KungfuBot: Physics-Based Humanoid Whole-Body Control for Learning Highly-Dynamic Skills

Weiji Xie[*1,2] **Jinrui Han**[*1,2] **Jiakun Zheng**[*1,3] **Huanyu Li**[1,4] **Xinzhe Liu**[1,5]
Jiyuan Shi[1] **Weinan Zhang**[2] **Chenjia Bai**[†1] **Xuelong Li**[†1]

[1]Institute of Artificial Intelligence (TeleAI), China Telecom
[2]Shanghai Jiao Tong University [3]East China University of Science and Technology
[4]Harbin Institute of Technology [5]ShanghaiTech University

Abstract

Humanoid robots are promising to acquire various skills by imitating human behaviors. However, existing algorithms are only capable of tracking smooth, low-speed human motions, even with delicate reward and curriculum design. This paper presents a physics-based humanoid control framework, aiming to master highly-dynamic human behaviors such as Kungfu and dancing through multi-steps motion processing and adaptive motion tracking. For motion processing, we design a pipeline to extract, filter out, correct, and retarget motions, while ensuring compliance with physical constraints to the maximum extent. For motion imitation, we formulate a bi-level optimization problem to dynamically adjust the tracking accuracy tolerance based on the current tracking error, creating an adaptive curriculum mechanism. We further construct an asymmetric actor-critic framework for policy training. In experiments, we train whole-body control policies to imitate a set of highly-dynamic motions. Our method achieves significantly lower tracking errors than existing approaches and is successfully deployed on the Unitree G1 robot, demonstrating stable and expressive behaviors. The project page is `https://kungfu-bot.github.io`.

1 Introduction

Humanoid robots, with their human-like morphology, have the potential to mimic various human behaviors in performing different tasks [1]. The ongoing advancement of motion capture (MoCap) systems and motion generation methods has led to the creation of extensive motion datasets [2, 3], which encompass a multitude of human activities annotated with textual descriptions [4]. Consequently, it becomes promising for humanoid robots to learn whole-body control to imitate human behaviors. However, controlling high-dimensional robot actions to achieve ideal human-like performance presents a substantial challenge. One major difficulty arises from the fact that motion sequences captured from humans may not comply with the physical constraints of humanoid robots, including joint limits, dynamics, and kinematics [5, 6]. Hence, directly training policies through Reinforcement Learning (RL) to maximize rewards (e.g., the negative tracking error) often fails to yield desirable policies, as it may not exist within the solution space.

Recently, several RL-based whole-body control frameworks have been proposed to track motions [7, 8], which often take a reference kinematic motion as input and output the control actions for a humanoid robot to imitate it. To address physical feasibility issues, H2O and OmniH2O [9, 10] remove the infeasible motions using a trained privileged imitation policy, producing a clean motion

[*]Equal contributions.
[†]Correspondence to: Chenjia Bai (baicj@chinatelecom.cn)

Preprint. Under review.

dataset. ExBody [7] constructs a feasible motion dataset by filtering via language labels, such as 'wave' and 'walk'. Exbody2 [5] trains an initial policy on all motions and uses the tracking error to measure the difficulty of each motion. However, it would be costly to train the initial policy and find an optimal dataset. There is also a lack of suitable tolerance mechanisms for difficult-to-track motions in the training process. As a result, previous methods are only capable of tracking low-speed and smooth motions. Recently, ASAP [6] introduces a multi-stage mechanism and learned a residual policy to compensate for the sim-to-real gap, reducing the difficulties in tracking agile motions. However, ASAP involves a total of four training stages, and the training of residual policy requires MoCap systems to record real-robot states.

In this paper, we propose *Physics-Based Humanoid motion Control (PBHC)*, which utilizes a two-stage framework to tackle the challenges associated with agile and highly-dynamic motions. (i) In the motion processing stage, we first extract motions from videos and establish physics-based metrics to filter out human motions by estimating physical quantities within the human model, thereby eliminating motions beyond the physical limits. Then, we compute contact masks of motions followed by motion correction, and finally retarget processed motions to the robot using differential inverse kinematics. (ii) In the motion imitation stage, we propose an adaptive motion tracking mechanism that adjusts the tracking reward via a tracking factor. Perfectly tracking hard motions is impractical due to imperfect reference motions and the need of smooth control, so we adapt the tracking factor to different motions based on the tracking error. We then formulate a Bi-Level Optimization (BLO) [11] to derive the optimal factor and design an adaptive update rule that estimates the tracking error online to dynamically refine the factor during training.

Building on the two-stage framework, we design an asymmetric actor-critic architecture for policy optimization. The critic adopts a reward vectorization technique and leverages privileged information to improve value estimation, while the actor relies solely on local observations. In experiments, PBHC enables whole-body control policies to track highly-dynamic motions with lower tracking errors than existing methods. We further demonstrate successful real-world deployment on the Unitree G1 robot, achieving stable and expressive behaviors, including complex motions like Kungfu and dancing.

2 Preliminaries

Problem Formulation. We adopt the Unitree G1 robot [12] in our work, which has 23 degrees of freedom (DoFs) to control, excluding the 3 DoFs in each wrist of the hand. We formulate the motion imitation problem as a goal-conditional RL problem with Markov Decision Process $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{S}^{\text{ref}}, \gamma, r, P)$, where $\mathcal{S}$ and $\mathcal{S}^{\text{ref}}$ are the state spaces of the humanoid robot and reference motion, respectively, $\mathcal{A}$ is the robot's action space, r is a mixed reward function consisting motion-tracking and regularization rewards, and P is the transition function depending on the robot morphology and physical constraints. At each time step t, the policy π observes the proprioceptive state $\boldsymbol{s}_t^{\text{prop}}$ of the robot and generates action $\boldsymbol{a}_t$, with the aim of obtaining the next-state $\boldsymbol{s}_{t+1}$ that follows the corresponding reference state $\boldsymbol{s}_{t+1}^{\text{ref}}$ in the reference trajectory $[\boldsymbol{s}_0^{\text{ref}}, \ldots, \boldsymbol{s}_{N-1}^{\text{ref}}]$. The action $\boldsymbol{a}_t \in \mathbb{R}^{23}$ is the target joint position for a PD controller to compute the motor torques. We adopt an off-the-shelf RL algorithm, PPO [13], for policy optimization with an actor-critic architecture.

Reference Motion Processing. For human motion processing, the Skinned Multi-Person Linear (SMPL) model [14] offers a general representation of human motions, using three key parameters: $\boldsymbol{\beta} \in \mathbb{R}^{10}$ for body shapes, $\boldsymbol{\theta} \in \mathbb{R}^{24\times3}$ for joint rotations in axis-angle representation, and $\boldsymbol{\psi} \in \mathbb{R}^3$ for global translation. These parameters can be mapped to a 3D mesh consisting of 6,890 vertices via a differentiable skinning function $M(\cdot)$, which formally expressed as $\mathcal{V} = M(\boldsymbol{\beta}, \boldsymbol{\theta}, \boldsymbol{\psi}) \in \mathbb{R}^{6890\times3}$. We employ a human motion recovery model to estimate SMPL parameters $(\boldsymbol{\beta}, \boldsymbol{\theta}, \boldsymbol{\psi})$ from videos, followed by additional motion processing. The resulting SMPL-format motions are then retargeted to G1 through an Inverse Kinematics (IK) method, yielding the reference motions for tracking purposes.

3 Methods

An overview of PBHC is illustrated in Fig. 1. First, raw human videos are processed by a Human Motion Recovery (HMR) model to produce SMPL-format motion sequences. These sequences are filtered via physics-based metrics and corrected using contact masks. The refined motions are then retargeted to the G1 robot. Finally, each resulting trajectory serves as reference motion for training a separate RL policy, which is then deployed on the real G1 robot. In the following, we detail the motion processing pipeline (§3.1), adaptive motion tracking module (§3.2) and RL framework (§3.3).

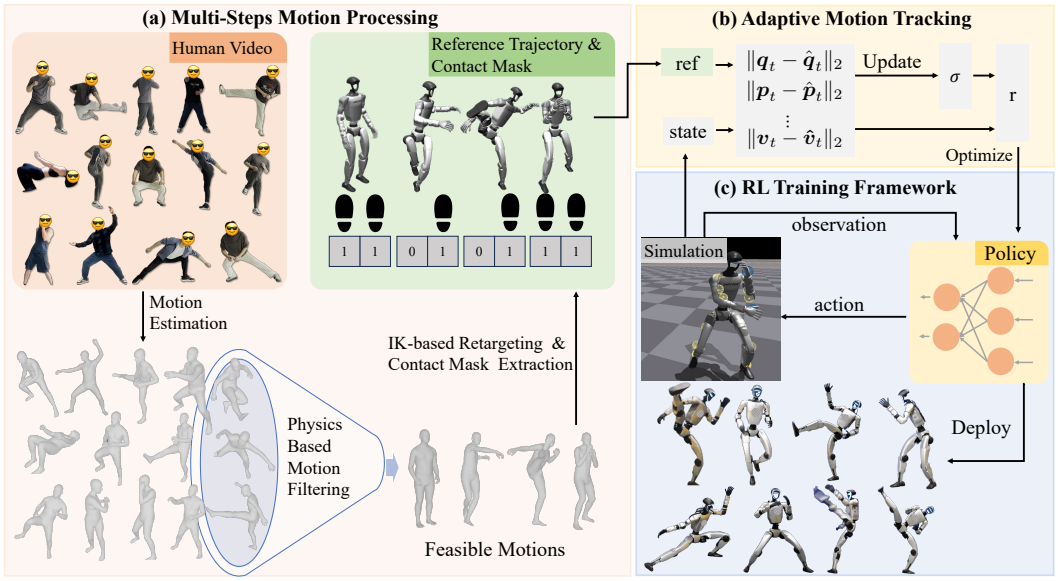

Figure 1: An overview of PBHC that includes three core components: (a) motion extraction from videos and multi-steps motion processing, (b) adaptive motion tracking based on the optimal tracking factor, (c) the RL training framework and sim-to-real deployment.

3.1 Motion Processing Pipeline

We propose a motion processing pipeline to extract motion from videos for humanoid motion tracking, comprising four steps: (i) SMPL-format motion estimation from monocular videos, (ii) physics-based motion filtering, (iii) contact-aware motion correction, and (iv) motion retargeting. This pipeline ensures that physically plausible motions can be transferred from videos to humanoid robots.

Motion Estimation from Videos. We employ GVHMR [15] to estimate SMPL-format motions from monocular videos. GVHMR introduces a gravity-view coordinate system that naturally aligns motions with gravity, eliminating body tilt issues caused by reconstruction solely relying on the camera coordinate system. Furthermore, it mitigates foot sliding artifacts by predicting foot stationary probabilities, thereby enhancing motion quality.

