A Unified and General Humanoid Whole-Body Controller for Versatile Locomotion

Yufei Xue[†1,2] Wentao Dong[†1,2] Minghuan Liu[ˆ1] Weinan Zhang[1] Jiangmiao Pang[2]
[1]Shanghai Jiao Tong University [2]Shanghai AI Lab
[†]Equal Contributions [ˆ]Project Lead
https://hugwbc.github.io

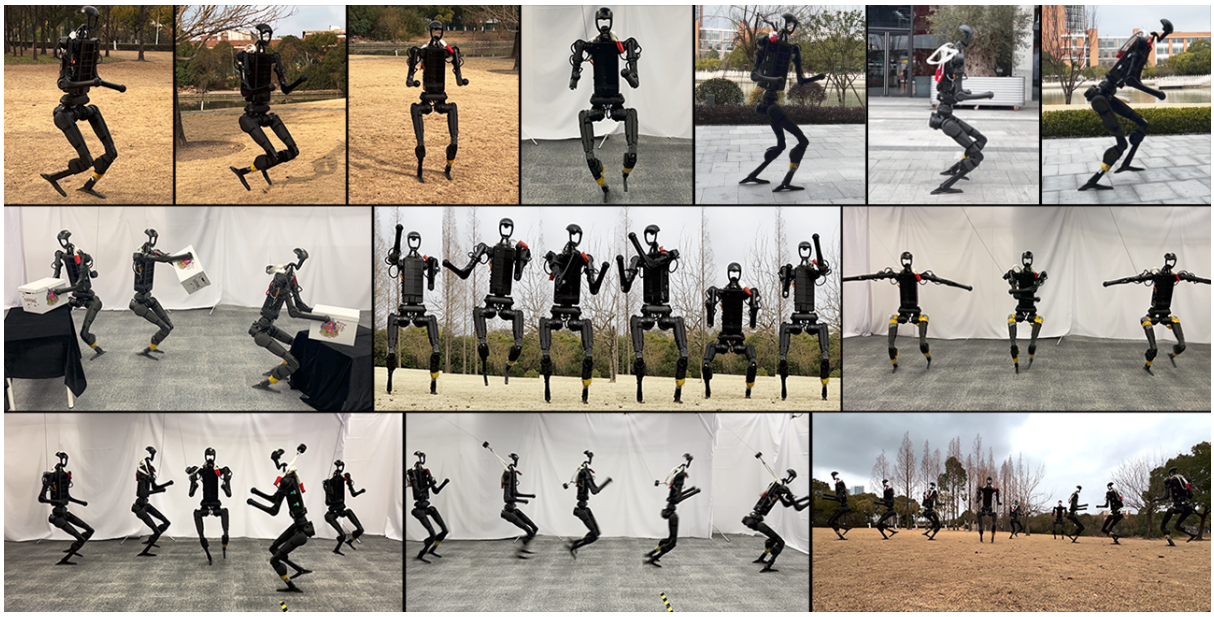


Fig. 1: **Humanoid capabilities supported by HUGWBC. First row:** HUGWBC allows four standard gaits - walking, jumping, standing, and hopping - with multiple customizable parameters to adjust the foot and pose behaviors, using one policy for 3 of the 4 gaits. **Second row:** HUGWBC supports real-time interventions from external upper-body controllers, enabling loco-manipulation while maintaining precise control under any locomotive behavior. **Third row**: Various command combinations enable the robot to perform highly dynamic movements.

***Abstract*—Locomotion is a fundamental skill for humanoid robots. However, most existing works make locomotion a single, tedious, unextendable, and unconstrained movement. This limits the kinematic capabilities of humanoid robots. In contrast, humans possess versatile athletic abilities—running, jumping, hopping, and finely adjusting gait parameters such as frequency and foot height. In this paper, we investigate solutions to bring such versatility into humanoid locomotion and thereby propose HUGWBC: a unified and general humanoid whole-body controller for versatile locomotion. By designing a general command space in the aspect of tasks and behaviors, along with advanced techniques like symmetrical loss and intervention training for learning a whole-body humanoid controlling policy in simulation, HUGWBC enables real-world humanoid robots to produce various natural gaits, including walking, jumping, standing, and hopping, with customizable parameters such as frequency, foot swing height, further combined with different body height, waist rotation, and body pitch. Beyond locomotion, HUGWBC also supports real-time interventions from external upper-body controllers like teleoperation, enabling loco-manipulation with precision under any locomotive behavior. Extensive experiments validate the high tracking accuracy and robustness of HUGWBC with/without upper-body intervention for all commands, and we further provide an in-depth analysis of how the various commands affect humanoid movement and offer insights into the relationships between these commands. To our knowledge, HUGWBC is the first humanoid whole-body controller that supports such versatile locomotion behaviors with high robustness and flexibility.**

I. INTRODUCTION

Recent progress in humanoid robots has shown impressive results in achieving complex tasks, and the huge potential to become a general robot platform [4, 55, 3, 41]. It is a fundamental skill to support various humanoid motions, enabling them to navigate environments and perform tasks with agility and adaptability. However, most current humanoid locomotion systems, although showing impressive results in motion-based control [20, 23, 12, 3] and mobile manipulation [29], pay limited attention to producing versatile and controllable gait styles, leading to single, tedious, unextendable, and unconstrained movements. Consider humans, we have versatile athletic abilities, such as running, jumping, and even hopping. Even when only walking, we can fine-tune our frequencies, strides, and foot heights. Bringing such versatility into humanoid locomotion is challenging, but it is the key to exploring the edge of humanoid robots' abilities. To resolve the challenge and build a unified and general humanoid whole-body controller, in this work, we propose HUGWBC,

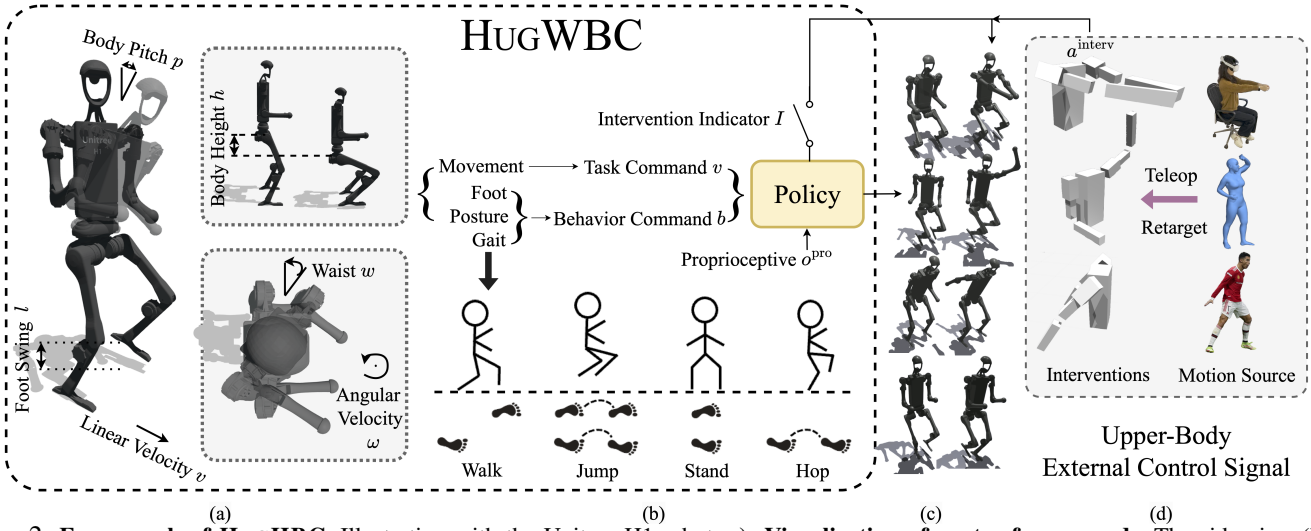


Fig. 2: **Framework of HUGHBC.** Illustration with the Unitree H1 robot. **a): Visualization of parts of commands**. The side view (left) highlights the linear velocity, foot swing height, and body pitch commands. The top-right view shows the angular velocity and waist yaw commands, and the bottom-right view shows the body height command. **b): Policy inputs/outputs.** The policy is provided with commands, proprioceptive observations, the intervention indicator, and outputs all joints of the robots. **c): Illustrations of four gaits on the robot without/with external intervention.** By default, the policy controls both the upper-body and the lower-body joints. **d): External control support.** Feasible external control signals can be seamlessly integrated into the robot's behavior without hurting locomotion performance.

namely, **H**umanoid's **U**nified and **G**eneral **W**hole-**B**ody **C**ontrol. HUGWBC is designed for generating versatile locomotion with dynamic, customizable control, enabling the robot to perform gaits such as walking, standing, jumping, and hopping. Furthermore, HUGWBC provides the flexibility to adjust foot behavior parameters foot swing height and gait frequency, and allows combining posture parameters such as body height, waist rotation, and body pitch. To achieve this, HUGWBC includes a general command space designed for humanoid locomotion, along with advanced training techniques to learn versatile gaits within *one single policy* (except the hopping gait) using reinforcement learning in simulation, which can be directly transferred onto real robots.

Positioned as a basic controller for humanoid robots to perform a wider range of tasks in diverse real-world scenarios, HUGWBC introduces intervention training and supports real-time external control signals of the upper body, like teleoperation, allowing for highly robust loco-manipulation, while maintaining precise locomotion control. An overview of the framework is illustrated in Fig. 2.

In experiments, we show HUGWBC preserves high tracking accuracy on eight different commands under four different gaits; we also ablate the improvement in stability and robustness of the upper body intervention training. We further provide a detailed analysis of how commands combination works, shedding light on the intricate relationships between these commands and how they can be leveraged to optimize movement performance. Through this work, we aim to significantly broaden the scope of humanoid locomotion capabilities, pushing the boundaries of what is possible with current robotic systems.

The key contributions are summarized as follows:

- An extended general command space with advanced training techniques designed for versatile humanoid locomotion.
- Accurate tracking for eight different commands under four different gaits, using one policy for 3 of the 4 gaits.
- A basic humanoid controller that supports external upper-body intervention and enables a wider range of loco-manipulation tasks.

II. RELATED WORK

A. Model-Based Humanoid Controller

Controlling humanoid robots has become one of the most fascinating problems since decades ago, many researchers and engineers have built complicated systems and tried to solve them with model-based methods in a perspective of optimal control (OC) [47, 49, 1, 11, 51, 34, 40]. These works typically employ trajectory optimization with dynamic models of varying levels of complexity, such as the linear inverted pendulum model [24], centroidal dynamics model [37, 50], or full-body dynamics model [44, 25, 53], to perform online optimization, or generate periodic motion control through the hybrid zero dynamics model [7, 48, 21]. However, most of them can only generate motion based on predefined contact sequences. Even some have successfully incorporated online optimization to generate real-time motion sequences and contact schedules based on instant environmental feedback and user commands and run on humanoid robots in the real world [16], the nonlinear dynamics and multi-contact optimization of humanoid systems demand significant computational resources, making it challenging to meet real-time performance requirements. A promising solution is to decouple the whole-body multi-contact optimization control problem into two subproblems: contact planning and motion optimization [35, 34, 8]. The goal of the contact planning stage is to generate the desired multi-contact sequence for rich whole-body motion and gait control, including the order and position of both hand and foot contacts [39, 25]. The motion optimization phase optimizes the robot motion trajectory based on the contact sequence. Although decoupling

simplifies the problem, model-based approaches still rely on several assumptions, including perfect state estimation and flawless execution of planned movements. However, most assumptions no longer hold in the real world, and the dynamics model is not perfect to describe real robot systems, which results in poor robustness when applied in real environments.

B. Learning-Based Humanoid Controller

Recent advancements in learning-based controllers have demonstrated the locomotion capability to go through rough terrains [41, 17], achieving smooth and efficient motions [2]. However, controllers relying on proprioceptive sensing must predict surrounding terrain through collision detection and swiftly adapt their motion, presenting significant challenges for inherently unstable humanoid robotic systems. Some recent approaches incorporated depth maps or elevation maps into the policy observations, enabling impressive parkour tasks [56, 28]. Some researchers have utilized chain-contact reward functions to learn jumping gaits for humanoid robots [55].

Additionally, with the support of teleoperation systems for humanoid robots [4, 13] and large-scale humanoid motion datasets [30], researchers have made progress in motion tracking and learned rich whole-body motion representations for humanoid robots. Some studies focused on upper body tracking combined with maintaining balance in the lower body [3]. Some others explored controlling whole-body joints in one policy, differing primarily in their control interfaces/command spaces: He et al. tracked whole-body motion capture keypoints; Fu et al., Ji et al. track retargeted joint position; He et al. tracked VR-based head and hands keypoints; He et al. tracked all of these and propose a universal interface approach. Different from them, Lu et al. decoupled the control interface, and combined an IK-based upper-body controller with a learning-based lower-body controller. The lower-body command space includes the task command and the pose command as used in this work, and they introduced the prior knowledge of upper-body movements to the lower-body policy to help its robustness. However, we show that without such a component, we can still construct a robust loco-manipulation controller.

