48.26±0.29	0.1019±0.001	0.3111±0.003	0.3795±0.005
GMT w.o. A.S.	43.54±0.23	0.0872±0.001	0.2064±0.001	0.2593±0.001	49.61±0.30	0.1041±0.002	0.3019±0.003	0.3574±0.003
GMT w.o. A.S. & MoE	44.34±0.21	0.0920±0.001	0.2121±0.001	0.2534±0.001	52.34±0.33	0.1110±0.002	0.3263±0.003	0.3584±0.007
GMT (ours)	**42.07**±0.17	**0.0834**±0.001	**0.1747**±0.001	**0.2238**±0.002	**45.16**±0.35	**0.0975**±0.001	**0.2837**±0.004	**0.3314**±0.003
(b) Motion Inputs								
GMT-M	46.02±0.25	0.0942±0.001	0.2282±0.001	0.3311±0.003	51.16±0.34	0.1069±0.002	0.3476±0.001	0.4890±0.007
GMT-L0.5-M	43.64±0.19	0.0855±0.001	0.2051±0.001	0.2439±0.001	49.87±0.32	0.1032±0.002	0.3346±0.005	0.3648±0.003
GMT-L1-M	43.15±0.22	0.0867±0.001	0.1989±0.002	0.2465±0.001	47.41±0.35	0.1007±0.002	0.3047±0.003	0.3513±0.002
GMT-L2	49.52±0.27	0.1016±0.002	0.2201±0.001	0.2888±0.003	61.24±0.42	0.1368±0.002	0.3925±0.008	0.5558±0.009
GMT-L2-M (ours)	**42.07**±0.17	**0.0834**±0.001	**0.1747**±0.001	**0.2238**±0.002	**45.16**±0.35	**0.0975**±0.001	**0.2837**±0.004	**0.3314**±0.003

4 Experiments

4.1 Experimental Setups

We evaluate GMT in both simulation and real-world settings. For simulation experiments, each policy is trained using approximately 6.8 billion samples with domain randomization [41] and action delay [42], on a filtered subset of the AMASS and LAFAN1 [39] datasets; and evaluated on AMASS test set [3] and LAFAN1. We use IsaacGym [43] as the simulator and the number of parallel environments is 4096. For ablation studies and baseline comparisons, we focus on evaluating the performance of the privileged policy, as the deployable student policy is trained solely through imitation of the privileged teacher. For real-world experiments, we deploy our policy on Unitree G1 [44], a medium-sized humanoid robot with 23 DoFs and a height of 1.32 meters. The policy tracking performance is quantitatively evaluated with: **1)** E_{mpkpe}, mean per keybody position error, in mm; **2)** E_{mpjpe}, mean per joint position error, in rad; **3)** E_{vel}, linear velocity error, in m/s; **4)** $E_{\text{yaw vel}}$, yaw velocity error, in rad/s.

4.2 Baselines

We compare GMT's performance with ExBody2 [7] in simulation. We re-implement ExBody2 and train it on our filtered dataset. From table 2, we found that GMT outperforms ExBody2 on both local tracking performance (E_{mpkpe} and E_{mpjpe}) and global tracking performance (E_{vel} and $E_{\text{yaw vel}}$).

4.3 Ablation Studies

For this part, we conduct ablation studies to investigate the contribution of each component. Specifically, to evaluate the effects of the Motion MoE architecture and Adaptive Sampling strategy, we consider ablations include: **1) GMT w.o. A.S. & MoE:** MoE model is replaced with an MLP of equivalent number of parameters and Adaptive Sampling is removed. **2) GMT w.o. MoE:** Only MoE model is replaced with a MLP of the same size. **3) GMT w.o. A.S.:** Only Adaptive Sampling is removed during training.

Additionally, we investigate the impact of motion input configuration with ablations: **GMT-M:** Only the immediate next frame of motion is provided as input to the policy network; **GMT-Lx-M:** Both the immediate next frame and a window of *x* seconds of future motion frames are input to the policy network; **GMT-Lx:** Only a window of *x* seconds of future motion frames is provided, without including the immediate next frame.