Physics-based Motion Filtering. Due to reconstruction inaccuracies and out-of-distribution issues in HMR models, motions extracted from videos may violate physical and biomechanical constraints. Thus, we try to filter out these motions via physics-based principles. Previous work [16] suggests that proximity between the center of mass (CoM) and center of pressure (CoP) indicates greater stability, and proposes a method to estimate CoM and CoP coordinates from SMPL data. Building on this, we calculate the projected distance of CoM and CoP on the ground for each frame and apply a threshold to assess stability. Specifically, let $\bar{\boldsymbol{p}}_t^{\text{CoM}} = (p_{t,x}^{\text{CoM}}, p_{t,y}^{\text{CoM}})$ and $\bar{\boldsymbol{p}}_t^{\text{CoP}} = (p_{t,x}^{\text{CoP}}, p_{t,y}^{\text{CoP}})$ denote the projected coordinates of CoM and CoP on the ground at frame t respectively, and Δd_t represents the distance between these projections. We define the stability criterion of a frame as

$$\Delta d_t = \|\bar{\boldsymbol{p}}_t^{\text{CoM}} - \bar{\boldsymbol{p}}_t^{\text{CoP}}\|_2 < \epsilon_{\text{stab}}, \tag{1}$$

where ϵ_{stab} represents the stability threshold. Then, given an N-frame motion sequence, let $\mathcal{B} = [t_0, t_1, \ldots, t_K]$ be the increasingly sorted list of frame indices that satisfy Eq. (1), where $t_k \in [1, N]$. The motion sequence is considered stable if it satisfies two conditions: (i) Boundary-frame stability: $1 \in \mathcal{B}$ and $N \in \mathcal{B}$. (ii) Maximum instability gap: the maximum length of consecutive unstable frames must be less than threshold ϵ_{N}, i.e., $\max_k t_{k+1} - t_k < \epsilon_{\text{N}}$. Based on this criterion, motions that are clearly unable to maintain dynamic stability can be excluded from the original dataset.

Motion Correction based on Contact Mask. To better capture foot-ground contact in motion data, we estimate contact masks by analyzing ankle displacement across consecutive frames, based on the zero-velocity assumption [17, 18]. Let $\boldsymbol{p}_t^{\text{l-ankle}} \in \mathbb{R}^3$ denote the position of the left ankle joint at time t, and $c_t^{\text{left}} \in \{0, 1\}$ the corresponding contact mask. The contact mask is estimated as

$$c_t^{\text{left}} = \mathbb{I}[\|\boldsymbol{p}_{t+1}^{\text{l-ankle}} - \boldsymbol{p}_t^{\text{l-ankle}}\|_2^2 < \epsilon_{\text{vel}}] \cdot \mathbb{I}[p_{t,z}^{\text{l-ankle}} < \epsilon_{\text{height}}], \tag{2}$$

where ϵ_{vel} and ϵ_{height} are empirically chosen thresholds. Similarly for the right foot.

To address minor floating artifacts not eliminated by threshold-based filtering, we apply a correction step based on the estimated contact mask. Specifically, if either foot is in contact at frame t, a vertical offset is applied to the global translation. Let $\boldsymbol{\psi}_t$ denotes the global translation of the pose at time t, then the corrected vertical position is:

$$\psi_{t,z}^{\text{corr}} = \psi_{t,z} - \Delta h_t, \tag{3}$$

where $\Delta h_t = \min_{v \in \mathcal{V}_t} p_{t,z}^v$ is the lowest z-coordinate among the SMPL mesh vertices $\mathcal{V}_t$ at frame t. While the correction alleviates floating artifacts, it may cause frame-to-frame jitter. We address this by applying Exponential Moving Average (EMA) to smooth the motion.

Motion Retargeting. We adopt an inverse kinematics (IK)-based method [19] to retarget processed SMPL-format motions to the G1 robot. This approach formulates a differentiable optimization problem that ensures end-effector trajectory alignment while respecting joint limits.

To enhance motion diversity, we incorporate additional data from open-source datasets, AMASS [4] and LAFAN [20]. These motions are partially processed through our pipeline, including contact mask estimation, motion correction, and retargeting.

3.2 Adaptive Motion Tracking

3.2.1 Exponential Form Tracking Reward

The reward function in PBHC, detailed in Appendix C.2, comprises two components: task-specific rewards, which enforce accurate tracking of reference motions, and regularization rewards, which promote overall stability and smoothness.

The task-specific rewards include terms for aligning joint states, rigid body state, and foot contact mask. These rewards, except the foot contact tracking term, follow the exponential form as:

$$r(x) = \exp(-x/\sigma), \tag{4}$$

where x represents the tracking error, typically measured as the mean squared error (MSE) of quantities such as joint angles, while σ controls the tolerance of the error, referred to as the *tracking factor*. This exponential form is preferred over the negative error form because it is bounded, helps stabilize the training process, and provides a more intuitive approach for reward weighting.

Intuitively, when σ is much larger than the typical range of x, the reward remains close to 1 and becomes insensitive to changes in x, while an overly small σ causes the reward to approach 0 and also reduces its sensitivity, highlighting the importance of choosing σ appropriately to enhance responsiveness and hence tracking precision. This intuition is illustrated in Fig. 2.

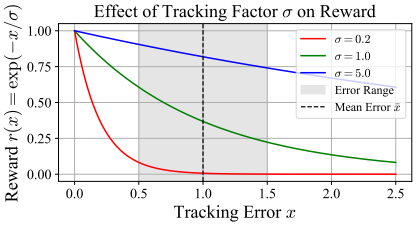

Figure 2: Illustration of the effect of tracking factor σ on the reward value.

3.2.2 Optimal Tracking Factor

To determine the choice of the optimal tracking factor, we introduce a simplified model of motion tracking and formulate it as a bi-level optimization problem. The intuition behind this formulation is that **the tracking factor σ should be chosen to minimize the accumulated tracking error of the converged policy over the reference trajectory**. In manual tuning scenarios, this is typically achieved through an iterative process where an engineer selects a value for σ, trains a policy, observes the results, and repeats the process until satisfactory performance is attained.

Given a policy π , there is a sequence of expected tracking error $\boldsymbol{x} \in \mathbb{R}_+^N$ for N steps, where x_i represents the expected tracking error at the i-th step of the rollout episodes. Rather than optimizing the policy directly, we treat the tracking error sequence $\boldsymbol{x}$ as decision variables. This allows us to reformulate the optimization problem of motion tracking as:

$$\max_{\boldsymbol{x} \in \mathbb{R}_+^N} J^{\text{in}}(\boldsymbol{x}, \sigma) + R(\boldsymbol{x}), \tag{5}$$

where the *internal* objective $J^{\text{in}}(\boldsymbol{x}, \sigma) = \sum_{i=1}^{N} \exp(-x_i/\sigma)$ is the simplified accumulated reward induced by the tracking reward in Eq. (4), and we introduce $R(\boldsymbol{x})$ to capture all additional effects beyond J^{in}, including environment dynamics and other policy objectives such as extra rewards. The solution $\boldsymbol{x}^*$ to Eq. (5) corresponds to the error sequence induced by the optimal policy π^*. Subsequently, the optimization objective of σ is to maximize the obtained accumulated negative tracking error $J^{\text{ex}}(\boldsymbol{x}^*) = \sum_{i=1}^{N} -x_i^*$, the *external* objective, formalized as the following bi-level optimization problem:

$$\max_{\sigma \in \mathbb{R}_+} \quad J^{\text{ex}}(\boldsymbol{x}^*), \qquad \text{s.t.} \quad \boldsymbol{x}^* \in \arg\max_{\boldsymbol{x} \in \mathbb{R}_+^N} J^{\text{in}}(\boldsymbol{x}, \sigma) + R(\boldsymbol{x}). \tag{6}$$

Under additional technical assumptions, we can solve Eq. (6) and derive that the optimal tracking factor is the average of the optimal tracking error, as detailed in Appendix A.

$$\sigma^* = \left(\sum_{i=1}^{N} x_i^* \right) / N. \tag{7}$$

3.2.3 Adaptive Mechanism

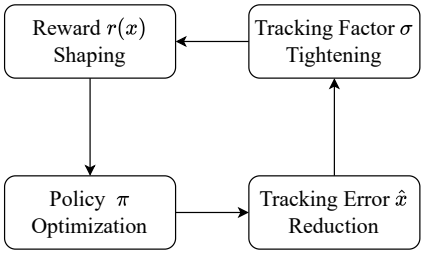

Figure 3: Closed-loop adjustment of tracking factor in the proposed adaptive mechanism.

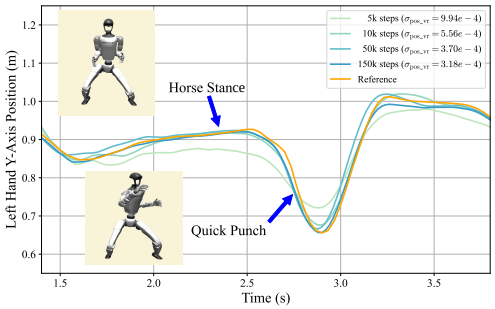

Figure 4: Example of the right hand y-position for 'Horse-stance punch'. The adaptive σ can progressively improve the tracking precision. $\sigma_{\text{pos\_vr}}$ is used for tracking the head and hands.

While Eq. (7) provides a theoretical guidance for determining the tracking factor, the coupling between σ^* and $\boldsymbol{x}^*$ creates a circular dependency that prevents direct computation. Additionally, due to the varying quality and complexity of reference motion data, selecting a single, fixed value for the tracking factor that works for all motion scenarios is impractical. To resolve this, we design an adaptive mechanism that dynamically adjusts σ during training through a feedback loop between error estimation and tracking factor adaptation.

In this mechanism, we maintain an Exponential Moving Average (EMA) $\hat{x}$ of the instantaneous tracking error over environment steps. This EMA serves as an online estimate of the expected tracking error under the current policy, and during training this value should approach the average optimal tracking error $\left(\sum_{i=1}^{N} x_i^* \right) / N$ under the current factor σ. At each step, PBHC updates σ to the current value of $\hat{x}$, creating a feedback loop where reductions in tracking error lead to tightening of σ. This closed-loop process drives further policy refinement, and as the tracking error decreases, the system converges to an optimal value of σ that asymptotically solves Eq. (9), as illustrated in Fig. 3.

To ensure stability during training, we constrain σ to be non-increasing and initialize it with a relatively large value, σ^{init}. The update rule is given by Eq. (8). As shown in Fig. 4, this adaptive mechanism allows the policy to progressively improve its tracking precision during training.

$$\sigma \leftarrow \min(\sigma, \hat{x}). \tag{8}$$

3.3 RL Training Framework

Asymmetric Actor-Critic. Following previous works [6, 21], the time phase variable $\phi_t \in [0, 1]$ is introduced to represent the current progress of the reference motion linearly, where $\phi_t = 0$ denotes the start of a motion and $\phi_t = 1$ denotes the end. The observation of the actor $\boldsymbol{s}_t^{\text{actor}}$ includes the robot's proprioception $\boldsymbol{s}_t^{\text{prop}}$ and the time phase variable ϕ_t. The proprioception $\boldsymbol{s}_t^{\text{prop}} = [\boldsymbol{q}_{t-4:t}, \dot{\boldsymbol{q}}_{t-4:t}, \boldsymbol{\omega}_{t-4:t}^{\text{root}}, \boldsymbol{g}_{t-4:t}^{\text{proj}}, \boldsymbol{a}_{t-5:t-1}]$ includes 5-step history of joint position $\boldsymbol{q}_t \in \mathbb{R}^{23}$, joint

velocity $\dot{\boldsymbol{q}}_t \in \mathbb{R}^{23}$, root angular velocity $\boldsymbol{\omega}_t^{\text{root}} \in \mathbb{R}^3$, root projected gravity $\boldsymbol{g}_t^{\text{proj}} \in \mathbb{R}^3$ and last-step action $\boldsymbol{a}_{t-1} \in \mathbb{R}^{23}$. The critic receives an augmented observation $\boldsymbol{s}_t^{\text{crtic}}$, including $\boldsymbol{s}_t^{\text{prop}}$, time phase, reference motion positions, root linear velocity, and a set of randomized physical parameters.