We made several choices in this work: 1) we extend the command space beyond all of these previous works, by introducing additional behavior commands that control the foot and the gait; 2) we employ a learning-based controller to control whole-body joints (instead of only lower-body as in Lu et al.) while supporting external controller (with IK or joint sequences) to take over upper-body joints, since upper-body and lower-body serves as different requirements. Accurate upper-body control is useful for tasks that require precision, while the robot should be robust to arbitrary upper-body intervention under any behavior.

III. Background

A. Humanoid Whole-Body Control

To support various high-level functionalities and allow the humanoid robot to perform complicated tasks, a basic whole-body controller is essential. Formally, given a set of continuous commands $\mathcal{C}$ and observations $\mathcal{O}$, our objective is to develop a control function that maps these inputs to appropriate control signals. Model-based approaches represent one solution paradigm, typically decomposing the control function into planning and tracking modules [15, 35]. The planning module generates optimal trajectories and contact sequences based on $\mathcal{O}$ and $\mathcal{C}$, while the tracking module translates these into control laws, specifying joint positions, velocities, and torques. However, these methods face computational challenges due to the complex dynamics of humanoid robots and the discrete nature of whole-body contact points. Learning-based methods offer an alternative approach by directly learning a policy function $a = \pi(o, c)$ that maps observations $\mathcal{O}$ and commands $\mathcal{C}$ to joint-level actions [45]. These actions typically represent offsets to target joint positions across three categories: upper-body, lower-body (legs), and hands. The final target position combines the nominal position with these learned offsets, which is then tracked using a proportional derivative (PD) controller with fixed gains.

B. Command Tracking as Reinforcement Learning

To achieve a generalized and powerful whole-body control behavior for humanoid robots, we learn a policy function by constructing a command-tracking problem. In detail, we want the learned policy π to control the robots to match the provided commands c. To this end, we use reinforcement learning (RL), where we define the reward functions r typically by distances d or similarities s of the command c and the observed robot state s_c corresponding to that command:

$$r(o, a, c) = -d(c, s_c) \text{ or } r(o, a, c) = s(c, s_c) \ . \tag{1}$$

Under the formulation of RL, the policy is trained to maximize the rewards, corresponding to matching these commands.

C. Simulation Training and Real-World Transfer

Many recent works, especially those of legged robots, take advantage of RL training a robust robot-control policy with a large set of parallel environments in simulation and directly deploying into the real world [5, 26, 20, 22]. Due to the partial observability of the real robot, whose onboard sensors can only provide limited and noisy observations, it is difficult to learn a deployable policy from them directly. Therefore, researchers have developed a set of sim-to-real techniques to resolve the challenge. Among them, one of the most commonly used techniques is asymmetric training [38, 36], which is proposed as a one-stage solution for sim-to-real training.

In this paper, we adopt an asymmetric actor-critic (AAC) framework proposed for quadruped locomotion [6]. In this framework, the critic network has access to all privileged information, while the actor network only receives data available from onboard sensors, with a separate encoder to estimate the key privileged information (*e.g.*, the linear velocity, robot's body height, and robot's feet swing height). The training paradigm incorporates the RL objective (including a value loss $\mathcal{L}^{\text{value}}$ and a policy loss $\mathcal{L}^{\text{policy}}$) with an estimation loss

[36, 27, 6] $\mathcal{L}^{\text{est}}$ to train the encoder:

$$\mathcal{L}^{\text{AAC}} = \mathcal{L}^{\text{value}} + \lambda^{\text{policy}}\mathcal{L}^{\text{policy}} + \lambda^{\text{est}}\mathcal{L}^{\text{est}} \tag{2}$$

In this work, we take AAC as our default training framework, but the proposed techniques are not limited to it.

IV. HUGWBC

A. A General Command-Space for Humanoid Locomotion

We define the command space of the humanoid whole-body controller $\mathcal{C} = \mathcal{K} \times \mathcal{B}$ by two sets of commands, the task commands $\mathcal{K}$ and the behavior commands $\mathcal{B}$. The task commands determine a goal for the robot to reach, typically for movement, while the behavior commands construct the specific behavior pattern of the humanoid robots. In this work, we specify the task command as the target velocity $v_t \in \mathbb{R}^3$ (including the longitudinal and horizontal linear velocities $v_{t,x}$, $v_{t,y}$ and the angular velocity ω_t) at each time step t. As for the behavior command, we define the behavior command b_t as a vector:

$$\Big[\underbrace{f_t, l_t}_{\text{foot}}, \underbrace{h_t, p_t, w_t}_{\text{posture}}, \underbrace{\psi_t, \phi_{t,1}, \phi_{t,2}, \phi_{t,\text{stance}}}_{\text{gait}}\Big] \ , \tag{3}$$

where $f_t \in \mathbb{R}$ is the gait frequency and $l_t \in \mathbb{R}$ is the maximum foot swing height, both of which can be explained as foot behaviors. Besides, $h_t \in \mathbb{R}$ represents the body height, $p_t \in \mathbb{R}$ is the body pitch angle, and $w_t \in \mathbb{R}$ is the waist yaw rotation. These commands can be regarded as controlling the posture behavior.

Beyond the commands above, we further introduce distinct gaits, such as walking, standing, jumping, and hopping. To do so, we refer to legged gait control [45, 33] and define $\phi_i \in [0, 1)$, $i = 1, 2$ as two time-varying periodic phase variables to represent the humanoid gaits, on behalf of two legs (feet). These two phase variables can either be set as constants, or be computed by the phase offset ψ and the gait frequency $f_t \in \mathbb{R}$ at each time:

$$\begin{aligned} \phi_{t+1,1} &= (\phi_{t,1} + f_t \times \mathrm{d}t) \ , \\ \phi_{t+1,2} &= (\phi_{t+1,1} + \psi) \ , \end{aligned} \tag{4}$$

where $\mathrm{d}t$ is a discrete time step. When following the computation of Eq. 4, each ϕ_i loops in a range of $[0, 1)$, resulting in repeated phase cycles. $\phi_{\text{stance}} \in [0, 1]$ is the duty cycle, which divides the gait cycle into two stages: stance (i.e., foot in contact with the ground) when $\phi_i < \phi_{\text{stance}}$, and swinging (i.e., foot in the air) otherwise. f is the stepping frequency, determining the wall time of each gait cycle.

Humanoid gait control. We consider four distinct standard gaits in this project, *i.e.*, *walking*, *jumping*, *standing*, *hopping*[1] By constructing the behavior commands above, we can adjust the phase offset ψ, the duty cycle ϕ_{stance}, and the phase variable ϕ_i for each leg to control the humanoid robots in versatile gaits. In this work, we only consider standard gaits, so we set the phase offset $\psi = 0.5$ for *walking* gaits [45], since the phase difference between the left and right foot is half a cycle; Regarding *jumping* gaits, the phase of the left and right foot is the same, thus, we set ψ to 0. As for the *standing* and the *hopping* gaits, a certain foot of the robot is always in two states of contact or non-contact with the ground, which motivates a constant ϕ_i (resulting in constant contact probability of either 0 or 1, and ψ is not working). In particular, for the standing gait, we set $\phi_i = 0.25$ for both feet; and for the hopping gait, $\phi_i = 0.75$ for the flying leg, and ϕ_i of the other leg steps with frequency f. The ϕ_{stance} determines the time ratio of stance and swinging during a gait cycle, and a smaller ϕ_{stance} will promote longer leg flight time. To represent a smooth switch between stance and swinging, we introduce the expected contact probability function $C(\phi_{t,i})$ for leg $i \in \{1, 2\}$ at each time step t as:

$$\begin{aligned} C(\phi_{t,i}) = \ & \Phi(\bar{\phi}_{t,i}/\sigma)[1 - \Phi((\bar{\phi}_{t,i} - 0.5)/\sigma)] \\ & + \Phi((\bar{\phi}_{t,i} - 1)/\sigma)[1 - \Phi((\bar{\phi}_{t,i} - 1.5)/\sigma)] \ , \end{aligned} \tag{5}$$

$$\bar{\phi}_i = \begin{cases} 0.5 \times \dfrac{\phi_i}{\phi_{\text{stance}}}, & \phi_i < \phi_{\text{stance}} \\ 0.5 + 0.5 \times \dfrac{\phi_i - \phi_{\text{stance}}}{1 - \phi_{\text{stance}}}, & \phi_i \geq \phi_{\text{stance}} \end{cases} , \tag{6}$$

where $\bar{\phi}_i \in [0, 1]$ is a homogenized phase variable that maps the ϕ_i of the stance and swinging phases to intervals $[0, 0.5]$ and $[0.5, 1]$ according to ϕ_{stance}, as computed in Eq. (6). $\Phi(\cdot)$ is the cumulative distribution function (CDF) of the standard normal distribution $N(0, 1)$. The standard deviation σ allows for the relaxation of switching points ($\bar{\phi}_i = 0, 0.5$) to switching interval ($\bar{\phi}_i \in [-3\sigma, 3\sigma], [0.5 - 3\sigma, 0.5 + 3\sigma]$) (see Fig. 3 for a detailed explanation). Intuitively, $C(\phi_{t,i})$ is the probability of leg i coming into contact with the ground. As one may notice, when $\bar{\phi}_{t,i} \in [0, 0.5]$, the first term of $C(\phi_{t,i})$ is dominant; otherwise, the second term becomes dominant. In this work, we set a constant $\phi_{\text{stance}} = 0.5$ for all supported gaits in all time steps, which means half-time stance/swinging during one cycle.

We highlight that HUGWBC trained *one single policy* for the standing, walking, and jumping gaits, and an independent policy for the hopping gait.

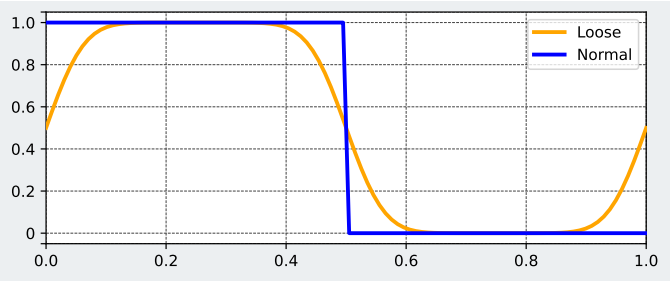


Fig. 3: The expected contact probability function $C(\phi_{t,i})$ in the loose and normal formulation. The larger $C(\phi_{t,i})$, the higher the expectation of contact with the ground. The CDF of the normal distribution is introduced into the normal contact probability function to relax the constraint of the foot contact at the switching boundary, resulting in a smooth transition between the swing and the stance phase.

[1]We note that *running* can be further derived from the *walking* gaits via high-velocity and small duty cycle commands, which promotes the prolonged flight of both feet.

B. Detailed Observation

In our asymmetric actor-critic framework, the observation for the critic network o_t^V obtains all information related to the environment, including proprioceptive observations o_t^{pro}, privileged observations o_t^{pri}, terrain observations o_t^{ter}, commands c_t, and an indicator signal $I(t)$. Regarding the actor network, its available observation o_t^{π} only contains history of proprioceptive observations within last k steps $o_t^{\text{his}} = (o_{t-k+1}^{\text{pro}}, \dots, o_t^{\text{pro}})$, commands c_t, and the indicator signal $I(t)$. The proprioceptive $o_t^{\text{pro}} \in \mathbb{R}^{63}$ consists of angular velocity and gravity projection in the robot's base frame, joint position, joint velocity, and previous policy output a_{t-1}. The privileged observations $o_t^{\text{pri}} \in \mathbb{R}^{24}$ contain the linear velocity, the base height error, foot clearance, friction coefficient of the ground, feet contact forces, and collision detection of the link (trunk, hip, thigh, shank, shoulder, and arm). The terrain observations $o_t^{\text{ter}} \in \mathbb{R}^{221}$ are samplings of terrain elevation points around the robot.

Commands. The commands $c_t = [v_t, \tilde{b}_t]$ includes the task command (*i.e.*, target velocity v_t in this work) and the extended behavior command $\tilde{b}_t \in \mathbb{R}^9$, where we extend the behavior command b_t defined above through replacing the phase variables $\phi_i,\ i = 1, 2$ with two additional clock functions $[Cl_L(t), Cl_R(t)] = [\sin\left(2\pi\bar{\phi}_{t,1}\right), \sin\left(2\pi\bar{\phi}_{t,2}\right)]$ representing the contact of both feet, where $\bar{\phi}_{t,i},\ i = 1, 2$ are the homogenized phase variables defined in Eq. (6). Note that the sine function $\sin\left(2\pi\bar{\phi}_{t,i}\right), i = 1, 2$ is a gait cycle contact indicator function, designed for a smoother transition between swinging and stance phases. An illustrative explanation of the phase variables and the clock function is shown in Fig. 4.

External upper-body control. We want to build a general humanoid whole-body controller that also supports external upper-body control (e.g., teleoperation). Thereafter, we introduce a binary indicator function $I(t)$ to identify whether an external upper-body controller is involved. If there is no external upper-body control signal involved, the upper-body joints are controlled by our developed whole-body controller by default, which swings the arms naturally.