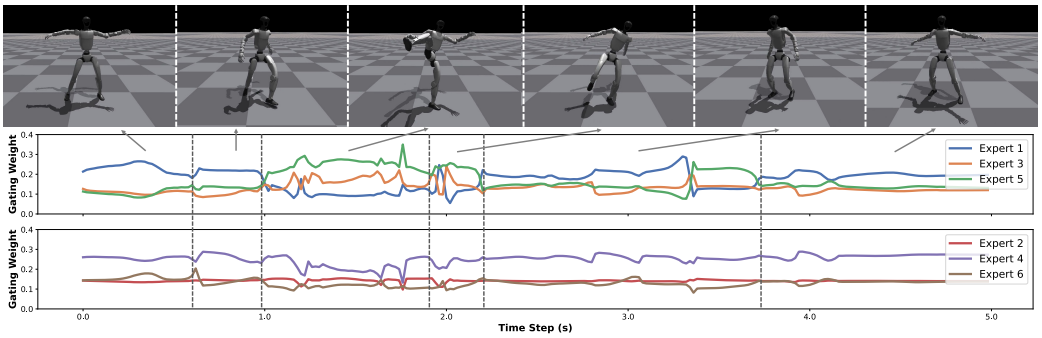


Figure 4: Plot of the output of gating network with respect to time on a motion clip composed of a sequence of skills.

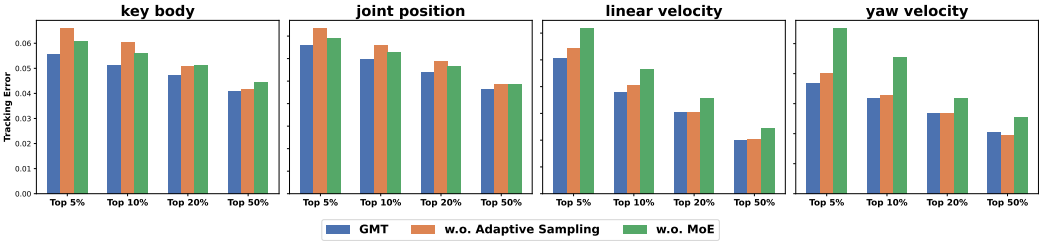


Figure 5: Top percentile tracking errors on the whole AMASS dataset.

4.3.1 Motion MoE

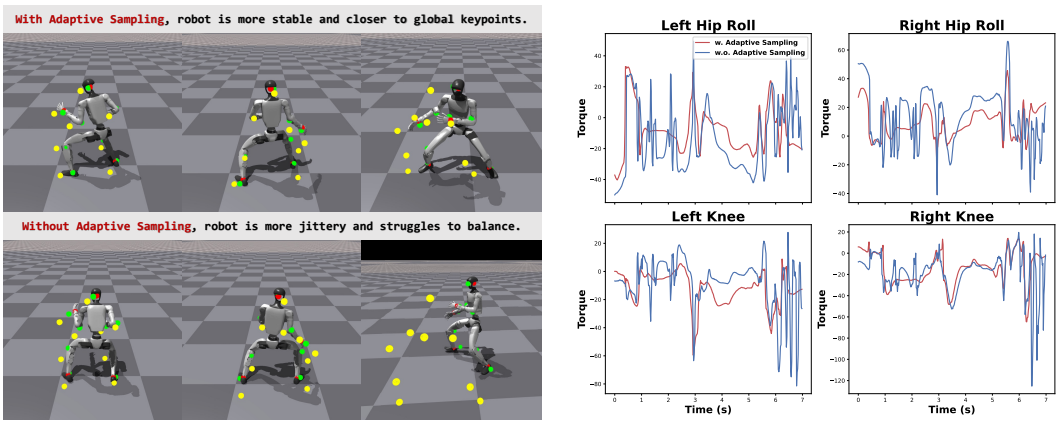
As shown in table 2(a), GMT outperforms the baselines in tracking performance on both the AMASS test set and the entire LAFAN1 dataset. Furthermore, as shown in Fig. 5, where top percentile tracking errors on AMASS are recorded. Both the statistics and the figure indicate that MoE helps improve a lot more on more challenging motions. For qualitative evaluation, Fig. 4, visualized the expert selection on a composite motion sequence consisting of standing, kicking, walking backward for a few steps, and standing again. The gating weights of each expert over time show clear transitions in expert activation across different phases of the motion. This suggests that individual experts specialize in different types of motion, validating the intended role of the MoE structure in capturing motion diversity.