Reward Vectorization. To facilitate the learning of value function with multiple rewards, we vectorize rewards and value functions as: $\boldsymbol{r} = [r_1, \ldots, r_n]$ and $\boldsymbol{V}(s) = [V_1(\boldsymbol{s}), \ldots, V_n(\boldsymbol{s})]$ following Xie et al. [22]. Rather than aggregating all rewards into a single scalar, each reward component r_i is assigned to a value function $V_i(\boldsymbol{s})$ that independently estimates returns, implemented by a critic network with multiple output heads. All value functions are aggregated to compute the action advantage. This design enables precise value estimation and promotes stable policy optimization.

Reference State Initialization. We use Reference State Initialization (RSI) [21], which initializes the robot's state from reference motion states at randomly sampled time phases. This facilitates parallel learning of different motion phases, significantly improving training efficiency.

Sim-to-Real Transfer. To bridge the sim-to-real gap, we adopt domain randomization by varying the physical parameters of the simulated environment and humanoids. The trained policies are validated through sim-to-sim testing before being directly deployed to real robots, achieving zero-shot sim-to-real transfer without any fine-tuning. Details are in Appendix C.3.

4 Related Works

Humanoid Motion Imitation. Robot motion imitation aims to learn lifelike and natural behaviors from human motions [21, 23]. Although there exist several motion datasets that contain diverse motions [24, 25, 4], humanoid robots cannot directly learn the diverse behaviors due to the significantly different physical structures between humans and humanoid robots [6, 26]. Meanwhile, most datasets lack physical information, such as foot contact annotations that would be important for robot policy learning [27, 28]. As a result, we adopt physics-based motion processing for motion filtering and contact annotation. After obtaining the reference motion, the humanoid robot learns a whole-body control policy to interact with the simulator [29, 30], with the aim of obtaining a state trajectory close to the reference [31, 32]. However, learning such a policy is quite challenging, as the robot requires precise control of high-dimensional DoFs to achieve stable and realistic movement [7, 8]. Recent advances adopt physics-based motion filtering and RL to learn whole-body control policies [5, 10], and perform real-world adaptation via sim-to-real transfer [33]. However, because of the lack of tolerance mechanisms for hard motions, these methods are only capable of tracking relatively simple motions. Other works also combine teleoperation [34, 35] and independent control of upper and lower bodies [36], while they may sacrifice the expressiveness of motions. In contrast, we propose an adaptive mechanism to dynamically adapt the tracking rewards for agile motions.

Humanoid Whole-Body Control. Traditional methods for humanoid robots usually learn independent control policies for locomotion and manipulation. For the lower-body, RL-based controller have been widely adopted to learn locomotion policies for complex tasks such as complex-terrain walking [37, 38], gait control [39], standing up [40, 41], jumping [42], and even parkour [43, 44]. However, each locomotion task requires delicate reward designs, and human-like behaviors are difficult to obtain [45, 46]. In contrast, we adopt human motion as references, which is straightforward for robots to obtain human-like behaviors. For the upper-body, various methods propose different architectures to learn manipulation tasks, such as diffusion policy [47, 48], visual-language-action model [49, 50, 51], dual-system architecture [52, 53], and world models [54, 55]. However, these methods may overlook the coordination of the two limbs. Recently, several whole-body control methods have been proposed, with the aim of enhancing the robustness of entire systems in locomotion [22, 39, 34] or performing loco-manipulation tasks [56]. Differently, the upper and lower bodies of our method have the same objective to track the reference motion, while the lower body still requires maintaining stability and preventing falling in motion imitation. Other methods collect whole-body control datasets to learn a humanoid foundation model [56, 57], while requiring a large number of trajectories. In contrast, we only require a small number of reference motions to learn diverse behaviors.

5 Experiments

In this section, we present experiments to evaluate the effectiveness of PBHC. Our experiments aim to answer the following key research questions:

Figure 5: Example motions in our constructed dataset. Darker opacity indicates later timestamps.

- **Q1.** Can our physics-based motion filtering effectively filter out untrackable motions?
- **Q2.** Does PBHC achieve superior tracking performance compared to prior methods in simulation?
- **Q3.** Does the adaptive motion tracking mechanism improve tracking precision?
- **Q4.** How well does PBHC perform in real-world deployment?

5.1 Experiment Setup

Evaluation Method. We assess the policy's tracking performance using a highly-dynamic motion dataset constructed through our proposed motion processing pipeline, detailed in Appendix B. Examples are shown in Fig. 5. We categorize motions into three difficulty levels: easy, medium, and hard, based on their agility requirements. For each setting, policies are trained in IsaacGym [29] with three random seeds and are evaluated over 1,000 rollout episodes.

Metrics. The tracking performance of polices is quantified through the following metrics: Global Mean Per Body Position Error ($E_{\text{g-mpbpe}}$, mm), root-relative Mean Per Body Position Error (E_{mpbpe}, mm), Mean Per Joint Position Error (E_{mpjpe}, 10^{-3} rad), Mean Per Joint Velocity Error (E_{mpjve}, 10^{-3} rad/frame), Mean Per Body Velocity Error (E_{mpbve}, mm/frame), and Mean Per Body Acceleration Error (E_{mpbae}, mm/frame^2). The definition of metrics is given in Appendix D.2.

5.2 Motion Filtering

To address **Q1**, we apply our physics-based motion filtering method (see §3.1) to 10 motion sequences. Among them, 4 sequences are rejected based on the filtering criteria, while the remaining 6 are accepted. To evaluate the effectiveness of the filtering, we train a separate policy for each motion and compute the Episode Length Ratio (ELR), defined as the ratio of average episode length to reference motion length.

As shown in Fig. 6, accepted motions consistently achieve high ELRs, demonstrating motions that satisfy the physics-based metric can lead to better performance in motion tracking. In contrast, rejected motions achieve a maximum ELR of only 54%, suggesting frequent violations of termination conditions. These results demonstrate that our filtering method effectively excludes inherently untrackable motions, thereby improving efficiency by focusing on viable candidates.

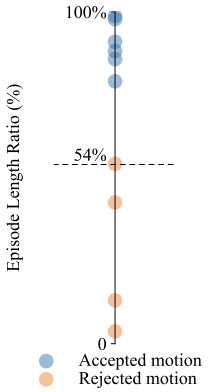

Figure 6: The distribution of ELR of accepted and rejected motions.

5.3 Main Result

To address **Q2**, we compare PBHC with three baseline methods: OmniH2O [10], Exbody2 [5], and MaskedMimic [23]. All baselines employ the exponential form of the reward function for tracking reference motion, as described in §3.2.1. Implementation details are provided in Appendix D.3.

As shown in Table 1, PBHC consistently outperforms the baselines OmniH2O and ExBody2 across all evaluation metrics. These improvements can be attributed to our adaptive motion tracking mechanism, which automatically adjusts tracking factors based on motion characteristics, whereas the fixed, empirically tuned parameters in the baselines fail to generalize across diverse motions. While

Table 1: Main results comparing different methods across difficulty levels. PBHC consistently outperforms deployable baselines and approaches oracle-level performance. Results are reported as mean ± one standard deviation. Bold indicates methods within one standard deviation of the best result, excluding MaskedMimic.

Method	$E_{\text{g-mpbpe}} \downarrow$	$E_{\text{mpbpe}} \downarrow$	$E_{\text{mpjpe}} \downarrow$	$E_{\text{mpbve}} \downarrow$	$E_{\text{mpbae}} \downarrow$	$E_{\text{mpjve}} \downarrow$
Easy						
OmniH2O	$233.54_{\pm 4.013}$	$103.67_{\pm 1.912}$	$1805.10_{\pm 12.33}$	$8.54_{\pm 0.125}$	$8.46_{\pm 0.081}$	$224.70_{\pm 2.043}$
ExBody2	$588.22_{\pm 11.43}$	$332.50_{\pm 3.584}$	$4014.40_{\pm 21.50}$	$14.29_{\pm 0.172}$	$9.80_{\pm 0.157}$	$206.01_{\pm 1.346}$
Ours	$\mathbf{53.25}_{\pm 17.60}$	$\mathbf{28.16}_{\pm 6.127}$	$\mathbf{725.62}_{\pm 16.20}$	$\mathbf{4.41}_{\pm 0.312}$	$\mathbf{4.65}_{\pm 0.140}$	$\mathbf{81.28}_{\pm 2.052}$
MaskedMimic (Oracle)	$41.79_{\pm 1.715}$	$21.86_{\pm 2.030}$	$739.96_{\pm 19.94}$	$5.20_{\pm 0.245}$	$7.40_{\pm 0.333}$	$132.01_{\pm 8.941}$
Medium						
OmniH2O	$433.64_{\pm 16.22}$	$151.42_{\pm 7.340}$	$2333.90_{\pm 49.50}$	$10.85_{\pm 0.300}$	$10.54_{\pm 0.152}$	$204.36_{\pm 4.473}$
ExBody2	$619.84_{\pm 26.16}$	$261.01_{\pm 1.592}$	$3738.70_{\pm 26.90}$	$14.48_{\pm 0.160}$	$11.25_{\pm 0.173}$	$204.33_{\pm 2.172}$
Ours	$\mathbf{126.48}_{\pm 27.01}$	$\mathbf{48.87}_{\pm 7.550}$	$\mathbf{1043.30}_{\pm 104.4}$	$\mathbf{6.62}_{\pm 0.412}$	$\mathbf{7.19}_{\pm 0.254}$	$\mathbf{105.30}_{\pm 5.941}$
MaskedMimic (Oracle)	$150.92_{\pm 133.4}$	$61.69_{\pm 46.01}$	$934.25_{\pm 155.0}$	$8.16_{\pm 1.974}$	$10.01_{\pm 0.883}$	$176.84_{\pm 26.14}$
Hard						
OmniH2O	$446.17_{\pm 12.84}$	$\mathbf{147.88}_{\pm 4.142}$	$1939.50_{\pm 23.90}$	$14.98_{\pm 0.643}$	$\mathbf{14.40}_{\pm 0.580}$	$190.13_{\pm 8.211}$
ExBody2	$689.68_{\pm 11.80}$	$246.40_{\pm 1.252}$	$4037.40_{\pm 16.70}$	$19.90_{\pm 0.210}$	$16.72_{\pm 0.160}$	$254.76_{\pm 3.409}$
Ours	$\mathbf{290.36}_{\pm 139.1}$	$\mathbf{124.61}_{\pm 53.54}$	$\mathbf{1326.60}_{\pm 378.9}$	$\mathbf{11.93}_{\pm 2.622}$	$\mathbf{12.36}_{\pm 2.401}$	$\mathbf{135.05}_{\pm 16.43}$
MaskedMimic (Oracle)	$47.74_{\pm 2.762}$	$27.25_{\pm 1.615}$	$829.02_{\pm 15.41}$	$8.33_{\pm 0.194}$	$10.60_{\pm 0.420}$	$146.90_{\pm 13.32}$

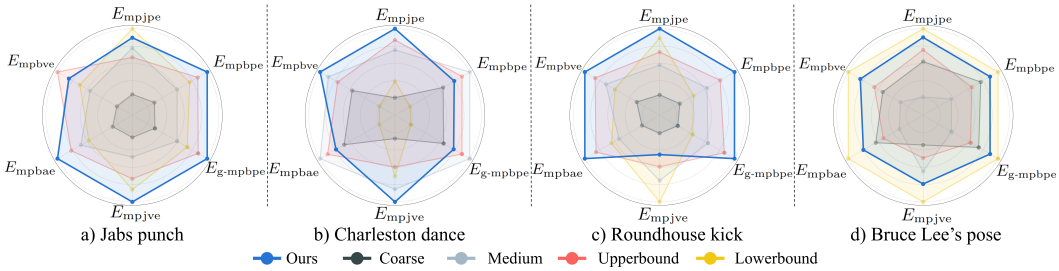

Figure 7: Ablation study comparing the adaptive motion tracking mechanism with fixed tracking factor variants. The adaptive mechanism consistently achieves near-optimal performance across all motions, whereas fixed variants exhibit varying performance depending on motions.