C. Reward Design for Policy Learning

Our humanoid whole-body controller is obtained through an asymmetric actor-critic training paradigm via reinforcement learning (RL). To learn a policy with general and diverse behaviors, we design a set of reward functions, which mainly consist of three parts: task rewards, behavior rewards, and regularization rewards. The details of the rewards are concluded in Tab. I.

The *task* rewards are meant to track any task command k. In this work, it is the target velocity v, including the linear and angular velocities. The *regularization* rewards take into account the performance of physical hardware and impose constraints on the smoothness and safety of the locomotion. These are commonly used in previous works [42]. In this work, since we want to build a general whole-body controller to support versatile locomotion behaviors for humanoid robots, we introduce a set of *behavior* rewards to encourage the robots to track any behavioral commands b, shown below.

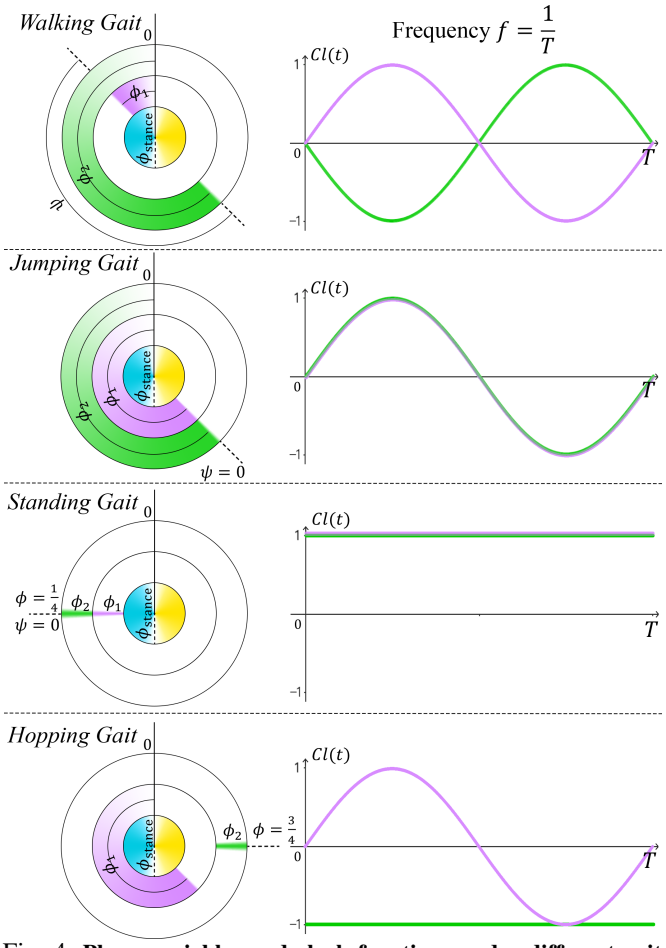


Fig. 4: **Phase variables and clock functions under different gaits. Left**: The purple ring represents the phase variable ϕ_1 for the left foot, while the green ring represents the phase variable ϕ_2 for the right foot. ψ is the phase offset from ϕ_1 to ϕ_2. The dividing phase between stance (marked in blue) and swing (marked in yellow) is the duty cycle $\phi_{\text{stance}}(0.5)$. **Right**: The purple line depicts the clock function $Cl_L(t)$ for the left foot over a cycle, while the green line represents the clock function $Cl_R(t)$ for the right foot over a cycle.

For most behavior commands, including body height h, body pitch p, and waist rotation w, we simply formulated the rewards with mean squared error (MSE):

$$r_t^{\text{cmd}} = \left\| e_t^{\text{target}} - e_t^{\text{cmd}} \right\|^2 . \tag{7}$$

Beyond these simple tracking rewards, we further introduce periodic contact-swing rewards r_t^{contact} [45, 33] and the foot trajectory rewards r_t^{traj} to help generate complicated gaits.

The periodic contact-swing reward r_t^{contact} is designed for precise adjustments between swinging and stance in different gaits, according to ϕ_i. Since humanoid gaits can be expressed as different combinations of contact sequences, like foot contact forces and velocities, we define the periodic contact-swing rewards r_t^{contact} over them to generate desired contact patterns. Based on $C(\phi_{t,i})$ defined as Eq. (6), we then construct the periodic contact-swing rewards r_t^{contact} to encourage humanoid robots to learn specific contact modes and generate various

TABLE I: **Reward definitions used in HUGWBC.**

Term	Definition	Weight
Task Reward		
Linear Velocity Tracking	$\exp\left(-\left\|v_{xy}^{\text{target}} - v_{xy}\right\|^2 / 0.2\right)$	2
Angular Velocity Tracking	$\exp\left(-\left\|\omega_z^{\text{target}} - \omega_z\right\|^2 / 0.2\right)$	2
Behavior Reward		
Body Height Tracking	$\left\|h^{\text{target}} - h\right\|^2$	-40
Body Pitch Tracking	$\left\|p^{\text{target}} - p\right\|^2$	-10
Waist Yaw Tracking	$\left\|w^{\text{target}} - w\right\|^2$	-2
Foot Swing Tracking	$\sum_i [1 - C(\phi_i)] \left\|l^{\text{target},i} - l^{\text{foot},i}\right\|^2$	-30
Contact-Swing Tracking	$-\sum_i [1 - C(\phi_i)] \left[1 - \exp(\left\|f^{\text{foot},i}\right\|^2 / 50)\right]$ $-C(\phi_i)\left[1 - \exp\left(\|v_{xy}^{\text{foot},i}\|^2 / 5\right)\right]$	-2
Regularization Reward		
R-P Angular Velocity	$\left\|\omega_{xy}\right\|^2$	-0.5
Vertical Body Movement	$\left\|v_z\right\|^2$	-0.1
Feet Slip	$1 - \sum_i \exp\left(-\left\|v_{xy}^{\text{foot,i}}\right\|^2\right)$	-0.2
Action Rate	$\left\|a_t - a_{t-1}\right\|^2$	-0.01
Action Smoothness	$\left\|a_{t-2} - 2a_{t-1} + a_t\right\|^2$	-0.01
Joint Torque	$\left\|\tau\right\|^2$	-5e-6
Joint Acceleration	$\left\|\ddot{q}\right\|^2$	-2.5e-7
Upper Joint Deviation	$\left\|q_{\text{upper}} - q_{\text{upper}}^{\text{nominal}}\right\|^2$	-0.5
Hip Joint Deviation	$\left\|q_{\text{hip,xz}} - q_{\text{hip,xz}}^{\text{nominal}}\right\|^2$	-2
Feet Symmetry	$\mathbb{1}[\bar{\phi}_1 = \bar{\phi}_2] \left\|p_{\text{foot},0}^{\text{xz}} - p_{\text{foot},1}^{\text{xz}}\right\|^2$	-5
Termination	$\mathbb{1}$[Early Terminate]	-200

humanoid gaits:

$$r_t^{\text{contact}} = -\sum_{i=1}^{2}[1 - C(\phi_{t,i})]\left[1 - \exp\left(\left\|f_t^{\text{foot},i}\right\|^2 / \sigma_{cf}\right)\right] - \sum_{i=1}^{2} C(\phi_{t,i})\left[1 - \exp\left(\|v_{t,xy}^{\text{foot},i}\|^2 / \sigma_{cv}\right)\right] , \tag{8}$$

where $f_t^{\text{foot},i}$ denotes the foot contact force and $v_{t,xy}^{\text{foot},i}$ is the foot velocity. σ_{cf} and σ_{cv} are hyperparameters, fine-tuned according to the range of previous work [33] (We set the value as $\sigma_{cf} = 50$, $\sigma_{cf} = 5$). Note that during the stance phase, this reward function penalizes the foot velocities and ignores the foot contact force; on the other hand, during the swing phase, it penalizes the foot contact force and ignores the foot velocity.

Except for the contact in gait control, we also require the foot to smoothly reach the highest point and fall down, ensuring a precise and controllable swing. We introduce the foot trajectory reward r_t^{swing} to achieve this:

$$r_t^{\text{swing}} = \sum_{i=1}^{2}[1 - C(\phi_{t,i})]\left\|l_t^{\text{target},i} - l_t^{\text{foot},i}\right\|^2 . \tag{9}$$

Note that in Eq. (9), $l_t^{\text{foot},i}$ denotes the actual swing height of foot i, $C(\phi_{t,i})$ is the expected contact probability function. $l_t^{\text{target},i}$ is the target swing height, derived from a desired foot trajectory, as discussed below.

A desired foot trajectory should typically require the fulfillment of three key criteria: 1) zero foot velocity and acceleration during the stance phase; 2) zero foot velocity and acceleration at the end of the swing phase; and 3) continuity of both foot velocity and acceleration during the transition between the two phases. This is beneficial for enhancing motion stability and reducing energy consumption. In this work, we follow the experience in robot kinetics and quadruped robots [9, 46], and incorporate the quintic polynomial trajectory to compute the target swing height $l_t^{\text{target},i}$ at each control step:

$$l_t^{\text{target},i} = \begin{cases} l_t \sum_{k=0}^{5} a_k \left(0.25 - \left|\bar{\phi}_{t,i} - 0.75\right|\right)^k, & \bar{\phi}_{t,i} > 0.5 \\ 0, & \bar{\phi}_{t,i} < 0.5 \end{cases} . \tag{10}$$

Here l_t is the foot swing height command, and the polynomial coefficient a_k is determined based on the homogenized phase variable $\bar{\phi}_i$, as well as the boundary conditions of swing position, velocity, and acceleration. A detailed explanation of the calculation process is provided in the Appendix B-C. Note that Eq. (10) only defines the target trajectory in the z-axis. On natural terrains, the robot's precise foothold planning is not required. As for swing trajectories in the x-axis and the y-axis, which determines the stride, they can be computed based on the gait frequency f and the velocities v, ω [14, 10].

D. Mirror Function and Symmetry Loss

Natural and symmetrical motion behavior is gradually mastered by humans through acquired learning, due to its inherent elegance and efficiency in minimizing energy expenditure. Humanoid robots, with highly biomimetic mechanisms, also have symmetrical structural features. However, without prior knowledge, the symmetrical morphology information is difficult to be explored by the policy, especially for policies that generate diverse behaviors. This makes the initial exploration much more difficult, making the policy easily fall into local optima and leading to unnatural movements. To leverage the advantage of this morphological symmetry and inspired by [54], we proposed the mirror function $\mathcal{F}(\cdot)$ for a humanoid robot to encourage the policy to generate symmetric and natural motion. Under such a symmetrical structure, ideally, the policy output should satisfy:

$$\pi(o_t^\pi) = \mathcal{F}_a(\pi(\mathcal{F}_o(o_t^\pi))) . \tag{11}$$

Intuitively, the mirror function produces a mirror output symmetric to the X-Z plane. Here $\mathcal{F}_a$ and $\mathcal{F}_o$ are called *action mirror function* and *observation mirror function*, respectively, which map actions and observations to their mirrored version. Derived from these symmetric functions, we define a symmetry loss function $\mathcal{L}_{\text{sym}}$. The policy learning objective for controlling robots with symmetrical structures can be written as:

$$\mathcal{L}_{\text{sym}} = \sum_t \left\|\pi\left(o_t^\pi\right) - \mathcal{F}_a\left(\pi\left(\mathcal{F}_o\left(o_t^\pi\right)\right)\right)\right\|^2 , \tag{12}$$

The $\mathcal{L}_{\text{sym}}$ is independent of the RL objective, making it easy to extend to different RL algorithms. It is worth noting that the symmetric loss function is in fact encouraging symmetric actions on symmetric states (and commands), and it can be utilized for behaviors from symmetric ones (like walking and jumping) to asymmetric ones, such as hopping gaits, where hopping with the left foot is symmetric to hopping with the right one.

Overall training objective. HUGWBC adopt an asymmetric actor-critic framework [36], taking PPO [43] as the RL algorithm to train the whole-body policy. Therefore, the total training objective can be written as:

$$\mathcal{L} = \mathcal{L}_{\text{AAC}} + \beta \mathcal{L}_{\text{sym}}, \tag{13}$$

where β is a weight coefficient to balance between minimizing the RL objective and symmetry gait (we simply set β = 0.5 in our practice). We implemented a critic network, an actor network, along with the privileged encoder, all as Multi-Layer Perceptrons (MLPs). The actor network, combined with the encoder, can be directly deployed onto the real robot at a control frequency of 50 Hz. The sampled trajectories have a maximum length of 1000 timesteps, and the termination conditions include trunk collision with the ground or other links, as well as large body inclinations.

E. External Upper-Body Intervention Training

So far we learned a whole-body controller, which controls the upper and lower body jointly. Nevertheless, the goal of this work is not a controller specifically designed for locomotion tasks, but to build a unified and general humanoid controller that can serve as a basic support for loco-manipulation tasks. In other words, our controller should also support flexible and precise control of the upper body (arm and hands). Different from some previous works [18, 20] that augment the command space with upper body commands (*e.g.*, arm joint positions), we consider decoupling the upper body control as external control intervention by teleoperation signals [4, 29] or retargeted motion joints [3, 23], while not affecting the lower-body gaits, due to their high control precision. Our solution is sampling alternative actions to replace the upper-body actions produced by the whole-body policy during training, making the policy robust to any intervention.

