Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Dual-Level Humanoid Whole-Body Controller

Ziluo Ding

ziluo@pku.edu.cn

Beijing Academy of Artificial Intelligence

Haobin Jiang

Beijing Academy of Artificial Intelligence

Yuxuan wang

Beijing Academy of Artificial Intelligence

yu zhang

Beijing Academy of Artificial Intelligence

Ming Yang

Beijing Academy of Artificial Intelligence

xinrun xu

Beijing Academy of Artificial Intelligence

Research Article

Keywords: Whole-body Control, Multi-agent Reinforcement Learning

Posted Date: November 2nd, 2026

DOI: https://doi.org/10.21203/rs.3.rs-10959555/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Dual-Level Humanoid Whole-Body Controller

Ziluo Ding[1*†], Haobin Jiang[1†], Yuxuan Wang[1†], Yu Zhang[1], Ming Yang[1], Xinrun Xu[1]

[1]Beijing Academy of Artificial Intelligence (BAAI), Beijing, China.

*Corresponding author(s). E-mail(s): ziluo@pku.edu.cn;
†Ziluo Ding, Haobin Jiang, and Yuxuan Wang contributed equally to this work.

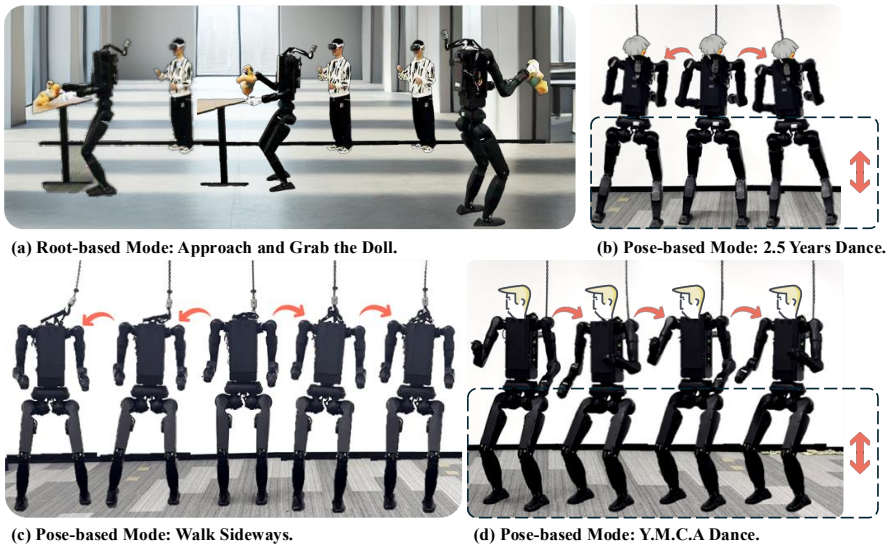


Fig. 1: Some real-world demonstrations of JAEGER deployed on the H1-2. For the root-based mode, we use a VR and a joystick to control the upper and lower body of a humanoid for grasping objects while in motion **(a)**. In the pose-based mode, the robot can perform various dance movements **(b, d)**, requiring coordination between the upper and lower body. It can also execute large-amplitude lower-body movements, such as walking forward and sideways **(c)**.

Abstract

This paper presents JAEGER, a dual-level whole-body controller for humanoid robots that addresses the challenges of training a more robust and versatile policy. Unlike traditional single-controller approaches, JAEGER separates the control of the upper and lower bodies into two independent controllers, so that they can better focus on their distinct tasks. This separation alleviates the dimensionality curse and improves fault tolerance. JAEGER supports both root velocity tracking (coarse-grained control) and local joint angle tracking (fine-grained control), enabling versatile and stable movements. To train the controller, we utilize a human motion dataset (AMASS), retargeting human poses to humanoid poses through an efficient retargeting network, and employ a curriculum learning approach. This method performs supervised learning for initialization, followed by reinforcement learning for further exploration. We conduct our experiments on two humanoid platforms and demonstrate the superiority of our approach against state-of-the-art methods in both simulation and real environments as partially illustrated in Figure 1. More demonstrations of JAEGER, including retargeting results, root-based mode and pose-based mode, can be found in the link: **https://anonymous.4open.science/r/JAEGER/**.

Keywords: Whole-body Control, Multi-agent Reinforcement Learning

1 Introduction

Humanity has long dreamed of using humanoid robots to assist or replace humans in completing various daily tasks, including grabbing objects, carrying goods, etc. Compared to wheeled robots, humanoid robots have higher degrees of freedom (DoF) to their lower bodies, allowing them to perform more complex whole-body motions. Specifically, they can overcome complex terrains like uneven ground and stairs, and perform a rich repertoire of skills and motions exhibited by humans, such as dancing and boxing. In addition, the alignment of the embodiment between humanoid robots and humans facilitates the seamless utilization of human Internet data, enabling cognitive learning abilities, and the operation of existing machinery and devices designed for human use.