MaskedMimic performs well on certain metrics, it is primarily designed for character animation and is not deployable for robot control, as it does not account for constraints such as partial observability and action smoothness. Therefore, we treat it as an oracle-style lower bound rather than a directly comparable baseline.

5.4 Impact of Adaptive Motion Tracking Mechanism

To investigate **Q3**, we conduct an ablation study evaluating our adaptive motion tracking mechanism (§3.2) against four baseline configurations with fixed tracking factor set: *Coarse, Medium, UpperBound, LowerBound*. The tracking factors in *Coarse*, *Medium*, *UpperBound*, and *LowerBound* are roughly progressively smaller, with *LowerBound* approximately corresponding to the smallest tracking factor derived from the adaptive mechanism after training convergence, while *UpperBound* approximately corresponds to the largest. The specific configuration of baselines and the converged tracking factors of the adaptive mechanism are given in Appendix D.4.

As shown in Fig. 7, the performance of the fixed tracking factor configurations (*Coarse, Medium, LowerBound* and *UpperBound*) varies between different motion types. Specifically, while *LowerBound* and *UpperBound* achieve strong performance on certain motions, they perform suboptimally on others, indicating that no single fixed setting consistently yields optimal tracking results on all motions. In contrast, our adaptive motion tracking mechanism consistently achieves near-optimal performance across all motion types, demonstrating its effectiveness in dynamically adjusting the tracking factor to suit varying motion characteristics.

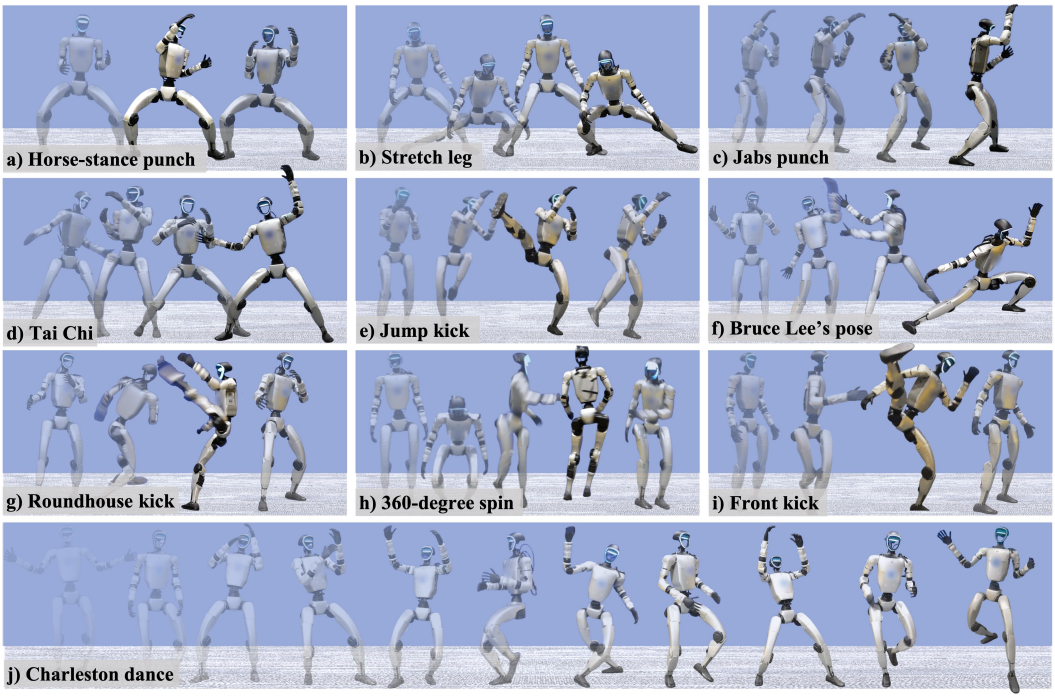

Figure 8: Our robot masters highly-dynamic skills in the real world. Time flows left to right.

5.5 Real-World Deployment

As shown in Fig. 8, 11 and the supporting videos, our robot in real world demonstrates outstanding dynamic capabilities through a diverse repertoire of advanced skills: (1) sophisticated martial arts techniques including powerful boxing combinations (jabs, hooks, and horse-stance punches) and high-degree kicking maneuvers (front kicks, jump kicks, side kicks, back kicks, and spinning roundhouse kicks); (2) acrobatic movements such as full 360-degree spins; (3) flexible motions including deep squats and stretches; (4) artistic performances ranging from dynamic dance routines to graceful Tai Chi sequences. This comprehensive skill set highlights our system's remarkable versatility, dynamic control, and real-world applicability across both athletic and artistic domains.

To quantitatively assess our policy's tracking performance, we conduct 10 trials of the Tai Chi motion and compute evaluation metrics based on the onboard sensor readings, as shown in Table 2. Notably, the metrics obtained in the real world are closely aligned with those from the sim-to-sim platform MuJoCo, demonstrating that our policy can robustly transfer from simulation to real-world deployment while maintaining high-performance control.

Table 2: Comparison of tracking performance of Tai Chi between real-world and simulation. The robot root is fixed to the origin since it's inaccessible in real-world.

Platform	$E_{\text{mpbpe}} \downarrow$	$E_{\text{mpjpe}} \downarrow$	$E_{\text{mpbve}} \downarrow$	$E_{\text{mpbae}} \downarrow$	$E_{\text{mpjve}} \downarrow$
MuJoCo	$33.18_{\pm 2.720}$	$1061.24_{\pm 83.27}$	$2.96_{\pm 0.342}$	$2.90_{\pm 0.498}$	$67.71_{\pm 6.747}$
Real	$36.64_{\pm 2.592}$	$1130.05_{\pm 9.478}$	$3.01_{\pm 0.126}$	$3.12_{\pm 0.056}$	$65.68_{\pm 1.972}$

6 Conclusion & Limitations

This paper introduces PBHC, a novel RL framework for humanoid whole-body motion control that achieves outstanding highly-dynamic behaviors and superior tracking accuracy through physics-based motion processing and adaptive motion tracking. The experiments show the motion filtering metric can efficiently filter out trajectories that are difficult to track, and the adaptive motion tracking method consistently outperforms baseline methods on tracking error. The real-world deployments demonstrate

robust behaviors for athletic and artistic domains. These contributions push the boundaries of humanoid motion control, paving the way for more agile and stable real-world applications.

However, our method still has limitations. (i) It lacks environment awareness, such as terrain perception and obstacle avoidance, which restricts deployment in unstructured real-world settings. (ii) Each policy is trained to imitate a single motion, which may not be efficient for applications requiring diverse motion repertoires. We leave research on how to maintain high dynamic performance while enabling broader skill generalization for the future.

Acknowledgments and Disclosure of Funding

This work is supported by the National Natural Science Foundation of China (Grant No.62306242), the Young Elite Scientists Sponsorship Program by CAST (Grant No. 2024QNRC001), and the Yangfan Project of the Shanghai (Grant No.23YF11462200).

References

[1] Zhaoyuan Gu, Junheng Li, Wenlan Shen, Wenhao Yu, Zhaoming Xie, Stephen McCrory, Xianyi Cheng, Abdulaziz Shamsah, Robert Griffin, C Karen Liu, et al. Humanoid locomotion and manipulation: Current progress and challenges in control, planning, and learning. *arXiv preprint arXiv:2501.02116*, 2025.

[2] Biao Jiang, Xin Chen, Wen Liu, Jingyi Yu, Gang Yu, and Tao Chen. Motiongpt: Human motion as a foreign language. *Advances in Neural Information Processing Systems*, 36:20067–20079, 2023.

[3] Zan Wang, Yixin Chen, Baoxiong Jia, Puhao Li, Jinlu Zhang, Jingze Zhang, Tengyu Liu, Yixin Zhu, Wei Liang, and Siyuan Huang. Move as you say interact as you can: Language-guided human motion generation with scene affordance. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 433–444, 2024.

[4] Naureen Mahmood, Nima Ghorbani, Nikolaus F Troje, Gerard Pons-Moll, and Michael J Black. Amass: Archive of motion capture as surface shapes. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5442–5451, 2019.

[5] Mazeyu Ji, Xuanbin Peng, Fangchen Liu, Jialong Li, Ge Yang, Xuxin Cheng, and Xiaolong Wang. Exbody2: Advanced expressive humanoid whole-body control. *arXiv preprint arXiv:2412.13196*, 2024.

[6] Tairan He, Jiawei Gao, Wenli Xiao, Yuanhang Zhang, Zi Wang, Jiashun Wang, Zhengyi Luo, Guanqi He, Nikhil Sobanbab, Chaoyi Pan, et al. Asap: Aligning simulation and real-world physics for learning agile humanoid whole-body skills. *arXiv preprint arXiv:2502.01143*, 2025.

[7] Xuxin Cheng, Yandong Ji, Junming Chen, Ruihan Yang, Ge Yang, and Xiaolong Wang. Expressive whole-body control for humanoid robots. In *Robotics: Science and Systems*, 2024.

[8] Zipeng Fu, Qingqing Zhao, Qi Wu, Gordon Wetzstein, and Chelsea Finn. Humanplus: Humanoid shadowing and imitation from humans. In *8th Annual Conference on Robot Learning*, 2024.

[9] Tairan He, Zhengyi Luo, Wenli Xiao, Chong Zhang, Kris Kitani, Changliu Liu, and Guanya Shi. Learning human-to-humanoid real-time whole-body teleoperation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2024.

[10] Tairan He, Zhengyi Luo, Xialin He, Wenli Xiao, Chong Zhang, Weinan Zhang, Kris M Kitani, Changliu Liu, and Guanya Shi. Omnih2o: Universal and dexterous human-to-humanoid whole-body teleoperation and learning. In *8th Annual Conference on Robot Learning*, 2024.

[11] Yihua Zhang, Prashant Khanduri, Ioannis Tsaknakis, Yuguang Yao, Mingyi Hong, and Sijia Liu. An introduction to bilevel optimization: Foundations and applications in signal processing and machine learning. *IEEE Signal Processing Magazine*, 41(1):38–59, 2024.

[12] Unitree Robotics. Humanoid robot G1_Humanoid Robot Functions_Humanoid Robot Price | Unitree Robotics, 2025. `https://www.unitree.com/g1/`.

[13] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[14] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. SMPL: A skinned multi-person linear model. *ACM Trans. Graphics (Proc. SIGGRAPH Asia)*, 34(6):248:1–248:16, October 2015.

[15] Zehong Shen, Huaijin Pi, Yan Xia, Zhi Cen, Sida Peng, Zechen Hu, Hujun Bao, Ruizhen Hu, and Xiaowei Zhou. World-grounded human motion recovery via gravity-view coordinates. In *SIGGRAPH Asia Conference Proceedings*, 2024.

[16] Shashank Tripathi, Lea Müller, Chun-Hao P Huang, Omid Taheri, Michael J Black, and Dimitrios Tzionas. 3d human pose estimation via intuitive physics. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4713–4725, 2023.