Switching between whole-body control and intervention. Denote $I(t)$ a binary indicator function for whether the external control signal intervenes at each time step t, we assign a small probability of p ($p = 0.005$ in this work) to reverse $I(t)$. This leads the expected length of a continuous sequence without changing the upper-body control mode to be $\sum_{n=1}^{\infty} np(1-p)^n = \frac{1-p}{p}$, ensuring infrequently switching between two control modes and most of the trajectories are either long sequence of whole-body controlling or intervention, preventing rapid switches.

Intervention sampling. The intervened actions of the humanoid upper body are sampled from uniform noises, which introduce the potential for collisions with the body, simulating erroneous operations during external intervention.

Noise intervention interpolation. To prevent meaningless jitters caused by noise intervention sampling, the intervention action $a^{\text{targ}}_{\text{noise}}$ is randomly sampled in the action space every $t_{\text{interval}} = 90$ time steps. During the first two third time steps in the interval, linear interpolation is applied to smoothly transition the intervention joint positions from the initial pose $a^{\text{init}}_{\text{noise}}$ to the target pose $a^{\text{targ}}_{\text{noise}}$, while the target intervention action is maintained for the remaining time steps.

$$\begin{aligned} a^{\text{interv}}_{\text{noise},t} &= (1-r)a^{\text{init}}_{\text{noise}} + r a^{\text{targ}}_{\text{noise}} \ , \\ r &= \min\left(1, \frac{3}{2}\frac{t-t_0}{t_{\text{interval}}}\right) \ . \end{aligned} \tag{14}$$

In this equation, r is the ratio for the linear interpolation and t_0 is the sampled time.

Reward mask. When the intervention is involved, we mask the regularization rewards of the upper body during training, in order to eliminate the potential conflict of the policy output that tries to take over the upper body.

Noise curriculum. The replaced intervention action a^{interv}_t is gradually transited from the policy action a_t to the sampled noise $a^{\text{interv}}_{\text{noise},t}$:

$$a^{\text{interv}}_t = \alpha a^{\text{interv}}_{\text{noise},t} + (1-\alpha)a_t \ , \tag{15}$$

where the smoothing factor α increases per the progression of the intervention curriculum. In detail, α increases by 0.01 when both the linear and angular velocity tracking rewards exceed predefined thresholds. Conversely, if either of the velocity rewards fails to reach two-thirds of these thresholds, α is decreased by 0.01. The noise curriculum is illustrated in Fig. 5.

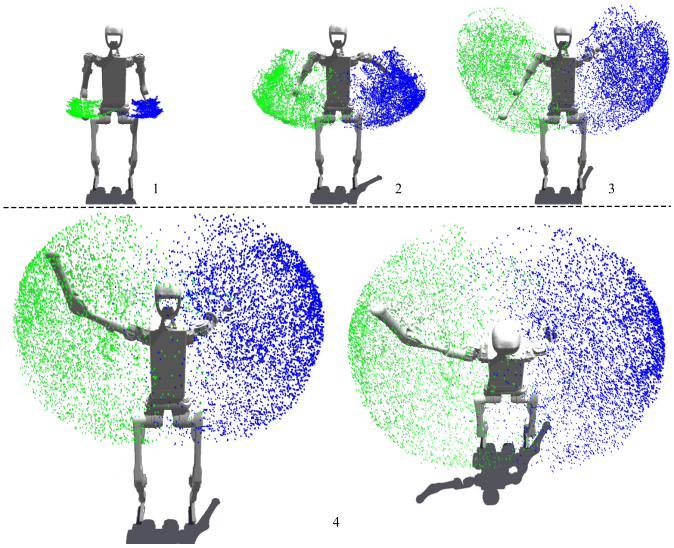


Fig. 5: **Intervention noise curriculum.** We illustrate sampled noise by visualizing the hand positions relative to the visualized robot hand joints. **Top 1-3**: Noise samples of three curriculum stages with noise levels ranging from small to large. These noises are only relative to the robot hand joints as visualized in the figures. **Bottom 4**: Front and top views of the noise samples from the final noise curriculum.

F. Curriculum Learning

Directly learning a diverse policy from manual rewards presents significant challenges due to the simultaneous optimization and exploration of multiple objectives. We thereby propose a curriculum learning approach to improve training efficiency. In particular, we split two distinct parallel robot training groups: an "agile group", tasked with learning high-speed, agile locomotion, and an "intervention group", focused

on developing a control policy for managing external upper-body interventions. At the beginning of training, each group of robots randomly samples one specific gait from four humanoid gaits, *i.e.*, standing, walking, jumping, hopping. The remaining behavioral commands $(f_t, l_t, h_t, p_t, w_t)$ and the task commands v_t are uniformly sampled from the specified ranges, which can be further referred to in Tab. II. Following [52], we employ a terrain curriculum for both groups, which consists of continuous rough terrain. Once the robot successfully masters the most challenging terrain, we keep that terrain and initiate an intervention noise curriculum and a speed curriculum simultaneously. On the one hand, the speed curriculum only works for the agile group, meant to learn high agility, which gradually increases the speed commands v_t following a grid adaptive curriculum strategy [32]. On the other hand, the intervention noise curriculum as described in Section IV-E works for the intervention group, focused on working with arbitrary upper-body intervention signals.

V. Simulations and Experiment

In this section, we conduct comprehensive experiments in both simulation and the real-world robot to address the following questions:

- **Q1(Sim)**: How does the HugWBC policy perform in tracking across different commands?
- **Q2(Sim)**: How to reasonably combine various commands in the general command space?
- **Q3(Sim)**: How does large-scale noise intervention training help in policy robustness?
- **Q4(Real)**: How does HugWBC behave in the real world?

Robot and Simulator. Our main experiments in this paper are conducted on the Unitree H1 robot, which has 19 Degrees of Freedom (DOF) in total, including two 3-DOF shoulder joints, two elbow joints, one waist joint, two 3-DOF hip joints, two knee joints, and two ankle joints. The simulation training is based on the NVIDIA IsaacGym simulator [31]. It takes 16 hours on a single RTX 4090 GPU to train one policy.

Command analysis principle and metric. One of the main contributions of this paper is an extended and general command space for humanoid robots. Therefore, we pay much attention to command analysis (regarding Q1 and Q2). This includes analysis of single command tracking errors, along with the combination of different commands under different gaits. For analysis, we evaluate the averaged episodic command tracking error (denoted as E_{cmd}), which measures the discrepancy between the actual robot states and the command space using L_1 norm. All commands are uniformly sampled within a pre-defined command range, as shown in Tab. II[2].

A. Single Command Tracking

We first analyze each command separately while keeping all other commands held at their default values. The results are shown in Tab. III. It is easily observed that the tracking errors in the walking and standing gaits are significantly lower than those in the jumping and hopping, with hopping exhibiting the largest tracking errors. For hopping gaits, the robot may fall during the tracking of specific commands, like high-speed tracking, body pitch, and waist-yaw control. This can be attributed to the fact that hopping requires rather high stability. Moreover, the complex postures and motions further exacerbate the risk of instability. Consequently, the policy prioritizes learning to maintain the balance, which, to some extent, compromises the accuracy of command tracking.

We conclude that the tracking accuracy of each gait aligns with the training difficulty of that gait in simulation. For example, the walking and standing patterns can be learned first during training, while the jumping and hopping gaits appear later and require an extended training period for the robot to acquire proficiency. Similarly, the tracking accuracy of robots under low velocity is significantly better than those under high velocity, since 1) the locomotion skills under low velocity are much easier to master, and 2) the dynamic stability of the robot decreases at high speeds, leading to a trade-off with tracking accuracy.

We also found that the tracking accuracy for longitudinal velocity commands v_x surpasses that of horizontal velocity commands v_y, which is due to the limitation of the hardware configuration of the selected Unitree H1 robots. In addition, the foot swing height l is the least accurately tracked. Furthermore, the tracking reward related to foot placement outperforms the tracking performance associated with posture control, since adjusting posture introduces greater challenges to stability. In response, the policy adopts more conservative actions to mitigate balance-threatening postural changes.

TABLE II: **Command ranges.** Ranges of curriculum starting, finishing, and default values of commands, for all gaits except hopping.

Group	Term	Default	Initial Range	Finishing Range
Task Commands	v_x	0	$[-0.6, 0.6]$	$[-0.6, 2.0]$
	v_y	0	$[-0.6, 0.6]$	$[-0.6, 0.6]$
	ω	0	$[-0.6, 0.6]$	$[-1.0, 1.0]$
Behavior Commands	f	2	$[1.5, 3.5]$	
	l	0.15	$[0.1, 0.35]$	
	h	0	$[-0.3, 0]$	
	p	0	$[0, 0.4]$	
	w	0	$[-1.0, 1.0]$	

B. Command Combination Analysis

To provide an in-depth analysis of the command space and to reveal the underlying interaction of various commands under different gaits. Here, we aim to analyze the *orthogonality* of commands based on the interference or conflict between the tracking errors of these commands across their reasonable ranges. For instance, when we say that a set of commands are *orthogonal*, each command does not significantly affect the tracking performance of each other in its range. To this end, we plot the tracking error E_{cmd} as heat maps, generated by systematically scanning the command values for each pair of

[2]Note that the hopping gait keeps a different command range, due to its asymmetric type of motion. More details can be referred to Appendix B-B.

TABLE III: **Single command tracking error.** The tracking errors for foot commands are calculated over a complete gait cycle, and the remaining ones are over one environmental step. For standing gait, we only tested the body height, body pitch, and waist yaw tracking error. E^{high} and E^{low} represents high-speed ($v_x > 1m/s$) and low-speed ($v_x \leq 1m/s$) modes categorized by the linear velocity v. The tracking error is computed by sampling each command in a predefined range (Tab. II) while keeping all other commands held at their default values.

Gait	Movement				Foot		Posture		
	$E^{\text{low}}_{v_x}$ (m/s)	$E^{\text{high}}_{v_x}$ (m/s)	E_{v_y} (m/s)	E_{ω} rad/s	E_f (HZ)	E_l (m)	E_h (m)	E_p (rad)	E_w (rad)
Standing	-	-	-	-	-	-	0.035	0.047	0.022
Walking	0.030	0.216	0.085	0.054	0.028	0.011	0.064	0.038	0.075
Jumping	0.090	0.532	0.069	0.077	0.027	0.012	0.058	0.048	0.022
Hopping	0.033	-	0.046	0.078	-	-	0.103	-	-

parameters, revealing the correlation of each command. We leave the full heat maps at Appendix C-A, and conclude our main observation for all gaits.

Walking. Walking is the most basic gait, which preserves the best performance of the robot hardware.

- The linear velocity v_x, the angular velocity yaw ω, the body height h, and the waist yaw w are orthogonal during walking.
- When the linear velocity v_x exceeds $1.5m/s$, the orthogonality between v_x and other commands decreases due to reduced dynamic stability and the robot's need to maintain body stability over tracking accuracy.
- The gait frequency f shows discrete orthogonality, with optimal tracking performance at frequencies of 1.5 or 2. High-frequency gait conditions reduce tracking accuracy.
- The linear velocity v_y, the foot swing height l, and the body pitch p are orthogonal to other commands only within a narrow range.

Jumping. The command orthogonality in jumping is similar to walking, but the overall orthogonal range is smaller, due to the increased challenge of the jumping gait, especially in high-speed movement modes. During each gait cycle, the robot must leap forward significantly to maintain its speed. To execute this complex jumping action continuously, the robot must adopt an optimal posture at the beginning of each cycle. Both legs exert substantial torque to propel the body forward. Upon landing, the robot must quickly readjust its posture to maintain stability and repeat the actions. Consequently, during movement, the robot can only execute other commands within a relatively narrow range.

Hopping. The hopping gait introduces more instability, and the robot's control system must focus more on maintaining balance, making it difficult to simultaneously handle complex, multi-dimensional commands.

- Hopping gait commands lack clear orthogonal relationships.
- Effective tracking is limited to the x-axis linear velocity v_x, the y-axis linear velocity v_y, the angular velocity yaw ω, and the body height h.
- Adjustments to h can be understood that a lower body height improves dynamic stability, therefore, it plays a positive role in maintaining the target body posture.

Standing. As for the standing gait, we tested the tracking errors of commands related to posture. The results showed that the tracking errors were similar to those observed during walking with zero velocity.

- The waist yaw w command is almost orthogonal to the other two commands.
- As the range of commands increases, orthogonality between the body height h and the body pitch p decreases. This is because the H1 robot has only one degree of freedom at the waist, limiting posture adjustments to the hip pitch joint.
- A 0.3 m decrease of the body height relative to the default height reduces the range of motion of the hip pitch joint to almost zero, hindering precise tracking of body pitch.