4.3.2 Adaptive Sampling

As shown in table 2(a), Adaptive Sampling effectively improved the tracking performance on both datasets. Similar to Motion MoE, as shown in Fig. 5, Adaptive Sampling improves more on more challenging motions. To qualitatively evaluate the influence of this strategy, we extract a short clip from a long and composite 240s motion, and compare the performance of policies trained with and without Adaptive Sampling in Fig. 6(a). In addition, we plot the torque of key joints, including the knee and hip roll, in Fig. 6(b). The results show that without Adaptive Sampling, the policy fails to learn this clip with high-quality and struggles to balance. These artifacts make real-world deployment impossible.

4.3.3 Motion Inputs

Experimental results in table 2(b) show that increasing the motion input window length leads to improved tracking accuracy. However, the performance of **GMT-L2** shows a significant degradation, indicating that inputting the immediate next frame into the policy network is crucial as well. This can be explained as that while a sequence of future frames captures the overall tendency of upcoming motions, it can lose some detailed information. In this way, inputting the immediate next



(a) (b)

Figure 6: The performance of policies with and without Adaptive Sampling on one segment extracted from a long motion clip. (a) Visualization of policy performance in the simulator. (b) Torque outputs of several joints corresponding to this segment.

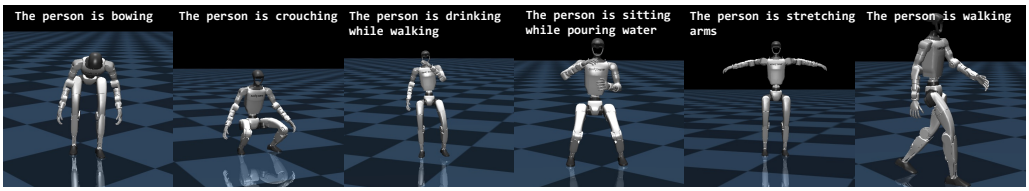


Figure 7: Motion tracking on MDM-generated motions.

frame into the policy can greatly enhance tracking performance by providing the nearest relevant information.

4.4 Real-World Deployment

As shown in Fig. 1, we successfully deploy our policy on a real-world humanoid robot, reproducing a wide range of human motions — including stylized walking, high kicking, dancing, spinning, crouch walking, soccer kicking, and many others — with high fidelity and state-of-the-art performance. For more detailed deployment results, please refer to the project website.

4.5 Applications - Tracking MDM-Generated Motions

We test our policy with motion diffusion model (MDM) [45] generated motions in MuJoCo [46] sim-to-sim settings. Results in Fig. 7 show that GMT can perform well on motions generated with MDM by text prompts, proving the potential of GMT to be applied to other downstream tasks.

5 Conclusion

In this paper, we introduce GMT, a general and scalable motion tracking framework that trains a single unified policy to enable humanoid robots to imitate diverse motions in the real world. We conduct extensive experiments to evaluate the contribution of each component to the overall tracking performance and demonstrate that GMT can track motions from other resources like MDM. Real-world deployment results demonstrate that our general unified policy achieves state-of-the-art performance compared to prior works. We believe that this general controller can serve as a foundation for future whole-body algorithm development on humanoid robots.

6 Limitations

While GMT achieves state-of-the-art performance as a unified, general motion tracking controller, it still has several limitations:

- **Lack of Contact-Rich Skills.** Due to the significant additional simulation complexity required to model contact-rich behaviors [27], along with hardware limitations, our framework does not currently support skills such as getting up from a fallen state or rolling on the ground.
- **Limitations on Challenging Terrains.** Our current policy is trained without any terrain observations and is not designed for imitation on challenging terrains like slopes and stairs. In future work, we aim to extend our framework to develop a general and robust controller capable of operating across both flat and challenging terrains.

References

[1] X. B. Peng, P. Abbeel, S. Levine, and M. Van de Panne. Deepmimic: Example-guided deep reinforcement learning of physics-based character skills. *ACM Transactions On Graphics (TOG)*, 37(4):1–14, 2018.