[17] Chun-Hao P. Huang, Hongwei Yi, Markus Höschle, Matvey Safroshkin, Tsvetelina Alexiadis, Senya Polikovsky, Daniel Scharstein, and Michael J. Black. Capturing and inferring dense full-body human-scene contact. In *IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 13274–13285, June 2022.

[18] Yuliang Zou, Jimei Yang, Duygu Ceylan, Jianming Zhang, Federico Perazzi, and Jia-Bin Huang. Reducing footskate in human motion reconstruction with ground contact constraints. In *Winter Conference on Applications of Computer Vision*, 2020.

[19] Kevin Zakka. Mink: Python inverse kinematics based on MuJoCo, July 2024. `https://github.com/kevinzakka/mink`.

[20] Félix G. Harvey, Mike Yurick, Derek Nowrouzezahrai, and Christopher Pal. Robust motion in-betweening. *ACM Trans. Graph.*, 39(4), 2020.

[21] Xue Bin Peng, Pieter Abbeel, Sergey Levine, and Michiel Van de Panne. Deepmimic: Example-guided deep reinforcement learning of physics-based character skills. *ACM Transactions On Graphics (TOG)*, 37(4):1–14, 2018.

[22] Weiji Xie, Chenjia Bai, Jiyuan Shi, Junkai Yang, Yunfei Ge, Weinan Zhang, and Xuelong Li. Humanoid whole-body locomotion on narrow terrain via dynamic balance and reinforcement learning. *arXiv preprint arXiv:2502.17219*, 2025.

[23] Chen Tessler, Yunrong Guo, Ofir Nabati, Gal Chechik, and Xue Bin Peng. Maskedmimic: Unified physics-based character control through masked motion inpainting. *ACM Transactions on Graphics (TOG)*, 2024.

[24] Yang Gao, Po-Chien Luan, and Alexandre Alahi. Multi-transmotion: Pre-trained model for human motion prediction. In *8th Annual Conference on Robot Learning*, 2024.

[25] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE transactions on pattern analysis and machine intelligence*, 36(7):1325–1339, 2013.

[26] Zhengyi Luo, Jinkun Cao, Kris Kitani, Weipeng 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.

[27] He Zhang, Shenghao Ren, Haolei Yuan, Jianhui Zhao, Fan Li, Shuangpeng Sun, Zhenghao Liang, Tao Yu, Qiu Shen, and Xun Cao. Mmvp: A multimodal mocap dataset with vision and pressure sensors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21842–21852, 2024.

[28] Davis Rempe, Leonidas J Guibas, Aaron Hertzmann, Bryan Russell, Ruben Villegas, and Jimei Yang. Contact and human dynamics from monocular video. In *European Conference on Computer Vision (ECCV)*, pages 71–87. Springer, 2020.

[29] Viktor Makoviychuk, Lukasz Wawrzyniak, Yunrong Guo, Michelle Lu, Kier Storey, Miles Macklin, David Hoeller, Nikita Rudin, Arthur Allshire, Ankur Handa, et al. Isaac gym: High performance gpu-based physics simulation for robot learning. *arXiv preprint arXiv:2108.10470*, 2021.

[30] Nikita Rudin, David Hoeller, Philipp Reist, and Marco Hutter. Learning to walk in minutes using massively parallel deep reinforcement learning. In *Conference on Robot Learning*, pages 91–100, 2022.

[31] Sirui Xu, Hung Yu Ling, Yu-Xiong Wang, and Liang-Yan Gui. Intermimic: Towards universal whole-body control for physics-based human-object interactions. *arXiv preprint arXiv:2502.20390*, 2025.

[32] Yinhuai Wang, Qihang Zhao, Runyi Yu, Hok Wai Tsui, Ailing Zeng, Jing Lin, Zhengyi Luo, Jiwen Yu, Xiu Li, Qifeng Chen, Jian Zhang, Lei Zhang, and Ping Tan. Skillmimic: Learning basketball interaction skills from demonstrations. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

[33] Haoran He, Peilin Wu, Chenjia Bai, Hang Lai, Lingxiao Wang, Ling Pan, Xiaolin Hu, and Weinan Zhang. Bridging the sim-to-real gap from the information bottleneck perspective. In *Annual Conference on Robot Learning*, 2024.

[34] Qingwei Ben, Feiyu Jia, Jia Zeng, Junting Dong, Dahua Lin, and Jiangmiao Pang. Homie: Humanoid loco-manipulation with isomorphic exoskeleton cockpit. In *Robotics: Science and Systems*, 2025.

[35] Yanjie Ze, Zixuan Chen, JoÃ̧o Pedro AraÃšjo, Zi-ang Cao, Xue Bin Peng, Jiajun Wu, and C Karen Liu. Twist: Teleoperated whole-body imitation system. *arXiv preprint arXiv:2505.02833*, 2025.

[36] Chenhao Lu, Xuxin Cheng, Jialong Li, Shiqi Yang, Mazeyu Ji, Chengjing Yuan, Ge Yang, Sha Yi, and Xiaolong Wang. Mobile-television: Predictive motion priors for humanoid whole-body control. In *IEEE International Conference on Robotics and Automation*, 2025.

[37] Huayi Wang, Zirui Wang, Junli Ren, Qingwei Ben, Jiangmiao Pang, Tao Huang, and Weinan Zhang. Beamdojo: Learning agile humanoid locomotion on sparse footholds. *arXiv preprint arXiv:2502.10363*, 2024.

[38] Xinyang Gu, Yen-Jen Wang, and Jianyu Chen. Humanoid-gym: Reinforcement learning for humanoid robot with zero-shot sim2real transfer. *arXiv preprint arXiv:2404.05695*, 2024.

[39] Yufei Xue, Wentao Dong, Minghuan Liu, Weinan Zhang, and Jiangmiao Pang. A unified and general humanoid whole-body controller for fine-grained locomotion. *arXiv preprint arXiv:2502.03206*, 2025.

[40] Xialin He, Runpei Dong, Zixuan Chen, and Saurabh Gupta. Learning getting-up policies for real-world humanoid robots. *arXiv preprint arXiv:2502.12152*, 2025.

[41] Tao Huang, Junli Ren, Huayi Wang, Zirui Wang, Qingwei Ben, Muning Wen, Xiao Chen, Jianan Li, and Jiangmiao Pang. Learning humanoid standing-up control across diverse postures. *arXiv preprint arXiv:2502.08378*, 2025.

[42] Zhongyu Li, Xue Bin Peng, Pieter Abbeel, Sergey Levine, Glen Berseth, and Koushil Sreenath. Robust and versatile bipedal jumping control through reinforcement learning. *arXiv preprint arXiv:2302.09450*, 2023.

[43] Junfeng Long, Junli Ren, Moji Shi, Zirui Wang, Tao Huang, Ping Luo, and Jiangmiao Pang. Learning humanoid locomotion with perceptive internal model. *arXiv preprint arXiv:2411.14386*, 2024.

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

[45] Xue Bin Peng, Erwin Coumans, Tingnan Zhang, Tsang-Wei Lee, Jie Tan, and Sergey Levine. Learning agile robotic locomotion skills by imitating animals. *Robotics: Science and Systems (RSS), Virtual Event/Corvalis, July*, pages 12–16, 2020.

[46] Xue Bin Peng, Ze Ma, Pieter Abbeel, Sergey Levine, and Angjoo Kanazawa. Amp: Adversarial motion priors for stylized physics-based character control. *ACM Transactions on Graphics (ToG)*, 40(4), 2021.

[47] Haoran He, Chenjia Bai, Kang Xu, Zhuoran Yang, Weinan Zhang, Dong Wang, Bin Zhao, and Xuelong Li. Diffusion model is an effective planner and data synthesizer for multi-task reinforcement learning. *Advances in Neural Information Processing Systems*, 36:64896–64917, 2023.

[48] Songming Liu, Lingxuan Wu, Bangguo Li, Hengkai Tan, Huayu Chen, Zhengyi Wang, Ke Xu, Hang Su, and Jun Zhu. Rdt-1b: a diffusion foundation model for bimanual manipulation. *arXiv preprint arXiv:2410.07864*, 2024.

[49] Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024.

[50] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, et al. Rt-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*, 2022.

[51] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.

[52] AgiBot-World-Contributors. Agibot world colosseo: A large-scale manipulation platform for scalable and intelligent embodied systems, 2025.

[53] Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, et al. Gr00t n1: An open foundation model for generalist humanoid robots. *arXiv preprint arXiv:2503.14734*, 2025.

[54] Haoran He, Chenjia Bai, Ling Pan, Weinan Zhang, Bin Zhao, and Xuelong Li. Learning an actionable discrete diffusion policy via large-scale actionless video pre-training. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[55] Yucheng Hu, Yanjiang Guo, Pengchao Wang, Xiaoyu Chen, Yen-Jen Wang, Jianke Zhang, Koushil Sreenath, Chaochao Lu, and Jianyu Chen. Video prediction policy: A generalist robot policy with predictive visual representations. In *International Conference on Machine Learning*, 2025.

[56] Jiyuan Shi, Xinzhe Liu, Dewei Wang, Ouyang Lu, Sören Schwertfeger, Fuchun Sun, Chenjia Bai, and Xuelong Li. Adversarial locomotion and motion imitation for humanoid policy learning. *arXiv preprint arXiv:2504.14305*, 2025.

[57] Jiageng Mao, Siheng Zhao, Siqi Song, Tianheng Shi, Junjie Ye, Mingtong Zhang, Haoran Geng, Jitendra Malik, Vitor Guizilini, and Yue Wang. Learning from massive human videos for universal humanoid pose control. *arXiv preprint arXiv:2412.14172*, 2024.

[58] Luigi Campanaro, Siddhant Gangapurwala, Wolfgang Merkt, and Ioannis Havoutis. Learning and deploying robust locomotion policies with minimal dynamics randomization. In *6th Annual Learning for Dynamics & Control Conference*, pages 578–590. PMLR, 2024.