Furthermore, we conclude that gait frequency f highly affects the tracking accuracy of *movement* commands when it is excessively high and low; the *posture* commands can significantly impact the tracking errors of other commands, especially when they are near the range limits. For different gaits, the orthogonality range between commands is greatest in the walking gait and smallest in the hopping gait.

C. Ablation on Intervention Training Strategy

To validate the effectiveness of the intervention training strategy on the policy robustness when external upper-body intervention is involved, we compare the policies trained with different strategies, including noise curriculum (HUGWBC), filtered AMASS data [18], and no intervention. We test the tracking errors under two different intervention tasks, *i.e.*, uniform noise, AAMAS dataset, along with a no-intervention setup. The results under the walking gait are shown in Tab. IV, and we leave other gaits in Appendix C-B. It is obvious that the noise curriculum strategy of HUGWBC achieved the best performance under almost all test cases, except the posture-related tracking with no intervention. In particular, HUGWBC showed less of a decrease in tracking accuracy with various interventions, indicating our noise curriculum intervention strategy enables the control policy to handle a large range of arm movements, making it very useful and supportive for loco-manipulation tasks. In comparison, the policy trained with AMASS data shows a significant decrease in the tracking accuracy when intervening with uniform noise, due to the limited motion in the training data. The policy trained without any intervention only performs well without external upper-body control.

It is worth noting that when intervention training is involved, the tracking error related to the movement and foot is also better than those of the policy trained without intervention, and HUGWBC provides the most accurate tracking. This shows that intervention training also contributes to the robustness of the policy. During our real robot experiments, we further observed that the robot behaves with a harder force when in contact with the floor, indicating a possible trade-off between motion regularization and tracking accuracy when involving intervention.

Stability under standing gait. Adjusting posture in the standing state introduces additional requirements for stability, since the robot pacing to maintain balance may increase the difficulty of achieving manipulation tasks that require stand still. To

TABLE IV: **Tracking errors with different intervention strategies under the walking gait**. We evaluate three upper-body intervention training strategies: Noise (HUGWBC), the AMASS dataset, and no intervention at all. The tracking errors across various task and behavior commands reflect the intervention tolerance, *i.e.*, the ability of precise locomotion control under external intervention.

Training Strategy	Intervention Task	Task Commands			Behavior Commands				
		Movement			Foot		Posture		
		E_{v_x} (m/s)	E_{v_y} (m/s)	E_ω (rad/s)	E_f (Hz)	E_l (m)	E_h (m)	E_p (rad)	E_w (rad)
Noise Curriculum (HUGWBC)	Noise	**0.0483**	**0.0962**	**0.1879**	**0.0471**	**0.0542**	**0.0402**	**0.0432**	**0.0552**
	AMASS	**0.0391**	**0.0920**	**0.1039**	**0.0464**	**0.0543**	**0.0387**	**0.0364**	**0.0540**
	None	**0.0264**	**0.0863**	**0.0543**	**0.0447**	**0.0522**	0.0372	0.0375	0.0475
AMASS	Noise	0.1697	0.1055	0.2156	0.0621	0.0542	0.0620	0.0812	0.0694
	AMASS	0.0567	0.0965	0.1593	0.0466	0.0555	0.0579	0.0458	0.0554
	None	0.0645	0.0916	0.0802	0.0460	0.0531	0.0577	0.0455	0.0568
No Intervention	Noise	0.8658	0.7511	0.9116	0.1930	0.1913	0.1658	0.3622	0.2241
	AMASS	0.6299	0.4026	0.5758	0.2245	0.2527	0.1305	0.2367	0.1112
	None	0.0755	0.1076	0.1151	0.0450	0.0678	**0.0255**	**0.0211**	**0.0380**

TABLE V: **Averaged foot displacement under intervention**. We compare foot displacement D_{cmd} of different training strategies under various intervention tasks, which computes the total movement of both feet in one episode with sampled posture behavior commands.

Training Strategy	Intervention Task	D_h (m/s)	D_p (m/s)	D_w (m/s)
Noise Curriculum (HUGWBC)	Noise	**0.0339**	**0.0892**	**0.0199**
	AMASS	**0.0454**	**0.0728**	**0.0196**
	None	**0.0003**	**0.0016**	**0.0007**
AMASS only	Noise	2.0815	2.8978	3.2630
	AMASS	0.0536	0.1743	0.0396
	None	0.0139	0.0160	0.0013
No Intervention	Noise	17.5358	17.9732	25.7132
	AMASS	25.3802	26.3496	21.3078
	None	0.0159	1.7065	1.7152

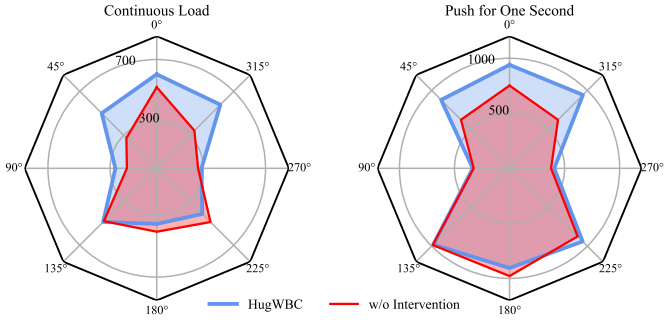


Fig. 6: **External disturbance tolerance**. Left: A constant and continuous force is applied to the robot. Right: A one-second force is exerted on the robot. The experiment is conducted under a standing gait with default commands. If the robot's survival ratio exceeds 98%, it is deemed capable of tolerating such external disturbance. The survival ratio computes the trajectory ratio of non-termination (ends of timeout) during 4096 rollouts.

investigate the necessity of noise curriculum for manipulation, we further measured the averaged foot displacement (in meters) under the standing gait, which computes the total movement of both feet in one episode (20 seconds) while tracking the posture behavior commands. Results in Tab. V show that HUGWBC exhibits minimal foot displacement. On the contrary, the strategy trained on AMASS data requires frequent small steps to adjust the posture and maintain stability for noise interventions. Without intervention training, the policy tends to tip over when involving intervention, leading to failure of the entire task.

Robustness for external disturbance. Finally, we test the contribution of intervention training and noise curriculum to the robustness of external disturbance. In particular, we evaluated the robot's maximum tolerance to external disturbance forces in eight directions and compared the policy trained without intervention. Results illustrated in Fig. 6 demonstrate that HUGWBC preserves greater tolerance for external disturbances in both pushing and loading scenarios across most of the directions. The reason behind this is that the intervention brings the robot exposed to various disturbances originating from its upper body, and thereby enhances the overall stability by dynamically adjusting leg strength.

D. Real-World Experiments

We deploy HUGWBC on a real-world robot to verify its effectiveness. In Fig. 1, we illustrate the humanoid capabilities supported by HUGWBC, showing the versatile behavior of the Unitree H1 robot. In particular, we demonstrate the intriguing potential of the comprehensive task range that HUGWBC is able to achieve, with a flexible combination of commands in high dynamics. To qualitatively analyze the performance of HUGWBC, we estimate the tracking error of two pose parameters (body pitch p and waist rotation w from the motor readings) on real robots, since other commands are hard to measure without a highly accurate motion capture system. The results are shown in Tab. VI, where $E_{\text{cmd}}^{\text{real}}$ illustrates the tracking error of the posture command. We observe that the tracking error in real-world experiments is slightly higher than in simulation environments, primarily due to sensor noise and the wear of the robot's hardware. Among different gaits, the tracking error for the waist rotation w is smaller compared to that for the body pitch p, as waist control has less impact on the robot's overall stability. In both error tests, the jumping gait exhibited the smallest E_{cmd}, while the walking gait showed slightly higher errors, consistent with the findings observed in the simulation environment.

TABLE VI: **Tracking error in real world.** We conducted five tests to measure the tracking error for each command under three gaits. The tracking error for each command was calculated during each control step. The tested commands gradually increased from the minimum to the maximum values within a predefined range, while the remaining commands were kept at their default values.

Gait	E_p^{real}	E_w^{real}
Standing	0.0712 ± 0.0425	0.0718 ± 0.0614
Walking	0.1006 ± 0.0581	0.0571 ± 0.0489
Jumping	0.0674 ± 0.0569	0.0552 ± 0.0469

VI. Conclusion and Limitations

We present a unified and general humanoid whole-body controller. Through an extended command space and intervention training, HugWBC enables versatile gait control while supporting external upper-body control, which can serve as a basic controller for extensive loco-manipulation tasks. Future works can adopt HugWBC to control various humanoid robots, or take the policy trained by HugWBC as a unified low-level controller to build a high-level planner to achieve complicated tasks.

Acknowledgments

We thank Jingxiao Chen, Xinyao Li, Jiahang Cao, and Xin Liu for their kind support of upper body control, motion generation, and demo recording. We thank anonymous reviewers for their kind suggestions. We thank Unitree for their help on the hardware.

References

[1] Justin Carpentier and Nicolas Mansard. Multicontact locomotion of legged robots. *IEEE Transactions on Robotics*, 34(6):1441–1460, 2018.

[2] Zixuan Chen, Xialin He, Yen-Jen Wang, Qiayuan Liao, Yanjie Ze, Zhongyu Li, S Shankar Sastry, Jiajun Wu, Koushil Sreenath, Saurabh Gupta, et al. Learning smooth humanoid locomotion through lipschitz-constrained policies. *arXiv preprint arXiv:2410.11825*, 2024.

[3] Xuxin Cheng, Yandong Ji, Junming Chen, Ruihan Yang, Ge Yang, and Xiaolong Wang. Expressive whole-body control for humanoid robots. *arXiv preprint arXiv:2402.16796*, 2024.

[4] Xuxin Cheng, Jialong Li, Shiqi Yang, Ge Yang, and Xiaolong Wang. Open-television: Teleoperation with immersive active visual feedback. *arXiv preprint arXiv:2407.01512*, 2024.

[5] Xuxin Cheng, Kexin Shi, Ananye Agarwal, and Deepak Pathak. Extreme parkour with legged robots. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11443–11450. IEEE, 2024.

[6] Suyoung Choi, Gwanghyeon Ji, Jeongsoo Park, Hyeongjun Kim, Juhyeok Mun, Jeong Hyun Lee, and Jemin Hwangbo. Learning quadrupedal locomotion on deformable terrain. *Science Robotics*, 8(74):eade2256, 2023.

[7] Xingye Da and Jessy Grizzle. Combining trajectory optimization, supervised machine learning, and model structure for mitigating the curse of dimensionality in the control of bipedal robots. *The International Journal of Robotics Research*, 38(9):1063–1097, 2019.

[8] Min Dai, Xiaobin Xiong, and Aaron Ames. Bipedal walking on constrained footholds: Momentum regulation via vertical com control. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 10435–10441, 2022.

[9] C. Dario Bellicoso, Christian Gehring, Jemin Hwangbo, Péter Fankhauser, and Marco Hutter. Perception-less terrain adaptation through whole body control and hierarchical optimization. In *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*, pages 558–564, 2016.

[10] Jared Di Carlo, Patrick M. Wensing, Benjamin Katz, Gerardo Bledt, and Sangbae Kim. Dynamic locomotion in the mit cheetah 3 through convex model-predictive control. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1–9, 2018. doi: 10.1109/IROS.2018.8594448.

[11] Pierre Fernbach, Steve Tonneau, Olivier Stasse, Justin Carpentier, and Michel Taïx. C-croc: Continuous and convex resolution of centroidal dynamic trajectories for legged robots in multicontact scenarios. *IEEE Transactions on Robotics*, 36(3):676–691, 2020.

[12] Zipeng Fu, Qingqing Zhao, Qi Wu, Gordon Wetzstein, and Chelsea Finn. Humanplus: Humanoid shadowing and imitation from humans. *arXiv preprint arXiv:2406.10454*, 2024.

[13] Zipeng Fu, Tony Z Zhao, and Chelsea Finn. Mobile aloha: Learning bimanual mobile manipulation with low-cost whole-body teleoperation. *arXiv preprint arXiv:2401.02117*, 2024.

[14] Christian Gehring, Stelian Coros, Marco Hutter, Michael Bloesch, Markus A Hoepflinger, and Roland Siegwart. Control of dynamic gaits for a quadrupedal robot. In *2013 IEEE international conference on Robotics and automation*, pages 3287–3292. IEEE, 2013.

[15] Ruben Grandia, Fabian Jenelten, Shaohui Yang, Farbod Farshidian, and Marco Hutter. Perceptive locomotion through nonlinear model-predictive control. *IEEE Transactions on Robotics*, 39(5):3402–3421, 2023. doi: 10.1109/TRO.2023.3275384.

[16] Robert J. Griffin, Georg Wiedebach, Stephen McCrory, Sylvain Bertrand, Inho Lee, and Jerry Pratt. Footstep planning for autonomous walking over rough terrain. In *2019 IEEE-RAS 19th International Conference on Humanoid Robots (Humanoids)*, pages 9–16, 2019.

[17] Xinyang Gu, Yen-Jen Wang, Xiang Zhu, Chengming Shi, Yanjiang Guo, Yichen Liu, and Jianyu Chen. Advancing humanoid locomotion: Mastering challenging terrains with denoising world model learning. *arXiv preprint arXiv:2408.14472*, 2024.