[2] X. B. Peng, Y. Guo, L. Halper, S. Levine, and S. Fidler. Ase: Large-scale reusable adversarial skill embeddings for physically simulated characters. *ACM Transactions On Graphics (TOG)*, 41(4):1–17, 2022.

[3] Z. Luo, J. Cao, K. Kitani, W. Xu, et al. Perpetual humanoid control for real-time simulated avatars. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10895–10904, 2023.

[4] T. E. Truong, M. Piseno, Z. Xie, and K. Liu. Pdp: Physics-based character animation via diffusion policy. In *SIGGRAPH Asia 2024 Conference Papers*, pages 1–10, 2024.

[5] 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)*, 43(6):1–21, 2024.

[6] 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.

[7] 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.

[8] 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.

[9] Z. Fu, Q. Zhao, Q. Wu, G. Wetzstein, and C. Finn. Humanplus: Humanoid shadowing and imitation from humans. *arXiv preprint arXiv:2406.10454*, 2024.

[10] 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.

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

[12] H. Miura and I. Shimoyama. Dynamic walk of a biped. *The International Journal of Robotics Research*, 3(2):60–74, 1984.

[13] K. Sreenath, H.-W. Park, I. Poulakakis, and J. W. Grizzle. A compliant hybrid zero dynamics controller for stable, efficient and fast bipedal walking on mabel. *The International Journal of Robotics Research*, 30(9):1170–1193, 2011.

[14] H. Geyer, A. Seyfarth, and R. Blickhan. Positive force feedback in bouncing gaits? *Proceedings of the Royal Society of London. Series B: Biological Sciences*, 270(1529):2173–2183, 2003.

[15] I. Radosavovic, B. Zhang, B. Shi, J. Rajasegaran, S. Kamat, T. Darrell, K. Sreenath, and J. Malik. Humanoid locomotion as next token prediction. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[16] I. Radosavovic, S. Kamat, T. Darrell, and J. Malik. Learning humanoid locomotion over challenging terrain. *arXiv preprint arXiv:2410.03654*, 2024.

[17] I. Radosavovic, T. Xiao, B. Zhang, T. Darrell, J. Malik, and K. Sreenath. Real-world humanoid locomotion with reinforcement learning. *Science Robotics*, 9(89):eadi9579, 2024.

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

[19] Z. Chen, X. He, Y.-J. Wang, Q. Liao, Y. Ze, Z. Li, S. S. Sastry, J. Wu, K. Sreenath, S. Gupta, et al. Learning smooth humanoid locomotion through lipschitz-constrained policies. *arXiv preprint arXiv:2410.11825*, 2024.

[20] Q. Ben, F. Jia, J. Zeng, J. Dong, D. Lin, and J. Pang. Homie: Humanoid loco-manipulation with isomorphic exoskeleton cockpit. *arXiv preprint arXiv:2502.13013*, 2025.

[21] Q. Zhang, P. Cui, D. Yan, J. Sun, Y. Duan, G. Han, W. Zhao, W. Zhang, Y. Guo, A. Zhang, et al. Whole-body humanoid robot locomotion with human reference. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 11225–11231. IEEE, 2024.

[22] T. He, W. Xiao, T. Lin, Z. Luo, Z. Xu, Z. Jiang, J. Kautz, C. Liu, G. Shi, X. Wang, et al. Hover: Versatile neural whole-body controller for humanoid robots. *arXiv preprint arXiv:2410.21229*, 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. *arXiv preprint arXiv:2403.04436*, 2024.

[24] C. Zhang, W. Xiao, T. He, and G. Shi. Wococo: Learning whole-body humanoid control with sequential contacts. *arXiv preprint arXiv:2406.06005*, 2024.

[25] Z. Zhuang, S. Yao, and H. Zhao. Humanoid parkour learning. *arXiv preprint arXiv:2406.10759*, 2024.

[26] Y. Xue, W. Dong, M. Liu, W. Zhang, and J. Pang. A unified and general humanoid whole-body controller for fine-grained locomotion. *arXiv preprint arXiv:2502.03206*, 2025.

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

[28] 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. *arXiv preprint arXiv:2502.08378*, 2025.

[29] X. B. Peng, Z. Ma, P. Abbeel, S. Levine, and A. Kanazawa. Amp: Adversarial motion priors for stylized physics-based character control. *ACM Transactions on Graphics (ToG)*, 40(4): 1–20, 2021.