A Derivation of Optimal Tracking Sigma

We recall the bi-level optimization problem in (6), as

$$\max_{\sigma \in \mathbb{R}_+} \quad J^{\text{ex}}(\boldsymbol{x}^*) \tag{9a}$$

$$\text{s.t.} \quad \boldsymbol{x}^* \in \arg\max_{\boldsymbol{x} \in \mathbb{R}_+^N} J^{\text{in}}(\boldsymbol{x}, \sigma) + R(\boldsymbol{x}) \tag{9b}$$

Assuming $R(\boldsymbol{x})$ takes a linear form $R(\boldsymbol{x}) = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}$, J^{ex}, and J^{in} are twice continuously differentiable and the lower-level problem Eq. (9b) has a unique solution $\boldsymbol{x}^*(\sigma)$. Then we take an implicit gradient approach to solve it. The gradient of J^{ex} w.r.t. σ is:

$$\frac{dJ^{\text{ex}}}{d\sigma} = \frac{d\boldsymbol{x}^*(\sigma)}{d\sigma}^\top \nabla_{\boldsymbol{x}} J^{\text{ex}}(\boldsymbol{x}^*(\sigma)). \tag{10}$$

To obtain $\frac{d\boldsymbol{x}^*(\sigma)}{d\sigma}$, since $\boldsymbol{x}^*(\sigma)$ is a lower-level solution, it satisfies:

$$\nabla_{\boldsymbol{x}}(J^{\text{in}}(\boldsymbol{x}^*(\sigma), \sigma) + R(\boldsymbol{x})) = 0. \tag{11}$$

Take the first-order derivative of Eq. (11) w.r.t. σ, then we have:

$$\frac{d}{d\sigma}(\nabla_{\boldsymbol{x}}(J^{\text{in}}(\boldsymbol{x}^*(\sigma), \sigma) + R(\boldsymbol{x})) = \nabla^2_{\sigma,\boldsymbol{x}} J^{\text{in}} + \frac{d\boldsymbol{x}^*(\sigma)}{d\sigma}^\top \nabla^2_{\boldsymbol{x},\boldsymbol{x}} J^{\text{in}} = 0, \tag{12}$$

$$\frac{d\boldsymbol{x}^*(\sigma)}{d\sigma}^\top = -\nabla^2_{\sigma,\boldsymbol{x}} J^{\text{in}}(\boldsymbol{x}^*(\sigma), \sigma) \nabla^2_{\boldsymbol{x},\boldsymbol{x}} J^{\text{in}}(\boldsymbol{x}^*(\sigma), \sigma)^{-1}. \tag{13}$$

Substituting Eq. (13) into Eq. (10), we have

$$\frac{dJ^{\text{ex}}}{d\sigma} = -\nabla^2_{\sigma,\boldsymbol{x}} J^{\text{in}}(\boldsymbol{x}^*(\sigma), \sigma) \nabla^2_{\boldsymbol{x},\boldsymbol{x}} J^{\text{in}}(\boldsymbol{x}^*(\sigma), \sigma)^{-1} \nabla_{\boldsymbol{x}} J^{\text{ex}}(\boldsymbol{x}^*(\sigma)), \tag{14}$$

where

$$J^{\text{ex}}(\boldsymbol{x}) = \sum_{i=1}^{N} -x_i, \tag{15a}$$

$$J^{\text{in}}(\boldsymbol{x}, \sigma) = \sum_{i=1}^{N} \exp(-x_i/\sigma). \tag{15b}$$

Compute first- and second-order gradients in Eq. (14) as

$$\nabla_{\boldsymbol{x}} J^{\text{in}}(\boldsymbol{x}, \sigma) = \exp(-\boldsymbol{x}/\sigma)(-\frac{1}{\sigma}), \tag{16a}$$

$$\nabla_{\boldsymbol{x}} J^{\text{ex}}(\boldsymbol{x}) = \mathbf{1}, \tag{16b}$$

$$\nabla^2_{\sigma,\boldsymbol{x}} J^{\text{in}}(\boldsymbol{x}, \sigma) = \frac{\sigma - \boldsymbol{x}}{\sigma^3} \odot \exp(-\boldsymbol{x}/\sigma), \tag{16c}$$

$$\nabla^2_{\boldsymbol{x},\boldsymbol{x}} J^{\text{in}}(\boldsymbol{x}, \sigma) = \text{diag}(\exp(-\boldsymbol{x}/\sigma))/\sigma^2, \tag{16d}$$

where $\odot$ means element-wise multiplication. Substituting (16) into (14) and let the gradient equals to zero $\frac{dJ^{\text{ex}}}{d\sigma} = 0$, then we have

$$\sigma = \frac{\sum_{i=1}^{N} x_i^*(\sigma)}{N}. \tag{17}$$

B Dataset Description

Our dataset integrates motions from: (i) video-based sources, from which motion data is extracted through our proposed multi-steps motion processing pipeline. The hyperparameters of the pipeline are listed in Table 3; (ii)

open-source datasets: selected motions from AMASS and LAFAN. The dataset comprises 13 distinct motions, which are categorized into three difficulty levels—easy, medium, and hard. To ensure smooth transitions, we linearly interpolate at the beginning and end of each sequence to move from a default pose to the reference motion and back. The details are given in Table 4.

Table 3: Hyperparameters of multi-steps motion processing.

Hyperparameter	Value
ϵ_{stab}	0.1
ϵ_{N}	100
ϵ_{vel}	0.002
ϵ_{height}	0.2

Table 4: The details of the highly-dynamic motion dataset.

Motion name	Motion frames	Source
Easy		
Jabs punch	285	video
Hooks punch	175	video
Horse-stance pose	210	LAFAN
Horse-stance punch	200	video
Medium		
Stretch leg	320	video
Tai Chi	500	video
Jump kick	145	video
Charleston dance	610	LAFAN
Bruce Lee's pose	330	AMASS
Hard		
Roundhouse kick	158	AMASS
360-degree spin	180	video
Front kick	155	video
Side kick	179	AMASS

C Algorithm Design

C.1 Observation Space Design

- **Actor observation space:** The actor's observation $\boldsymbol{s}_t^{\text{actor}}$ includes 5-step history of the robot's proprioceptive state $\boldsymbol{s}_t^{\text{prop}}$ and the time-phase variable ϕ_t.
- **Critic observation space:** The critic's observation $\boldsymbol{s}_t^{\text{crtic}}$ additionally includes the base linear velocity, the body position of the reference motion, the difference between the current and reference body positions, and a set of domain-randomized physical parameters. The details are given in Table 5.

Table 5: Actor and critic observation state space.

State term	Actor Dim	Critic Dim
Joint position	23×5	23×5
Joint velocity	23×5	23×5
Root angular velocity	3×5	3×5
Root projected gravity	3×5	3×5
Reference motion phase	1×5	1×5
Actions	23×5	23×5
Root linear velocity	–	3×5
Reference body position	–	81
Body position difference	–	81
Randomized base CoM offset*	–	3
Randomized link mass*	–	22
Randomized stiffness*	–	23
Randomized damping*	–	23
Randomized friction coefficient*	–	1
Randomized control delay*	–	1
Total dim	380	630

**Several randomized physical parameters used in domain randomization are part of the critic observation to improve value estimation robustness. The detailed settings of domain randomization are given in Appendix C.3.*

C.2 Reward Design

All reward functions are detailed in Table 6. Our reward design consists of two main parts: task rewards and regularization rewards. Specifically, we impose penalties when joint position exceeds the soft limits, which are symmetrically scaled from the hard limits by a fixed ratio ($\alpha = 0.95$):

$$\boldsymbol{m} = (\boldsymbol{q}_{\min} + \boldsymbol{q}_{\max})/2, \tag{18a}$$

$$\boldsymbol{d} = \boldsymbol{q}_{\max} - \boldsymbol{q}_{\min}, \tag{18b}$$

$$\boldsymbol{q}_{\text{soft-min}} = \boldsymbol{m} - 0.5 \cdot \boldsymbol{d} \cdot \alpha, \tag{18c}$$

$$\boldsymbol{q}_{\text{soft-max}} = \boldsymbol{m} + 0.5 \cdot \boldsymbol{d} \cdot \alpha, \tag{18d}$$

where $\boldsymbol{q}$ is the joint position. The same procedure is applied to compute the soft limits for joint velocity $\dot{\boldsymbol{q}}$ and torque $\boldsymbol{\tau}$.

Table 6: Reward terms and weights.

Term	Expression	Weight
	Task	
Joint position	$\exp(-\lVert\boldsymbol{q}_t - \hat{\boldsymbol{q}}_t\rVert_2^2/\sigma_{\text{jpos}})$	1.0
Joint velocity	$\exp(-\lVert\dot{\boldsymbol{q}}_t - \hat{\dot{\boldsymbol{q}}}_t\rVert_2^2/\sigma_{\text{jvel}})$	1.0
Body position	$\exp(-\lVert\boldsymbol{p}_t - \hat{\boldsymbol{p}}_t\rVert_2^2/\sigma_{\text{pos}})$	1.0
Body rotation	$\exp(-\lVert\boldsymbol{\theta}_t \ominus \hat{\boldsymbol{\theta}}_t\rVert_2^2/\sigma_{\text{rot}})$	0.5
Body velocity	$\exp(-\lVert\boldsymbol{v}_t - \hat{\boldsymbol{v}}_t\rVert_2^2/\sigma_{\text{vel}})$	0.5
Body angular velocity	$\exp(-\lVert\boldsymbol{\omega}_t - \hat{\boldsymbol{\omega}}_t\rVert_2^2/\sigma_{\text{ang}})$	0.5
Body position VR 3 points	$\exp(-\lVert\boldsymbol{p}_t^{\text{yr}} - \hat{\boldsymbol{p}}_t^{\text{yr}}\rVert_2^2/\sigma_{\text{pos\_vr}})$	1.6
Body position feet	$\exp\left(-\lVert\boldsymbol{p}_t^{\text{feet}} - \hat{\boldsymbol{p}}_t^{\text{feet}}\rVert_2^2/\sigma_{\text{pos\_feet}}\right)$	1.0
Max Joint position	$\exp\left(-\lVert\boldsymbol{q}_t - \hat{\boldsymbol{q}}_t\rVert_\infty \,/\, \sigma_{\text{max\_jpos}}\right)$	1.0
Contact Mask	$1 - \lVert c_t - \hat{c}_t\rVert_1/2$	0.5
	Regularization	
Joint position limits	$\mathbb{I}(\boldsymbol{q} \notin [\boldsymbol{q}_{\text{soft-min}},\ \boldsymbol{q}_{\text{soft-max}}])$	-10.0
Joint velocity limits	$\mathbb{I}(\dot{\boldsymbol{q}} \notin [\dot{\boldsymbol{q}}_{\text{soft-min}},\ \dot{\boldsymbol{q}}_{\text{soft-max}}])$	-5.0
Joint torque limits	$\mathbb{I}(\boldsymbol{\tau} \notin [\boldsymbol{\tau}_{\text{soft-min}},\ \boldsymbol{\tau}_{\text{soft-max}}])$	-5.0
Slippage	$\lVert\boldsymbol{v}_{\text{xy}}^{\text{feet}}\rVert_2^2 \cdot \mathbb{I}[\lVert F^{\text{feet}}\rVert_2 \geq 1]$	-1.0
Feet contact forces	$\min(\lVert F^{\text{feet}} - 400\rVert_2^2, 0)$	-0.01
Feet air time[30]	$\mathbb{I}[T_{\text{air}} > 0.3]$	-1.0
Stumble	$\mathbb{I}[\lVert\boldsymbol{F}_{xy}^{\text{feet}}\rVert > 5 \cdot F_z^{\text{feet}}]$	-2.0
Torque	$\lVert\boldsymbol{\tau}\rVert_2^2$	-1e-6
Action rate	$\lVert\boldsymbol{a}_t - \boldsymbol{a}_{t-1}\rVert_2^2$	-0.02
Collision	$\mathbb{I}_{\text{collision}}$	-30
Termination	$\mathbb{I}_{\text{termination}}$	-200

C.3 Domain Randomization

To improve the transferability of our trained polices to real-world settings, we incorporate domain randomization during training to support robust sim-to-sim and sim-to-real transfer. The specific settings are given in Table 7.

C.4 PPO Hyperparameter

The detailed PPO hyperparameters are shown in Table 8.

Table 7: Domain randomization settings.

Term	Value
Dynamics Randomization	
Friction	$\mathcal{U}(0.2,\ 1.2)$
PD gain	$\mathcal{U}(0.9,\ 1.1)$
Link mass(kg)	$\mathcal{U}(0.9,\ 1.1)\times$ default
Ankle inertia(kg·m^2)	$\mathcal{U}(0.9,\ 1.1)\times$ default
Base CoM offset(m)	$\mathcal{U}(-0.05,\ 0.05)$
ERFI[58](N·m/kg)	$0.05\times$ torque limit
Control delay(ms)	$\mathcal{U}(0,\ 40)$
External Perturbation	
Random push interval(s)	$[5,\ 10]$
Random push velocity(m/s)	0.1

Table 8: Hyperparameters related to PPO.