[18] Tairan He, Zhengyi Luo, Xialin He, Wenli Xiao, Chong

Zhang, Weinan Zhang, Kris Kitani, Changliu Liu, and Guanya Shi. Omnih2o: Universal and dexterous human-to-humanoid whole-body teleoperation and learning. *arXiv preprint arXiv:2406.08858*, 2024.

[19] Tairan He, Zhengyi Luo, Wenli Xiao, Chong Zhang, Kris Kitani, Changliu Liu, and Guanya Shi. Learning human-to-humanoid real-time whole-body teleoperation. *arXiv preprint arXiv:2403.04436*, 2024.

[20] Tairan He, Wenli Xiao, Toru Lin, Zhengyi Luo, Zhenjia Xu, Zhenyu Jiang, Jan Kautz, Changliu Liu, Guanya Shi, Xiaolong Wang, et al. Hover: Versatile neural whole-body controller for humanoid robots. *arXiv preprint arXiv:2410.21229*, 2024.

[21] Ayonga Hereid, Eric A. Cousineau, Christian M. Hubicki, and Aaron D. Ames. 3d dynamic walking with underactuated humanoid robots: A direct collocation framework for optimizing hybrid zero dynamics. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1447–1454, 2016.

[22] Fabian Jenelten, Junzhe He, Farbod Farshidian, and Marco Hutter. Dtc: Deep tracking control. *Science Robotics*, 9 (86):eadh5401, 2024.

[23] 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.

[24] Shuuji Kajita, Mitsuharu Morisawa, Kanako Miura, Shin'ichiro Nakaoka, Kensuke Harada, Kenji Kaneko, Fumio Kanehiro, and Kazuhito Yokoi. Biped walking stabilization based on linear inverted pendulum tracking. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4489–4496. IEEE, 2010.

[25] J. Koenemann, A. Del Prete, Y. Tassa, E. Todorov, O. Stasse, M. Bennewitz, and N. Mansard. Whole-body model-predictive control applied to the hrp-2 humanoid. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3346–3351, 2015.

[26] Minghuan Liu, Zixuan Chen, Xuxin Cheng, Yandong Ji, Ri-Zhao Qiu, Ruihan Yang, and Xiaolong Wang. Visual whole-body control for legged loco-manipulation. *arXiv preprint arXiv:2403.16967*, 2024.

[27] Xin Liu, Jinze Wu, Yufei Xue, Chenkun Qi, Guiyang Xin, and Feng Gao. Skill latent space based multigait learning for a legged robot. *IEEE Transactions on Industrial Electronics*, 2024.

[28] 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.

[29] 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. *arXiv preprint arXiv:2412.07773*, 2024.

[30] 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.

[31] 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.

[32] Gabriel Margolis, Ge Yang, Kartik Paigwar, Tao Chen, and Pulkit Agrawal. Rapid locomotion via reinforcement learning. In *Robotics: Science and Systems*, 2022.

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

[34] Carlos Mastalli, Rohan Budhiraja, Wolfgang Merkt, Guilhem Saurel, Bilal Hammoud, Maximilien Naveau, Justin Carpentier, Ludovic Righetti, Sethu Vijayakumar, and Nicolas Mansard. Crocoddyl: An efficient and versatile framework for multi-contact optimal control. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2536–2542, 2020.

[35] Avadesh Meduri, Paarth Shah, Julian Viereck, Majid Khadiv, Ioannis Havoutis, and Ludovic Righetti. Biconmp: A nonlinear model predictive control framework for whole body motion planning. *IEEE Transactions on Robotics*, 39(2):905–922, 2023.

[36] I Made Aswin Nahrendra, Byeongho Yu, and Hyun Myung. Dreamwaq: Learning robust quadrupedal locomotion with implicit terrain imagination via deep reinforcement learning. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5078–5084. IEEE, 2023.

[37] David E Orin, Ambarish Goswami, and Sung-Hee Lee. Centroidal dynamics of a humanoid robot. *Autonomous robots*, 35:161–176, 2013.

[38] Lerrel Pinto, Marcin Andrychowicz, Peter Welinder, Wojciech Zaremba, and Pieter Abbeel. Asymmetric actor critic for image-based robot learning. In *Robotics: Science and Systems*, 2018.

[39] Brahayam Ponton, Alexander Herzog, Stefan Schaal, and Ludovic Righetti. A convex model of humanoid momentum dynamics for multi-contact motion generation. In *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*, pages 842–849, 2016.

[40] Brahayam Ponton, Majid Khadiv, Avadesh Meduri, and Ludovic Righetti. Efficient multicontact pattern generation with sequential convex approximations of the centroidal dynamics. *IEEE Transactions on Robotics*, 37(5):1661–1679, 2021.

[41] Ilija Radosavovic, Tete Xiao, Bike Zhang, Trevor Darrell, Jitendra Malik, and Koushil Sreenath. Real-world humanoid locomotion with reinforcement learning. *Science Robotics*, 9(89):eadi9579, 2024.

[42] Nikita Rudin, David Hoeller, Philipp Reist, and Marco Hutter. Learning to walk in minutes using massively parallel deep reinforcement learning. In *5th Annual*

Conference on Robot Learning, 2021.

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

[44] Gerrit Schultz and Katja Mombaur. Modeling and optimal control of human-like running. *IEEE/ASME Transactions on mechatronics*, 15(5):783–792, 2009.

[45] Jonah Siekmann, Yesh Godse, Alan Fern, and Jonathan Hurst. Sim-to-real learning of all common bipedal gaits via periodic reward composition. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021.

[46] Jean-Pierre Sleiman, Farbod Farshidian, Maria Vittoria Minniti, and Marco Hutter. A unified mpc framework for whole-body dynamic locomotion and manipulation. *IEEE Robotics and Automation Letters*, 6(3):4688–4695, 2021.

[47] Daeun Song, Pierre Fernbach, Thomas Flayols, Andrea Del Prete, Nicolas Mansard, Steve Tonneau, and Young J. Kim. Solving footstep planning as a feasibility problem using l1-norm minimization. *IEEE Robotics and Automation Letters*, 6(3):5961–5968, 2021.

[48] Koushil Sreenath, Hae-Won Park, Ioannis Poulakakis, and Jessy 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.

[49] Jiayi Wang, Sanghyun Kim, Sethu Vijayakumar, and Steve Tonneau. Multi-fidelity receding horizon planning for multi-contact locomotion. In *2020 IEEE-RAS 20th International Conference on Humanoid Robots (Humanoids)*, pages 53–60, 2021.

[50] Patrick M Wensing and David E Orin. Improved computation of the humanoid centroidal dynamics and application for whole-body control. *International Journal of Humanoid Robotics*, 13(01):1550039, 2016.

[51] Alexander W. Winkler, C. Dario Bellicoso, Marco Hutter, and Jonas Buchli. Gait and trajectory optimization for legged systems through phase-based end-effector parameterization. *IEEE Robotics and Automation Letters*, 3(3):1560–1567, 2018.

[52] Jinze Wu, Guiyang Xin, Chenkun Qi, and Yufei Xue. Learning robust and agile legged locomotion using adversarial motion priors. *IEEE Robotics and Automation Letters*, 8(8):4975–4982, 2023. doi: 10.1109/LRA.2023.3290509.

[53] X Xinjilefu, Siyuan Feng, and Christopher G. Atkeson. Dynamic state estimation using quadratic programming. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 989–994, 2014.

[54] Wenhao Yu, Greg Turk, and C. Karen Liu. Learning symmetric and low-energy locomotion. *ACM Transactions on Graphics (TOG)*, 37(4), jul 2018.

[55] Chong Zhang, Wenli Xiao, Tairan He, and Guanya Shi. Wococo: Learning whole-body humanoid control with sequential contacts. *arXiv preprint arXiv:2406.06005*, 2024.

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

APPENDIX A
EXTENDED BACKGROUND

A. Proximal Policy Optimization

Proximal policy optimization (PPO) [43] is one of the popular algorithms that solve reinforcement learning (RL) problems. The goal of RL is to find the optimal policy $\pi^* : \mathcal{O} \times \mathcal{C} \rightarrow \mathcal{A}$ for command tracking that maximizes the expected discounted return:

$$\pi^* = \arg\max_{\pi} \mathbb{E}_{\pi} \left[\sum_{t=0}^{\infty} \gamma^t r(o_t, a_t, c_t) \right] \tag{16}$$

The basic idea behind PPO is to maximize a surrogate objective that constrains the size of the policy update. In particular, PPO optimizes the following objective:

$$\mathcal{L}_{\text{policy}} = \mathbb{E}_{\pi}[\min(rA, \text{clip}(r, 1-\epsilon, 1+\epsilon)A)], \tag{17}$$

where $r = \frac{\pi(a|o,c)}{\pi_{old}(a|o,c)}$ defines the probability ratio of the current policy and the old policy at the last optimization step, A is the advantage function, which is calculated by learning the value function:

$$\begin{aligned} &\mathcal{L}_{\text{value}} = \mathbb{E}_{\pi} \left[\|V_{\pi}(o,c) - V^{\text{targ}}(o,c)\|^2 \right] , \\ &A(o,a,c) = \sum_t \gamma^t r(o_t, a_t, c_t) - V(o,c)|_{o_0=o, a_0=a, c_0=c} , \end{aligned} \tag{18}$$

where V^{targ} is the target value function, defined as the expected return on the state o, c:

$$V^{\text{targ}}(o,c) = \mathbb{E}_{\pi} \left[\sum_t \gamma^t r(o_t, a_t, c_t) | o_0 = o, c_0 = c \right] \tag{19}$$

B. Asymmetric Training

The asymmetric training introduces a separate encoder to estimate the key privileged information s^{key} from k-step history proprioceptive observations h^k, which is trained by an estimation loss $\mathcal{L}_{\text{est}}$:

$$\mathcal{L}_{\text{est}} = \mathbb{E}_{\pi} \left[\|\mathcal{E}_{\pi}(h^k) - s^{\text{key}}\|^2 \right] \tag{20}$$

APPENDIX B
IMPLEMENTATIONS DETAILS

A. Unitree H1 DOF

The Unitree H1, as demonstrated in Fig. 7, has 19 DoFs in total, including two 3-DOF shoulder joints, two elbow joints, one waist joint, two 3-DOF hip joints, two knee joints, and two ankle joints.

B. Commands Space of The Hopping Gait

Hopping, characterized by a single foot consistently maintaining ground contact while the other remains in the air, represents an extremely unstable gait that requires coordinated whole-body motor control to maintain balance while tracking task commands. Among the behavior commands, all terms significantly challenge the delicate balance of the robot, except for body height, which poses minimal disruption. Therefore, the command space of hopping gait is designed as $\{v_x, v_y, \omega, h\}$, whose remaining behavior command terms for other gaits turn into regular terms. The command ranges and default value for gait hopping are illustrated in Tab. VII.

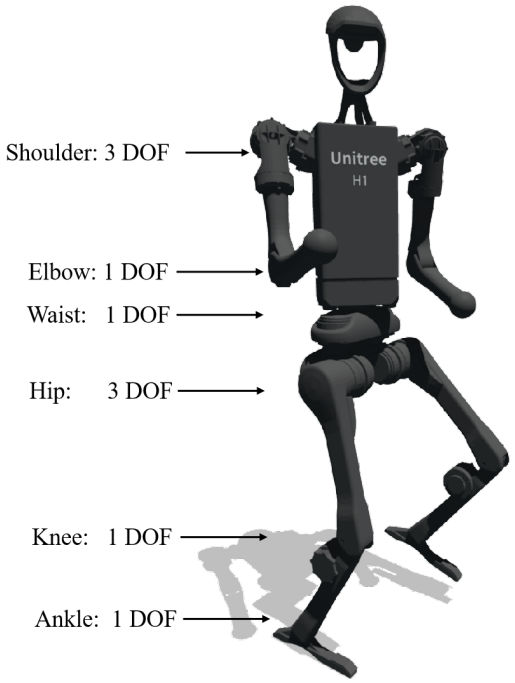


Fig. 7: DOF demonstration of Unitree H1.

TABLE VII: Ranges and default values of commands for gait hopping.

Group	Term	Default	Range
Movement	linear velocity v_x	0	$[-0.6, 0.6]$
	linear velocity v_y	0	$[-0.6, 0.6]$
	angular velocity ω	0	$[-0.6, 0.6]$
Posture	body height h	0	$[-0.3, 0]$

C. Foot Trajectory Target

There are various methods for robot foot trajectory planning, including Bezier trajectory, polynomial trajectory, and so on. Due to the smoothness provided by polynomial trajectory, they are widely used in the swing trajectory planning of quadruped robots [46]. Based on this, a polynomial foot trajectory planner integrated with homogeneous variables $\bar{\phi}_i$ is designed in this paper. In the z-axis, the swing trajectory is divided into two segments: from the starting position $p_{s,z}$ to the highest point l_t, and from the l_t to the end position $p_{e,z}$. In this study, a piecewise quintic polynomial is used for the foot trajectory planning. For the $p_{s,z}$ and $p_{e,z}$, it is desirable for the foot to make contact with the ground as smoothly as possible. Therefore, both velocity and acceleration are set to zero at these boundary points. The boundary conditions for the piecewise quintic polynomial trajectory are summarized in the Tab. VIII. The coefficients of a polynomial can be calculated:

TABLE VIII: **Boundary conditions** for z-direction quintic polynomial trajectory.