[30] 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*, pages 1–9, 2023.

[31] M. Hassan, Y. Guo, T. Wang, M. Black, S. Fidler, and X. B. Peng. Synthesizing physical character-scene interactions. In *ACM SIGGRAPH 2023 Conference Proceedings*, pages 1–9, 2023.

[32] Y. Yuan, V. Makoviychuk, Y. Guo, S. Fidler, X. Peng, and K. Fatahalian. Learning physically simulated tennis skills from broadcast videos. *ACM Trans. Graph*, 42(4), 2023.

[33] 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.

[34] A. Serifi, R. Grandia, E. Knoop, M. Gross, and M. Bächer. Vmp: Versatile motion priors for robustly tracking motion on physical characters. In *Computer Graphics Forum*, page e15175. Wiley Online Library, 2024.

[35] J. Mao, S. Zhao, S. Song, T. Shi, J. Ye, M. Zhang, H. Geng, J. Malik, V. Guizilini, and Y. Wang. Learning from massive human videos for universal humanoid pose control. *arXiv preprint arXiv:2412.14172*, 2024.

[36] F. Liu, Z. Gu, Y. Cai, Z. Zhou, S. Zhao, H. Jung, S. Ha, Y. Chen, D. Xu, and Y. Zhao. Opt2skill: Imitating dynamically-feasible whole-body trajectories for versatile humanoid loco-manipulation. *arXiv preprint arXiv:2409.20514*, 2024.

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

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

[39] F. G. Harvey, M. Yurick, D. Nowrouzezahrai, and C. Pal. Robust motion in-betweening. *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH)*, 39(4), 2020.

[40] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

[41] X. B. Peng, M. Andrychowicz, W. Zaremba, and P. Abbeel. Sim-to-real transfer of robotic control with dynamics randomization. In *2018 IEEE international conference on robotics and automation (ICRA)*, pages 3803–3810. IEEE, 2018.

[42] 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*, pages 91–100. PMLR, 2022.

[43] V. Makoviychuk, L. Wawrzyniak, Y. Guo, M. Lu, K. Storey, M. Macklin, D. Hoeller, N. Rudin, A. Allshire, A. Handa, et al. Isaac gym: High performance gpu-based physics simulation for robot learning. *arXiv preprint arXiv:2108.10470*, 2021.

[44] Unitree Robotics. Unitree g1 – humanoid agent ai avatar, 2025. URL https://www.unitree.com/g1.

[45] G. Tevet, S. Raab, B. Gordon, Y. Shafir, D. Cohen-Or, and A. H. Bermano. Human motion diffusion model. *arXiv preprint arXiv:2209.14916*, 2022.

[46] E. Todorov, T. Erez, and Y. Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5026–5033. IEEE, 2012. doi:10.1109/IROS.2012.6386109.

[47] R. S. Sutton, A. G. Barto, et al. *Reinforcement learning: An introduction*. MIT press Cambridge, 1998.

Appendix

6.1 Goal-Conditioned Reinforcement Learning

In this work, motion tracking problem is defined as a goal-conditioned RL problem where given the goal, the agent interacts with the environment according to the policy π to maximize an objective function [47]. At each timestep t, the agent's policy takes the state $\boldsymbol{s}_t$ and the goal $\boldsymbol{g}_t$ as input, and outputs the action $\boldsymbol{a}_t$, which is formulated as $\pi(\boldsymbol{a}_t|\boldsymbol{s}_t, \boldsymbol{g}_t)$. When applied to the environment, the action $\boldsymbol{a}_t$ leads to the next state $\boldsymbol{s}_{t+1}$ according to the environment dynamics $p(\boldsymbol{s}_{t+1}|\boldsymbol{s}_t, \boldsymbol{a}_t)$. A reward $r(\boldsymbol{s}_{t+1}, \boldsymbol{s}_t, \boldsymbol{a}_t)$ is received at each timestep. The agent's goal is to maximize the expected return:

$$J(\pi) = \mathbb{E}_{p(\tau|\pi)} \left[\sum_{t=0}^{T-1} \gamma^t r_t \right], \tag{2}$$

where $p(\tau|\pi)$ represents the likelihood of the trajectory τ, T denotes the time horizon, and γ is the discount factor.