Hyperparameter	Value
Optimizer	Adam
Batch size	4096
Mini Batches	4
Learning epoches	5
Entropy coefficient	0.01
Value loss coefficient	1.0
Clip param	0.2
Max grad norm	1.0
Init noise std	0.8
Learning rate	1e-3
Desired KL	0.01
GAE decay factor(λ)	0.95
GAE discount factor(γ)	0.99
Actor MLP size	[512, 256, 128]
Critic MLP size	[768, 512, 128]
MLP Activation	ELU

C.5 Curriculum Learning

To imitate high-dynamic motions, we introduce two curriculum mechanisms: a termination curriculum that gradually reduces tracking error tolerance, and a penalty curriculum that progressively increases the weight of regularization terms, promoting more stable and physically plausible behaviors.

- **Termination Curriculum**: The episode is terminated early when the humanoid's motion deviates from the reference beyond a termination threshold θ. During training, this threshold is gradually decreased to increase the difficulty:

$$\theta \leftarrow \text{clip}\left(\theta \cdot (1-\delta),\ \theta_{\min},\ \theta_{\max}\right), \tag{19}$$

  where the initial threshold $\theta = 1.5$, with bounds $\theta_{\min} = 0.3$, $\theta_{\max} = 2.0$, and decay rate $\delta = 2.5 \times 10^{-5}$.

- **Penalty Curriculum**: To facilitate learning in the early training stages while gradually enforcing stronger regularization, we introduce a scaling factor α that increases progressively to modulate the influence of the penalty term:

$$\alpha \leftarrow \text{clip}\left(\alpha \cdot (1+\delta),\ \alpha_{\min},\ \alpha_{\max}\right), \quad \hat{r}_{\text{penalty}} \leftarrow \alpha \cdot r_{\text{penalty}}, \tag{20}$$

  where the initial penalty scale $\alpha = 0.1$, with bounds $\alpha_{\min} = 0.0$, $\alpha_{\max} = 1.0$, and growth rate $\delta = 1.0 \times 10^{-4}$.

C.6 PD Controller Parameter

The gains of the PD controller are listed in Table 9. To improve the numerical stability and fidelity of the simulator in training, we manually set the inertia of the ankle links to a fixed value of 5×10^{-3}.

Table 9: PD controller gains.

Joint name	Stiffness (k_p)	Damping (k_d)
Left/right shoulder pitch/roll/yaw	100	2.0
Left/right shoulder yaw	50	2.0
Left/right elbow	50	2.0
Waist pitch/roll/yaw	400	5.0
Left/right hip pitch/roll/yaw	100	2.0
Left/right knee	150	4.0
Left/right ankle pitch/roll	40	2.0

D Experimental Details

D.1 Experiment Setup

- **Compute platform**: Each experiment is conducted on a machine with a 24-core Intel i7-13700 CPU running at 5.2GHz, 32 GB of RAM, and a single NVIDIA GeForce RTX 4090 GPU, with Ubuntu 20.04. Each of our models is trained for 27 hours.
- **Real robot setup**: We deploy our policies on a Unitree G1 robot. The system consists of an onboard motion control board and an external PC, connected via Ethernet. The control board collects sensor data and transmits it to the PC using the DDS protocol. The PC maintains observation history, performs policy inference, and sends target joint angles back to the control board, which then issues motor commands.

D.2 Evaluation Metrics

- Global Mean Per Body Position Error ($E_{\text{g-mpbpe}}$, mm): The average position error of body parts in global coordinates.

$$E_{\text{g-mpbpe}} = \mathbb{E}\left[\left\|\boldsymbol{p}_t - \boldsymbol{p}_t^{\text{ref}}\right\|_2\right]. \tag{21}$$

- Root-Relative Mean Per Body Position Error (E_{mpbpe}, mm): The average position error of body parts relative to the root position.

$$E_{\text{mpbpe}} = \mathbb{E}\left[\left\|(\boldsymbol{p}_t - \boldsymbol{p}_{\text{root,t}}) - (\boldsymbol{p}_t^{\text{ref}} - \boldsymbol{p}_{\text{root,t}}^{\text{ref}})\right\|_2\right]. \tag{22}$$

- Mean Per Joint Position Error (E_{mpjpe}, 10^{-3} rad): The average angular error of joint rotations.

$$E_{\text{mpjpe}} = \mathbb{E}\left[\left\|\boldsymbol{q}_t - \boldsymbol{q}_t^{\text{ref}}\right\|_2\right]. \tag{23}$$

- Mean Per Joint Velocity Error (E_{mpjve}, 10^{-3} rad/frame): The average error of joint angular velocities.

$$E_{\text{mpjve}} = \mathbb{E}\left[\left\|\Delta\boldsymbol{q}_t - \Delta\boldsymbol{q}_t^{\text{ref}}\right\|_2\right], \tag{24}$$

  where $\Delta\boldsymbol{q}_t = \boldsymbol{q}_t - \boldsymbol{q}_{t-1}$.

- Mean Per Body Velocity Error (E_{mpbve}, mm/frame): The average error of body part linear velocities.

$$E_{\text{mpbve}} = \mathbb{E}\left[\left\|\Delta\boldsymbol{p}_t - \Delta\boldsymbol{p}_t^{\text{ref}}\right\|_2\right], \tag{25}$$

  where $\Delta\boldsymbol{p}_t = \boldsymbol{p}_t - \boldsymbol{p}_{t-1}$.

- Mean Per Body Acceleration Error (E_{mpbae}, mm/frame²): The average error of body part accelerations.

$$E_{\text{mpbae}} = \mathbb{E}\left[\left\|\Delta^2\boldsymbol{p}_t - \Delta^2\boldsymbol{p}_t^{\text{ref}}\right\|_2\right], \tag{26}$$

  where $\Delta^2\boldsymbol{p}_t = \Delta\boldsymbol{p}_t - \Delta\boldsymbol{p}_{t-1}$.

D.3 Baseline Implementations

To ensure fair comparison, all baseline methods are trained separately for each motion. We consider the following baselines:

- **OmniH2O**: OmniH2O adopts a teacher-student training paradigm. We moderately increase the tracking reward weights to better match the G1 robot. In our setup, the teacher and student policies are trained for 20 and 10 hours, respectively.
- **Exbody2**: ExBody2 utilizes a decoupled keypoint-velocity tracking mechanism. The teacher and student policies are trained for 20 and 10 hours, respectively.
- **MaskedMimic**: MaskedMimic comprises three sequential training phases and we utilize only the first phase, as the remaining stages are not pertinent to our tasks. The method focuses on reproducing reference motions by directly optimizing pose-level accuracy, without explicit regularization of physical plausibility. Each policy is trained for 18 hours.

D.4 Tracking Factor Configurations

We define five sets of tracking factors: Coarse, Medium, UpperBound, LowerBound, and the initial values of Ours, as shown in Table 10. We also provide the converged tracking factors of our adaptive mechanism in Table 11.

Table 10: Tracking factors in different configurations.

Factor term	Ours(Init)	Coarse	Medium	Upperbound	Lowerbound
Joint position	0.3	0.3	0.1	0.08	0.02
Joint velocity	30.0	30.0	10.0	5.0	2.5
Body position	0.015	0.015	0.005	0.002	0.0003
Body rotation	0.1	0.1	0.03	0.4	0.02
Body velocity	1.0	1.0	0.3	0.12	0.03
Body angular velocity	15.0	15.0	5.0	3.0	1.5
Body position VRpoints	0.015	0.015	0.005	0.003	0.0003
Body position feet	0.01	0.01	0.003	0.003	0.0002
Max joint position	1.0	1.0	0.3	0.5	0.25

Table 11: Converged tracking factors of our adaptive mechanism across different motions in the ablation study of Section 5.4.

Factor term	Jabs punch	Charleston dance	Bruce Lee's pose	Roundhouse kick
Joint position	0.0310 ± 0.0002	0.0360 ± 0.0016	0.0268 ± 0.0009	0.0261 ± 0.0005
Joint velocity	2.8505 ± 0.0419	5.5965 ± 0.1797	3.6053 ± 0.0323	4.3859 ± 0.0537
Body position	0.0007 ± 0.0000	0.0023 ± 0.0001	0.0025 ± 0.0000	0.0010 ± 0.0000
Body rotation	0.0998 ± 0.0000	0.0544 ± 0.0016	0.0046 ± 0.0001	0.0829 ± 0.0176
Body velocity	0.0554 ± 0.0006	0.0941 ± 0.0013	0.0768 ± 0.0001	0.0929 ± 0.0008
Body angular velocity	1.8063 ± 0.0076	2.8267 ± 0.0841	2.1706 ± 0.0050	3.0238 ± 0.0303
Body position VRpoints	0.0008 ± 0.0000	0.0031 ± 0.0002	0.0024 ± 0.0000	0.0015 ± 0.0000
Body position feet	0.0006 ± 0.0000	0.0031 ± 0.0001	0.0028 ± 0.0000	0.0011 ± 0.0000
Max joint position	0.3963 ± 0.0003	0.4339 ± 0.0124	0.3299 ± 0.0064	0.3352 ± 0.0010

E Additional Experimental Results

E.1 Analysis of Contact Mask Estimation and Motion Correction Method

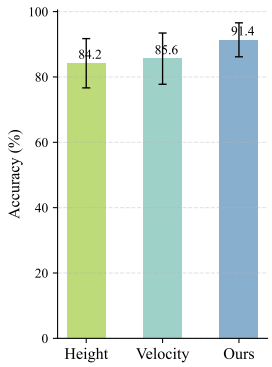

Figure 9: Accuracy of contact mask estimation across different methods.

Fig. 9 illustrates the accuracy of the proposed contact mask estimation method, evaluated on a manually labeled motion dataset with 10 samples. The proposed approach demonstrates an impressive accuracy of 91.4%.

Fig. 10 presents a visual comparison of the efficacy of the proposed motion correction technique in mitigating floating artifacts. Prior to motion correction, the overall height of the SMPL model is noticeably elevated relative to the ground level. In contrast, after applying the correction, the model's motion aligns more accurately with the ground plane, effectively reducing the observed floating artifacts.

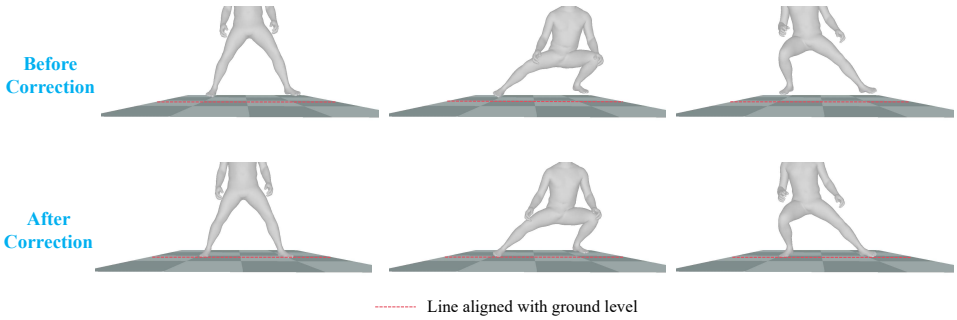

Figure 10: Visualization of motion correction effectiveness in mitigating floating artifacts.

E.2 Ablation Study of Adaptive Motion Tracking Mechanism

Table 12 presents the ablation study results evaluating the impact of different tracking factors on four motion tasks: Jabs Punch, Charleston Dance, Roundhouse Kick, and Bruce Lee's Pose.