Time	Position	Velocity	Acceleration
t=$\phi_i^{0.5}$	$p_{s,z}$	0	0
t=$\phi_i^{0.75}$	l_t	0	0
t=$\phi_i^{1.0}$	$p_{e,z}$	0	0

$$\begin{bmatrix} p_{s,z} \\ l_t \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} (\bar{\phi}_i^{0.5})^5 & (\bar{\phi}_i^{0.5})^4 & (\bar{\phi}_i^{0.5})^3 & (\bar{\phi}_i^{0.5})^2 & \bar{\phi}_i^{0.5} & 1 \\ (\bar{\phi}_i^{0.75})^5 & (\bar{\phi}_i^{0.75})^4 & (\bar{\phi}_i^{0.75})^3 & (\bar{\phi}_i^{0.75})^2 & \bar{\phi}_i^{0.75} & 1 \\ 5(\bar{\phi}_i^{0.5})^4 & 4(\bar{\phi}_i^{0.5})^3 & 3(\bar{\phi}_i^{0.5})^2 & 2\bar{\phi}_i^{0.5} & 1 & 0 \\ 5(\bar{\phi}_i^{0.75})^4 & 4(\bar{\phi}_i^{0.75})^3 & 3(\bar{\phi}_i^{0.75})^2 & 2\bar{\phi}_i^{0.75} & 1 & 0 \\ 20(\bar{\phi}_i^{0.5})^3 & 12(\bar{\phi}_i^{0.5})^2 & 6\bar{\phi}_i^{0.5} & 2 & 0 & 0 \\ 20(\bar{\phi}_i^{0.75})^3 & 12(\bar{\phi}_i^{0.75})^2 & 6\bar{\phi}_i^{0.75} & 2 & 0 & 0 \end{bmatrix} \begin{bmatrix} a_5^1 \\ a_4^1 \\ a_3^1 \\ a_2^1 \\ a_1^1 \\ a_0^1 \end{bmatrix},$$
$$\begin{bmatrix} l_t \\ p_{e,z} \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} (\bar{\phi}_i^{0.75})^5 & (\bar{\phi}_i^{0.75})^4 & (\bar{\phi}_i^{0.75})^3 & (\bar{\phi}_i^{0.75})^2 & \bar{\phi}_i^{0.75} & 1 \\ (\bar{\phi}_i^{1.0})^5 & (\bar{\phi}_i^{1.0})^4 & (\bar{\phi}_i^{1.0})^3 & (\bar{\phi}_i^{1.0})^2 & \bar{\phi}_i^{1.0} & 1 \\ 5(\bar{\phi}_i^{0.75})^4 & 4(\bar{\phi}_i^{0.75})^3 & 3(\bar{\phi}_i^{0.75})^2 & 2\bar{\phi}_i^{0.75} & 1 & 0 \\ 5(\bar{\phi}_i^{1.0})^4 & 4(\bar{\phi}_i^{1.0})^3 & 3(\bar{\phi}_i^{1.0})^2 & 2\bar{\phi}_i^{1.0} & 1 & 0 \\ 20(\bar{\phi}_i^{0.75})^3 & 12(\bar{\phi}_i^{0.75})^2 & 6\bar{\phi}_i^{0.75} & 2 & 0 & 0 \\ 20(\phi_i^{1.0})^3 & 12(\phi_i^{1.0})^2 & 6\bar{\phi}_i^{1.0} & 2 & 0 & 0 \end{bmatrix} \begin{bmatrix} a_5^2 \\ a_4^2 \\ a_3^2 \\ a_2^2 \\ a_1^2 \\ a_0^2 \end{bmatrix}. \tag{21}$$

where $p_{s,z}$ is the z-coordinate of the start position, $p_{e,z}$ is the z-coordinate of the start position and l_t is swing highest position. The piecewise quintic polynomial trajectory $l_t^{\text{target},i}$ is formulated as:

$$l_t^{\text{target},i} = \begin{cases} \frac{6(l_t - p_{s,z})}{(\bar{\phi}_i^{0.75} - \bar{\phi}_i^{0.5})^5}(\bar{\phi}_i - 0.5)^5 + \frac{15(p_{s,z} - l_t)}{(\bar{\phi}_i^{0.75} - \bar{\phi}_i^{0.5})^4}(\bar{\phi}_i - 0.5)^4 + \frac{10(p_{s,z} - l_t)}{(\bar{\phi}_i^{0.75} - \bar{\phi}_i^{0.5})^3}(\bar{\phi}_i - 0.5)^3 + p_{s,z}, & 0.5 < \phi_i < 0.75 \\ \frac{6(p_{e,z} - l_t)}{(\bar{\phi}_i^{1.0} - \bar{\phi}_i^{0.75})^5}(1 - \bar{\phi}_i)^5 + \frac{15(l_t - p_{e,z})}{(\bar{\phi}_i^{1.0} - \bar{\phi}_i^{0.75})^4}(1 - \bar{\phi}_i)^4 + \frac{10(l_t - p_{e,z})}{(\bar{\phi}_i^{1.0} - \bar{\phi}_i^{0.75})^3}(1 - \bar{\phi}_i)^3 + l_t, & 0.75 < \phi_i < 1.0 \end{cases}, \tag{22}$$

D. Details of Intervention Baseline

In experiment section V-C, we compare HUGWBC with a baseline policy that is trained with intervention actions sampled from the AAMAS motion dataset.

Motion intervention interpolation. Since the frequency of the motion data is different from the control frequency, we interpolate the intervened actions from motion capture datasets to match the control frequency. Formally, at time step t, the intervention action is a linear interpolation of the closest two frames from the dataset:

$$a_{t,\text{dataset}}^{\text{interv}} = (1-\gamma)a_k^{\text{traj}_j} + \gamma a_{k+1}^{\text{traj}_j} \tag{23}$$

where

$$\gamma = \frac{f^{\text{traj}_j} \cdot t - T_k^{\text{traj}_j}}{T_{k+1}^{\text{traj}_j} - T_k^{\text{traj}_j}}$$

is the interpolation coefficient, $T_k^{\text{traj}_j}$ is the original time stamp of the k-th frame of j-th trajectory in the dataset and f^{traj_j} is the frequency of the j-th trajectory. The training process keeps the same curricula as described in Eq. (15) by replacing $a_{\text{noise}}^{\text{interv}}$ with $a_{\text{dataset}}^{\text{interv}}$.

E. Details of Network Architecture

We deployed an asymmetric training framework. HUGWBC actor network consists of three key components: a historical state encoder, a state estimator, and a low-level network. The historical state encoder takes in five frames of historical proprioceptive observations o_t^{his} and outputs an encoded historical vector z_t. The state estimator leverages this encoded vector to implicitly estimate linear velocity $\hat{v}_t$, foot clearance $\hat{l}_t$, and body height $\hat{h}_t$ that are often challenging to measure accurately with onboard sensors. Finally, the low-level network processes the z_t, the estimated states $\hat{v}_t$, $\hat{l}_t$, $\hat{h}_t$,current proprioceptive observations o_t^{pro}, the commands c_t and binary indicator $I(t)$, ultimately generating the joint actions a_t. A more detailed description of network architecture is shown in the Tab. IX.

TABLE IX: **Network architectures.**

Module	Inputs	Hidden Layers	Outputs
Historical State Encoder	o_t^{his}	[256, 128]	z_t
State Estimator	z_t	[64, 32]	$\hat{v}_t, \hat{l}_t, \hat{h}_t$
Low-Level Network	$z_t, \hat{v}_t, \hat{l}_t, \hat{h}_t, o_t^{\text{pro}}, c_t, I(t)$	[256, 128, 64]	a_t
Critic	$o_t^{\text{pro}}, o_t^{\text{pri}}, o_t^{\text{ter}}$	[512, 256, 128]	V_t

F. Policy Learning Time

The overall policy learning time was 16 hours of wall-clock time, using a single NVIDIA RTX 4090 GPU.

APPENDIX C
EXTENDED EXPERIMENT

A. Extensive Analysis of Commands Combination

We draw heatmaps and line charts to illustrate the tracking accuracy when combining two different commands across their ranges under different gaits, shown in Fig. 8.

B. Commands Tracking with Interventions

We further show the single command tracking evaluation results for the standing gait and the jumping gait, in Tab. X. On these gaits, HUGWBC also achieves the best tracking performance under almost all test cases, except the body pitch and waist yaw tracking with no intervention. In contrast, the policy trained with AMASS data is still limited to handling actions within the scope of that data, and the policy trained without intervention fails with any external upper-body control. We thus can conclude that intervention training applies to a variety of gaits.

In particular, under the jumping gait, the intervention tasks had a significant impact on the robot tracking performance. This is mainly because jumping gait is more challenging for humanoid robots, which rely heavily on arm swings to complete the motion task. Therefore, when the arm movement is restricted, the robot's performance is notably compromised. Under the standing gait, HUGWBC shows significantly lower posture-related tracking errors compared to the walking and jumping gaits.

Since hopping is a highly unstable gait, which is rarely used for loco-manipulation tasks and is implemented with an independent policy, we did not involve intervention training for the hopping gait.

C. Comparison with Other Whole-body Controllers

We compare HUGWBC with two *open-sourced SOTA learning-based humanoid whole-body controllers*, HOVER [20] and Exbody [3] in simulation, shown in Tab. XI. Nevertheless,

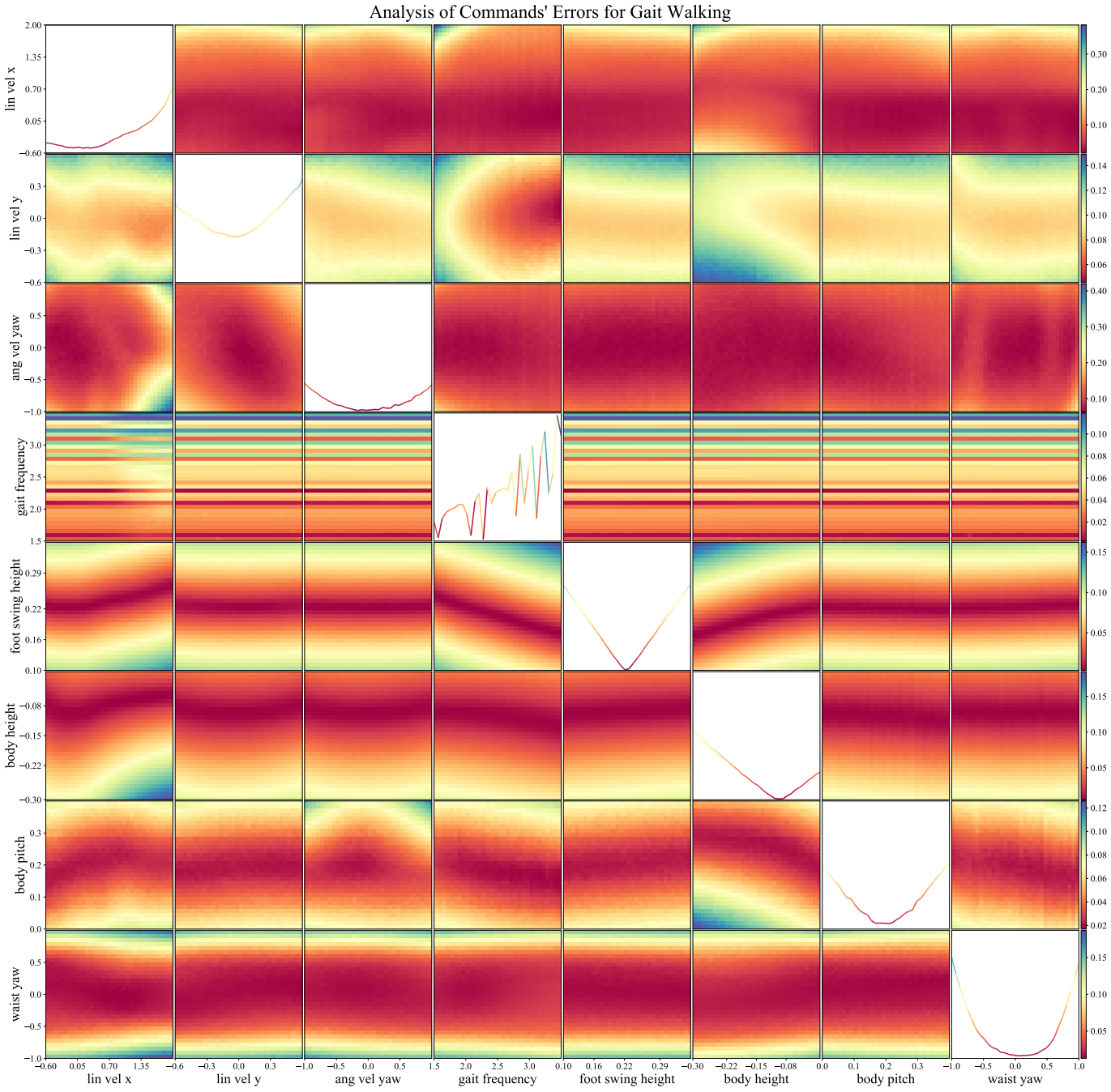
TABLE X: **Tracking error with different intervention strategies under the standing gait and the jumping gait**. We evaluate three upper-body intervention training strategies: noise curriculum (HUGWBC), the AMASS dataset, and no intervention at all. The tracking errors across various tasks and behavior commands reflect the intervention tolerance, *i.e.*, the ability of precise locomotion control under external intervention.