Due to hardware constraints and the inherent complexity of the robotic action space, achieving effective whole-body control (WBC) for adult-sized humanoid robots, such as the Unitree H1-2, remains a significant challenge. Recent studies on WBC have demonstrated promising advancements, enabling humanoid robots to perform versatile motions by learning from extensive human data [3, 6–8, 10]. Based on different task settings, WBC methodologies can be broadly categorized into three types: root velocity tracking [2], kinematic position tracking [6, 7], and local joint angle tracking [2, 3, 10]. Root velocity tracking emphasizes *coarse-grained control*, where the robot tracks a given velocity *without* relying on a specific reference pose. In contrast, kinematic position and local joint angle tracking focus on accurately reproducing a given trajectory of reference poses, which can be regarded as *fine-grained control* for

humanoids. Note that both the kinematic position and the local joint angle can be used to describe the pose of the humanoid at a given moment.

Intuitively, a more versatile WBC controller should integrate both coarse-grained and fine-grained commands, regardless of whether they are represented through positions or angles. However, *the challenge of effectively combining these two approaches remains unresolved.* Current approaches, such as OmniH2O [6] and HumanPlus [3], focus on learning locomotion by following reference pose trajectories and deriving velocity tracking capabilities from these trajectories. However, in practice, since the inevitable differences between embodiments of humanoid robots and humans, locomotion by fully imitating human pose might hinder humanoids from maintaining balance and prevent them from learning a robust policy with the capability of velocity tracking. Therefore, we propose to firstly separately train a root velocity tracking policy and a local joint angle tracking policy, and then distill the two policies into a unified WBC policy.

Given the complexity of the WBC problem and the heterogeneous functionalities of the upper and lower bodies, we reformulate the WBC problem as a multi-agent system [14, 22, 24]. During training, we observed that interactions between the upper and lower bodies often hinder convergence; for instance, the upper body's tracking behavior becomes overly conservative to accommodate the lower body's stability. Addressing this interference typically requires intricate parameter tuning and curriculum learning. Inspired by the multi-agent system, we propose decoupling the upper and lower bodies into two independent controllers, enabling each to focus on its specific tasks while alleviating the curse of dimensionality in action space [5]. This design not only simplifies the training process but also enhances the system's fault tolerance, allowing the lower body to continue functioning effectively even if the upper body encounters failure.

In this paper, we present JAEGER, a dual-level humanoid whole-body controller that supports both root velocity tracking, *i.e.* coarse-grained control, and local joint angle tracking, *i.e.* fine-grained control. Given the upper and lower bodies serve different functions, we use two controllers to manage them separately for a more robust WBC policy. To achieve effective coordination, the two controllers share both observations and rewards. For the training procedure, we follow recent success in leveraging a human motion dataset, AMASS [15]. We first retarget human poses to humanoid poses via our efficient retargeting network and then train the two controllers to follow these humanoid poses. Given the inherent challenges of WBC training, we adopt a curriculum learning approach. Initially, the lower-body controller is trained independently. Next, we implement a supervised initialization for the upper-body controller, followed by the whole-body control training using reinforcement learning (RL).

To summarize, our contributions are as follows: 1) We propose an MLP-based retargeting approach that outperforms optimization-based inverse kinematics (IK) methods by producing more accurate and smoother joint angles for humanoid control, while operating at a high execution frequency of up to 1 kHz. 2) We introduce a novel dual-level WBC controller, which reduces mutual interference between the upper and lower bodies while enabling each controller to focus more on its respective task. 3) To train the dual-level controller, we propose a structured curriculum learning strategy.

A tailored supervised initialization serves as the first stage before RL, which facilitates the following RL convergence to a more optimal solution. 4) Experiments in both simulation and on a real humanoid robot demonstrate that JAEGER delivers superior multi-mode control compared to other baselines.

2 Related work

2.1 Humanoid Whole-Body Control

Whole-body control for humanoid robots remains a challenging problem due to hardware limitations and the high complexity of the action space. Compared with traditional dynamics modeling and control methods [4, 16], recent reinforcement learning and sim-to-real transfer methods have advanced this field and demonstrated promising results in enabling complex whole-body motion [2, 6, 8, 10–12, 21, 23].