Table 12: Ablation results of adaptive motion tracking mechanism in Section 5.4.

Method	$E_{\text{g-mpbpe}} \downarrow$	$E_{\text{mpbpe}} \downarrow$	$E_{\text{mpjpe}} \downarrow$	$E_{\text{mpbve}} \downarrow$	$E_{\text{mpbae}} \downarrow$	$E_{\text{mpjve}} \downarrow$
Jabs punch						
Ours	**44.38**$_{\pm 7.118}$	**28.00**$_{\pm 3.533}$	**783.36**$_{\pm 11.73}$	5.52$_{\pm 0.156}$	**6.23**$_{\pm 0.063}$	**88.01**$_{\pm 2.465}$
Coarse	63.95$_{\pm 6.680}$	36.76$_{\pm 2.743}$	921.50$_{\pm 16.70}$	6.16$_{\pm 0.011}$	6.46$_{\pm 0.042}$	91.46$_{\pm 0.465}$
Medium	**51.07**$_{\pm 2.635}$	**30.93**$_{\pm 2.635}$	**790.54**$_{\pm 22.82}$	5.68$_{\pm 0.140}$	6.31$_{\pm 0.057}$	**90.19**$_{\pm 1.821}$
Upperbound	**45.74**$_{\pm 1.702}$	**28.72**$_{\pm 1.702}$	**793.52**$_{\pm 8.888}$	**5.43**$_{\pm 0.066}$	**6.29**$_{\pm 0.085}$	**88.68**$_{\pm 0.727}$
Lowerbound	**48.66**$_{\pm 0.488}$	**28.97**$_{\pm 0.487}$	**781.73**$_{\pm 16.72}$	5.61$_{\pm 0.079}$	6.31$_{\pm 0.026}$	**88.44**$_{\pm 1.397}$
Charleston dance						
Ours	94.81$_{\pm 14.18}$	43.09$_{\pm 5.748}$	**886.91**$_{\pm 74.76}$	**6.83**$_{\pm 0.346}$	**7.26**$_{\pm 0.034}$	**162.70**$_{\pm 7.133}$
Coarse	119.24$_{\pm 4.501}$	55.80$_{\pm 1.324}$	1288.02$_{\pm 3.807}$	7.54$_{\pm 0.180}$	7.28$_{\pm 0.021}$	178.61$_{\pm 3.304}$
Medium	**83.63**$_{\pm 3.159}$	**41.02**$_{\pm 1.743}$	**933.33**$_{\pm 38.23}$	**6.89**$_{\pm 0.185}$	**7.22**$_{\pm 0.011}$	**164.92**$_{\pm 4.380}$
Upperbound	86.90$_{\pm 8.651}$	**41.92**$_{\pm 2.632}$	**917.64**$_{\pm 14.85}$	**7.02**$_{\pm 0.103}$	**7.22**$_{\pm 0.041}$	**167.64**$_{\pm 1.089}$
Lowerbound	358.82$_{\pm 10.35}$	145.42$_{\pm 1.109}$	1199.21$_{\pm 12.78}$	8.99$_{\pm 0.050}$	8.48$_{\pm 0.033}$	**167.25**$_{\pm 0.783}$
Roundhouse kick						
Ours	**52.53**$_{\pm 2.106}$	**28.39**$_{\pm 1.400}$	**708.55**$_{\pm 16.04}$	**6.85**$_{\pm 0.196}$	**7.13**$_{\pm 0.046}$	**106.22**$_{\pm 0.715}$
Coarse	76.81$_{\pm 2.863}$	38.98$_{\pm 2.230}$	1008.32$_{\pm 29.74}$	7.49$_{\pm 0.234}$	7.57$_{\pm 0.044}$	108.40$_{\pm 0.010}$
Medium	63.12$_{\pm 5.178}$	33.74$_{\pm 2.336}$	806.84$_{\pm 66.23}$	**7.03**$_{\pm 0.125}$	7.32$_{\pm 0.046}$	**104.77**$_{\pm 1.319}$
Upperbound	54.95$_{\pm 2.164}$	31.31$_{\pm 0.344}$	766.32$_{\pm 12.92}$	**6.93**$_{\pm 0.013}$	7.19$_{\pm 0.012}$	**105.64**$_{\pm 1.911}$
Lowerbound	70.10$_{\pm 2.674}$	36.29$_{\pm 1.475}$	**715.01**$_{\pm 34.01}$	7.08$_{\pm 0.102}$	7.32$_{\pm 0.067}$	**102.50**$_{\pm 4.650}$
Bruce Lee's pose						
Ours	196.22$_{\pm 17.03}$	69.12$_{\pm 2.392}$	**972.04**$_{\pm 49.27}$	7.57$_{\pm 0.214}$	8.54$_{\pm 0.198}$	94.36$_{\pm 3.750}$
Coarse	239.06$_{\pm 51.74}$	80.78$_{\pm 15.81}$	1678.34$_{\pm 394.3}$	8.42$_{\pm 0.525}$	8.93$_{\pm 0.422}$	112.30$_{\pm 10.87}$
Medium	470.24$_{\pm 249.2}$	206.92$_{\pm 116.1}$	4490.80$_{\pm 105.1}$	9.58$_{\pm 0.085}$	9.61$_{\pm 0.080}$	99.65$_{\pm 2.441}$
Upperbound	250.64$_{\pm 178.6}$	93.70$_{\pm 65.09}$	1358.02$_{\pm 561.6}$	8.31$_{\pm 2.160}$	8.94$_{\pm 1.384}$	106.30$_{\pm 23.06}$
Lowerbound	**158.12**$_{\pm 2.934}$	**60.54**$_{\pm 1.554}$	**955.10**$_{\pm 37.04}$	**7.05**$_{\pm 0.040}$	**7.94**$_{\pm 0.051}$	**81.60**$_{\pm 1.277}$

E.3 Ablation Study of Contact Mask

To evaluate the effectiveness of the contact mask, we additionally conducted an ablation study on three representative motions characterized by distinct foot contact patterns: Charleston Dance, Jump Kick, and Roundhouse Kick. We additionally introduce the mean foot contact mask error as a metric:

$$E_{\text{contact-mask}} = \mathbb{E}\left[\|c_t - \hat{c}_t\|_1\right]. \tag{27}$$

The results, shown in Table 13, demonstrate that our method significantly reduces foot contact errors $E_{\text{contact-mask}}$ compared to the baseline without the contact mask. In addition, it also leads to noticeable improvements in other tracking metrics, validating the effectiveness of the proposed contact-aware design.

E.4 Additional Real-World Results

Fig. 11 presents additional results of deploying our policy in the real world, covering more highly-dynamic motions. These results further validate the effectiveness of our method in tracking high-dynamic motions, enabling the humanoid to learn more expressive skills.

Table 13: Ablation results of contact mask.

Method	$E_{\text{contact-mask}} \downarrow$	$E_{\text{mpbpe}} \downarrow$	$E_{\text{mpjpe}} \downarrow$	$E_{\text{mpbve}} \downarrow$	$E_{\text{mpbae}} \downarrow$
Charleston dance					
Ours	$217.82_{\pm 47.97}$	$43.09_{\pm 5.748}$	$886.91_{\pm 74.76}$	$6.83_{\pm 0.346}$	$7.26_{\pm 0.034}$
Ours w/o contact mask	$633.91_{\pm 49.74}$	$76.13_{\pm 53.01}$	$980.40_{\pm 222.0}$	$7.72_{\pm 1.439}$	$7.64_{\pm 0.594}$
Jump kick					
Ours	$294.22_{\pm 6.037}$	$42.58_{\pm 8.126}$	$840.33_{\pm 97.76}$	$9.48_{\pm 0.717}$	$10.21_{\pm 10.21}$
Ours w/o contact mask	$386.75_{\pm 6.036}$	$170.28_{\pm 97.29}$	$1259.21_{\pm 423.9}$	$16.92_{\pm 0.012}$	$16.57_{\pm 5.810}$
Roundhouse kick					
Ours	$243.16_{\pm 1.778}$	$28.39_{\pm 1.400}$	$708.55_{\pm 16.04}$	$6.85_{\pm 0.196}$	$7.33_{\pm 0.046}$
Ours w/o contact mask	$250.10_{\pm 6.123}$	$36.76_{\pm 2.743}$	$921.52_{\pm 16.70}$	$6.16_{\pm 0.012}$	$6.46_{\pm 0.042}$

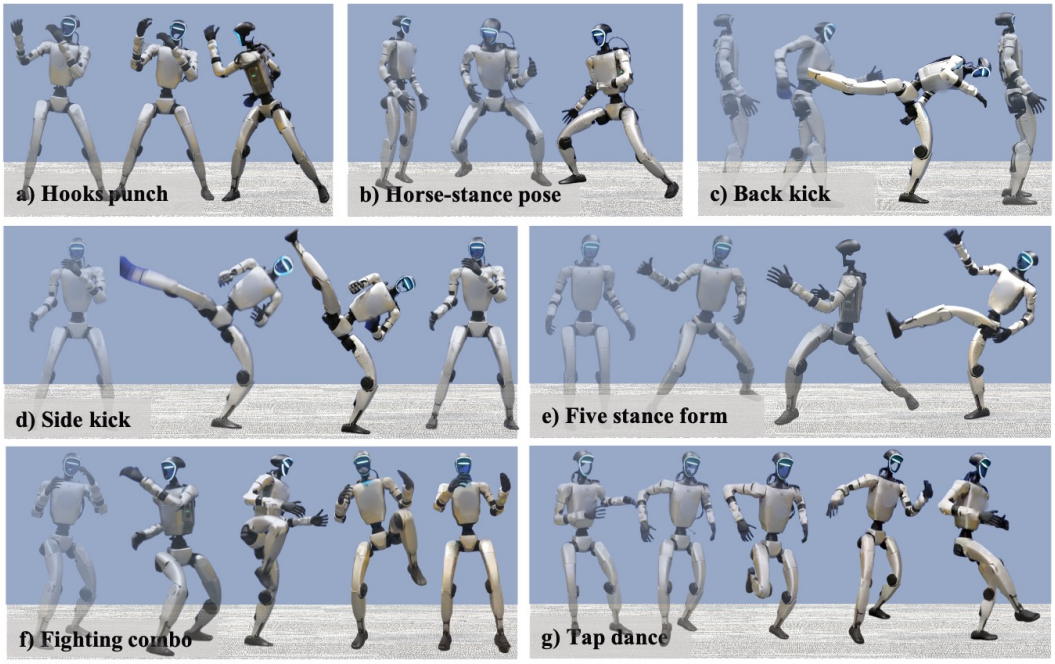

Figure 11: Our robot masters more dynamic skills in the real world. Time flows left to right.

F Broader Impact

Our work advances humanoid robotics by enabling the imitation of complex, highly-dynamic human motions such as martial arts and dancing. This has broad potential in fields like physical assistance, rehabilitation, education, and entertainment, where expressive and agile robot behavior can support training, therapy, and interactive experiences. However, such capabilities also raise important ethical and societal concerns. High-agility robots interacting closely with humans introduce safety risks, and their potential to replace skilled human roles in performance, instruction, or service contexts may lead to labor displacement. Moreover, the misuse of advanced motion imitation—for example, in surveillance or military applications—poses security concerns. These risks call for clear regulation, strong safety mechanisms, and human oversight. Additionally, the environmental cost of training models and operating physical robots highlights the need for energy-efficient and sustainable development. We believe this work should be viewed as a step toward responsible, human-aligned robotics, and we encourage continued dialogue on its societal impact.