Gait	Training Strategy	Intervention Task	Movement			Foot		Posture		
			E_{v_x} (m/s)	E_{v_y} (m/s)	E_ω (rad/s)	E_f (Hz)	E_l (m)	E_h (m)	E_p (rad)	E_w (rad)
Standing	Noise Curriculum (HUGWBC)	Noise	-	-	-	-	-	**0.0291**	**0.0662**	**0.0605**
		AMASS	-	-	-	-	-	0.0237	**0.0611**	**0.0624**
		None	-	-	-	-	-	0.0228	0.0476	0.0564
	AMASS	Noise	-	-	-	-	-	0.0301	0.1222	0.0875
		AMASS	-	-	-	-	-	**0.0204**	0.0782	0.0707
		None	-	-	-	-	-	**0.0198**	0.0778	0.0727
	None	Noise	-	-	-	-	-	0.1931	0.4051	0.2283
		AMASS	-	-	-	-	-	0.1357	0.2571	0.1243
		None	-	-	-	-	-	0.0213	**0.0218**	**0.0514**
Jumping	Noise Curriculum (HUGWBC)	Noise	**0.0886**	**0.1078**	**0.1785**	**0.0580**	**0.0457**	**0.0411**	**0.0471**	**0.0527**
		AMASS	**0.0750**	**0.0729**	**0.1010**	**0.0487**	**0.0458**	**0.0402**	**0.0417**	**0.0519**
		None	**0.0504**	**0.0606**	**0.0778**	**0.0556**	**0.0445**	**0.0417**	0.0491	0.0511
	AMASS	Noise	0.3026	0.1179	0.2736	0.0591	0.2560	0.0424	0.1217	0.0757
		AMASS	0.0826	0.0759	0.1104	0.0553	0.0469	0.0428	0.0486	0.0536
		None	0.0854	0.0743	0.0804	0.0583	0.0461	0.0431	0.0476	0.0551
	None	Noise	0.8082	0.5533	0.8340	0.0717	0.6358	0.1931	0.4051	0.2283
		AMASS	0.8632	0.4105	0.6888	0.0787	0.7720	0.1357	0.2591	0.1243
		None	0.0711	0.0976	0.1127	0.0625	0.3255	0.0449	**0.0368**	**0.0375**

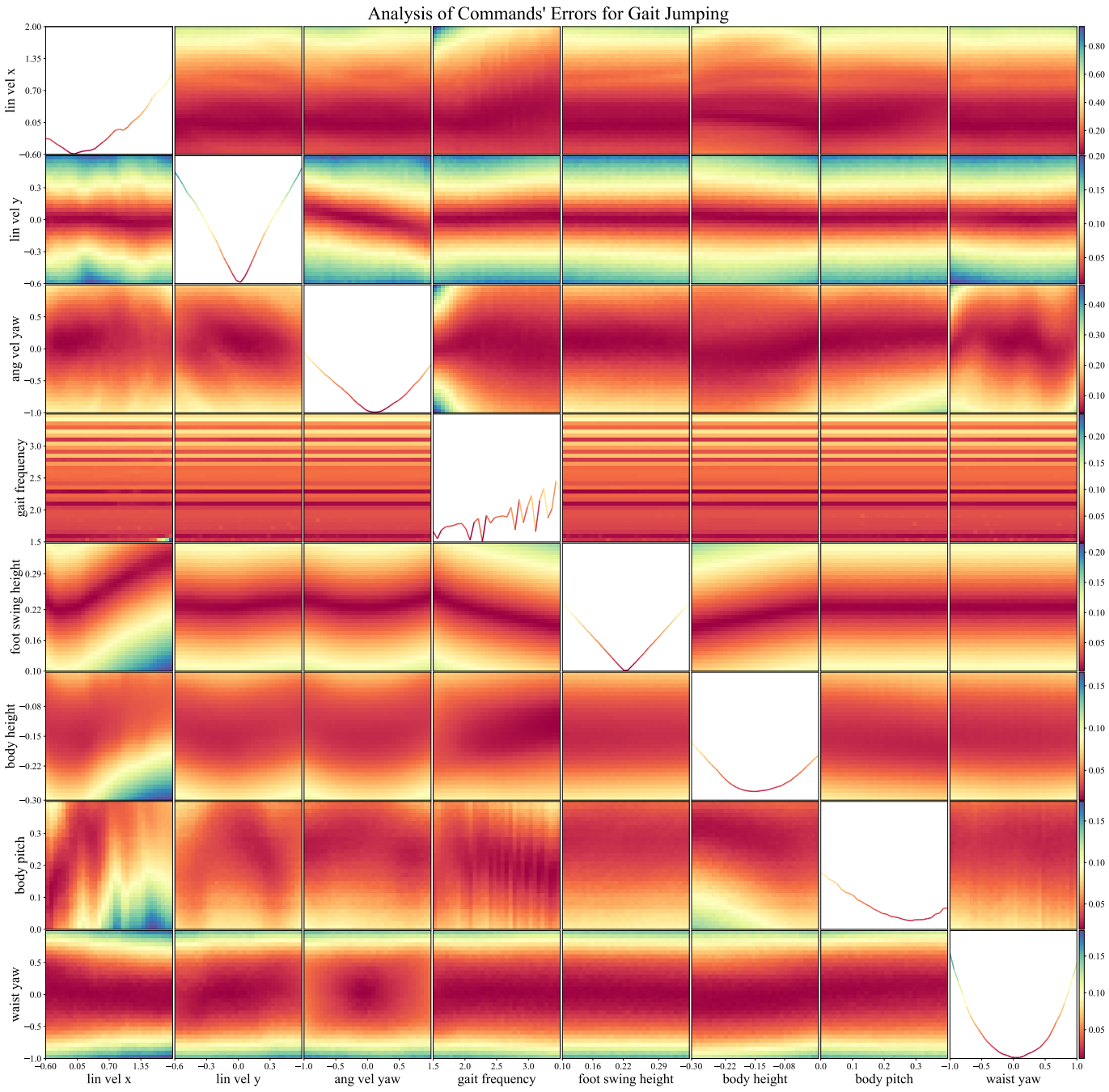
the training and control modes of these controllers rely heavily on motion datasets. For example, the command space of ExBody includes target expression goal (upper body) and root movement goal (lower body), which are sampled from trajectories. Although HOVER features a multi-mode command space, it requires high consistency between different command terms due to the motion tracking task setting. To compare them to ours, we keep the upper body of humanoids remaining at the default joint positions as the required reference, and compute the tracking error as in Tab. III. Note that we evaluate the performance of HUGWBC under the *walking* gait, as the baselines do not support gait switch without reference motion, and HUGWBC does not require upper-body reference motion and controls the whole-body joints. The comparison experiment forces HOVER and ExBody policies to perform tasks beyond their intended design, resulting in poorer performance than demonstrated in their respective papers.

TABLE XI: Single command tracking error comparison with learning based baselines.

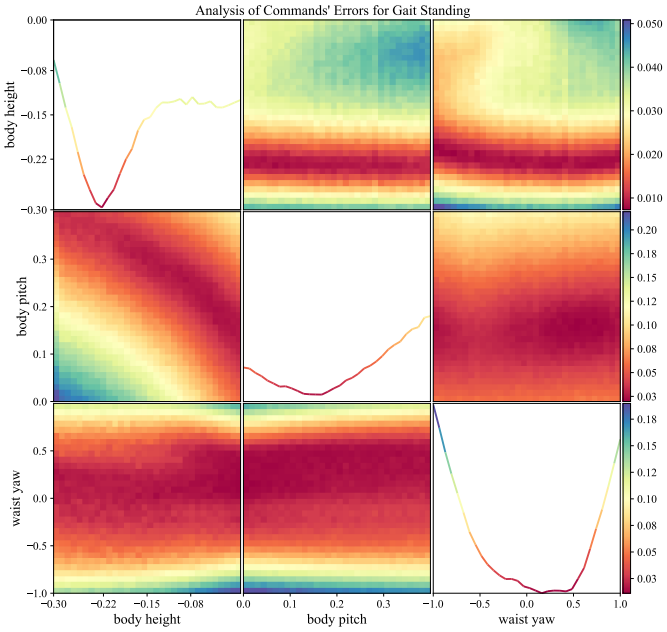
Methods	$E_{v_x}^{\text{low}}$	$E_{v_x}^{\text{high}}$	E_{v_y}	E_ω	E_h	E_p	E_w
HOVER [20]	0.559	1.324	0.328	0.436	0.270	0.127	0.082
ExBody [3]	0.109	0.242	0.114	0.587	0.145	0.122	0.097
HUGWBC (Ours)	**0.030**	**0.216**	**0.085**	**0.054**	**0.064**	**0.038**	**0.075**


Analysis of Commands' Errors for Gait Walking
lin vel x
lin vel y
ang vel yaw
gait frequency
foot swing height
body height
body pitch
waist yaw

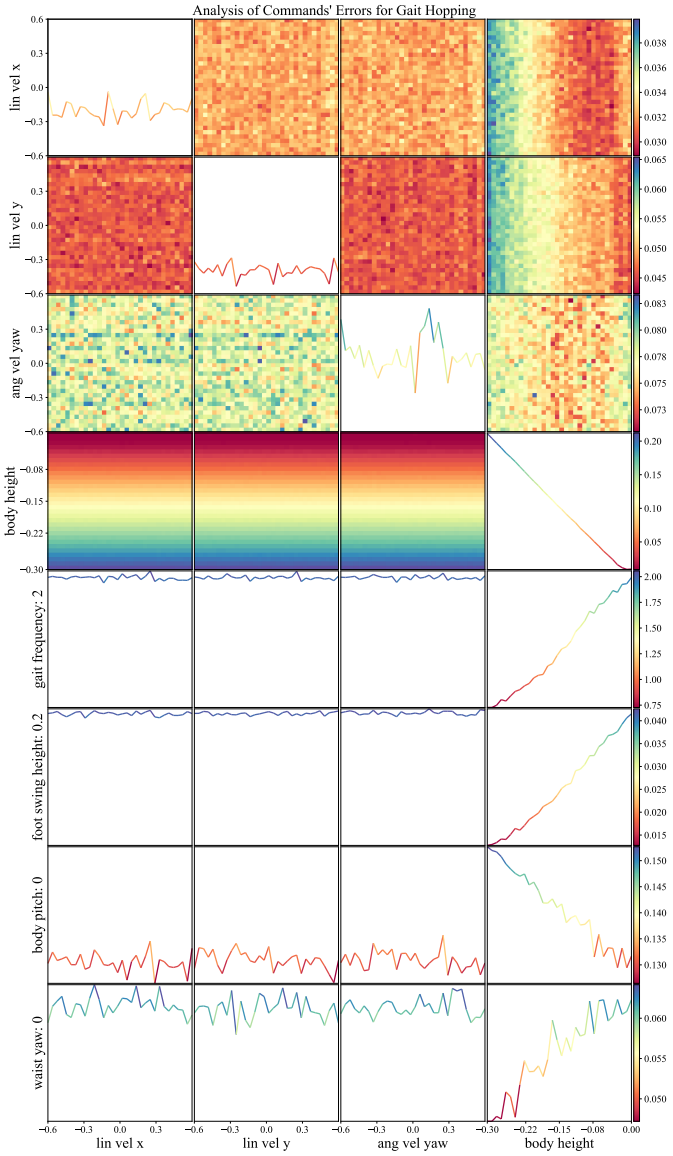
(a) Walking.


Analysis of Commands' Errors for Gait Jumping
lin vel x
lin vel y
ang vel yaw
gait frequency
foot swing height
body height
body pitch
waist yaw

(b) Jumping.



(c) Standing.



(d) Hopping.

Fig. 8: **Tracking-error heat maps of command combination under different gaits**. Each column represents one of the following command parameters: *linear velocity x*, *linear velocity y*, *angular velocity yaw*, *gait frequency*, *foot swing height*, *body height*, *body pitch*, and *waist roll*. The standing includes the three series commands for the *body height*, *body pitch*, and *waist yaw*. For the off-diagonal sub-figures, the range for each command is indicated along the vertical axis (left) and horizontal axis (bottom). The corresponding error values are indicated by ticks on the right-side color bar. The colder the color of the pixel, the larger the tracking error the commands faces, and the color bars in different rows have different ranges of error.