6.2 Sim-to-Real Transfer

To enable successful sim-to-real transfer, we apply domain randomizations during the training of both teacher and student policies [41, 42]. To further align simulation and real-world physical dynamics,we explicitly model the effect of the reduction drive's moment of inertia. Specifically, given a reduction ratio k, and a reduction drive moment of inertia I, we configure the `armature` parameter in the simulator as:

$$\texttt{armature} = k^2 I, \tag{3}$$

to approximate the effective inertia introduced by the reduction drive.

6.3 Policy Learning

Following previous works [7, 8], we adopt a two-stage training framework. In the first stage, we train a privileged teacher policy that observes both proprioceptive and privileged information, and outputs joint target actions $\hat{\boldsymbol{a}}_t$, optimized using PPO [37]. In the second stage, we train a deployable student policy that takes as input a sequence of proprioceptive observation history, and is supervised by the teacher policy through DAgger [38]. The student policy is optimized by minimizing the ℓ_2 loss between its output $\boldsymbol{a}_t$ and the teacher's output $\hat{\boldsymbol{a}}_t$ according to $\|\hat{\boldsymbol{a}}_t - \boldsymbol{a}\|_2^2$.

6.4 Observations and Actions

For the teacher policy observations consist of proprioception $\boldsymbol{o}_t$, privileged information $\boldsymbol{e}_t$, and motion targets $\boldsymbol{g}$. $\boldsymbol{o}_t$ consists of root angular velocity (3 dims), root roll and pitch (2 dims), joint positions (23 dims), joint velocities (23 dims), and last action (23 dims). $\boldsymbol{e}_t$ consists of root linear velocity (3 dims), root height (1 dim), key body positions, feet contact mask (2 dims), mass randomization params (6 dim), and motor strength (46 dims). For the student policy, observation consists of proprioception $\boldsymbol{o}_t$, proprioception history $\boldsymbol{o}_{t-20}, \ldots, \boldsymbol{o}_t$, and motion targets $\boldsymbol{g}$.

The output action is target joint positions.

6.5 Training Details

We use IsaacGym [43] as the physics simulator and the number of parallel environments is 4096. We train the privileged policy for around 3 days on an RTX4090 GPU, and then train the student policy for around 1 day. The simulation frequency is 500Hz and the control frequency is 50Hz. The trained policy is validated in Mujoco [46] before deploying onto the real robot.

6.6 Reward Functions

Reward functions of first-stage training. Here $\boldsymbol{q}$ denotes joint positions, $\dot{\boldsymbol{q}}$ denotes joint veocities, $\ddot{\boldsymbol{q}}$ denotes joint accelerations, $\boldsymbol{r}$ denotes root rotations, $\boldsymbol{v}$ denotes root velocites, h denotes root height, and $\boldsymbol{p}$ denotes key body positions. Full definitions are in table 3.

Table 3: Definitions of Reward Functions.

Name	**Definitions**
tracking joint positions	$\exp(\|-\boldsymbol{q}_t^{\text{ref}}-\boldsymbol{q}_t\|_2^2)$
tracking joint velocities	$\exp(\|-\boldsymbol{q}_t^{\text{ref}}-\dot{\boldsymbol{q}}_t\|_2^2)$
tracking root pose	$\exp(\|-\boldsymbol{r}_t^{\text{ref}}-\boldsymbol{r}_t\|_2^2-\|h_t^{\text{ref}}-h_t\|_2^2)$
tracking root vel	$\exp(\|-\boldsymbol{v}_t^{\text{ref}}-\boldsymbol{v}_t\|_2^2)$
tracking key body positions	$\exp(\|-\boldsymbol{p}_t^{\text{ref}}-\boldsymbol{p}_t\|_2^2)$
alive	1.0
foot slip	$(\boldsymbol{v}_t^{\text{foot}} I_t^{\text{foot contact}})$
joint velocities	$-\|\dot{\boldsymbol{q}}_t\|$
joint accelerations	$-\|\ddot{\boldsymbol{q}}_t\|$
action rate	$-\|\boldsymbol{a}_t-\boldsymbol{a}_{t-1}\|$