As pioneers in this field, the H20 series [6, 7] fully rely on global keypoint position tracking. However, the global keypoint positions may gradually drift over time, leading to cumulative errors that ultimately impede learning. HumanPlus [3], on the other hand, fully relies on joint angle tracking to learn motion and autonomous skills from human data. ExBody series combines the strengths of both approaches. In more detail, ExBody [2] expresses the diverse upper-body motion by tracking the joint angles and keypoint positions, while tracking the root velocity for the lower body. ExBody2 [10] achieves superior performance by only learning from a curated subset of the AMASS dataset. It shows the poses that are lower-body feasible are crucial for training stable behaviors. In addition, it claims that the local keypoint position tracking can lead to a more robust strategy. HOVER [8], as a unified framework, integrates all the interfaces, *e.g.,* root velocity, joint angles, and keypoint positions, into a single network through a distillation process. Our work is concurrent with HOVER and shares some similarities, particularly in the integration of root velocity and joint angle control. However, we train this more challenging controller from a different perspective, namely by decoupling the upper and lower bodies.

2.2 Humanoid Locomotion

Training a robust locomotion model is the core task for the humanoid. With the rise of learning algorithms, reinforcement learning (RL) and imitation learning (IL) have proven to be more robust than traditional model-based control methods [1, 13, 25]. By designing effective reward functions, RL has shown promise in enabling humanoids to learn complex locomotion skills. Recently, transformers [19] have been used for humanoid locomotion, relying on next-token prediction. In addition, an end-to-end vision-based whole-body control parkour policy [25] has been proposed to overcome multiple parkour skills without any motion prior.

IL can address reward design challenges by learning from human demonstrations. Advanced techniques like Generative Adversarial Imitation Learning (GAIL) [9] allow for more adaptable skills from unstructured data. An IL framework using adversarial motion priors has been proposed to demonstrate human-like locomotion. In addition,

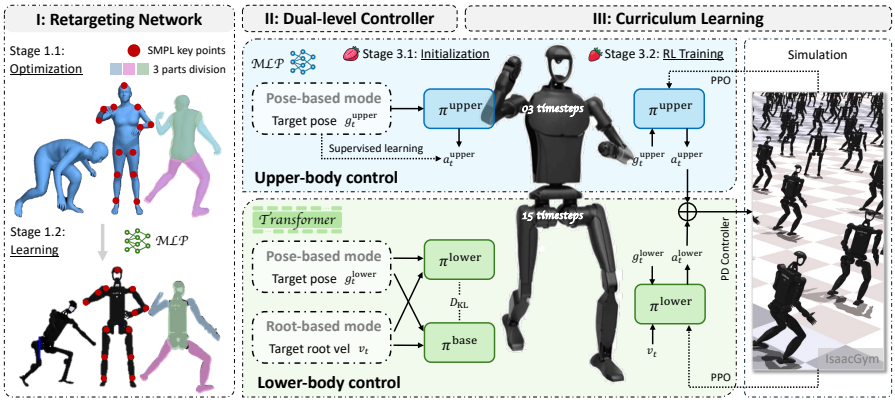


Fig. 2: The framework of JAEGER. The left shows the retargeting network, which processes the AMASS dataset through optimization to generate massive human and robot pose pairs, and then uses an MLP to learn the relationship between them. The middle section illustrates how we control the upper and lower body parts using two separate controllers. The right section presents the curriculum learning paradigm, where simple controllers are first trained using supervised learning and then improved through RL to explore a better policy.

Radosavovic et al. [20] constructed a large locomotion dataset, and a transformer was employed to imitate the demonstrations within it.

3 Method

We propose JAEGER, an effective framework for humanoid whole-body control. As illustrated in Figure 2, The framework of JAEGER consists of three main components: a retargeting network, a dual-level controller, and the curriculum learning procedure. Our whole-body controller supports two command modes: 1) **root-based mode**, which includes root velocity commands and reference joint angle commands for the upper body, and 2) **pose-based mode**, which provides reference joint angle commands for the entire body. Specifically, the root velocity command includes linear forward and sideways velocities, as well as angular turning velocity.

3.1 Retargeting Network

Retargeting is an essential data preprocessing procedure. At the same time, real-time retargeting is also crucial for low-latency teleoperation. **In this section, we want to propose a different perspective on retargeting for low-latency motion adaptation.** Specifically, we adopt deep mimic methods instead of directly using traditional IK techniques.

We first adopt a two-stage optimization approach as H2O [7] to process the AMASS dataset [15], generating a large set of human and humanoid pose pairs. Then, a

lightweight three-layer MLP is employed to learn the mapping between them. This approach significantly improves the real-time performance of the retargeting process while maintaining pose transfer accuracy. Unlike optimization methods used in prior works, MLP ensures smooth output even with small noise perturbations on input, eliminating frame-to-frame jitter, which is verified in Section 4.4. This is because the MLP is a continuous nonlinear function approximator and is trained on massive filtered motion data. Specifically, we manually remove frames with discontinuous motions resulting from unstable IK solutions. Note that, for the data preprocessing procedure, we retarget the motion sequences from the AMASS dataset and also apply the same filtering process as HumanPlus [3] to ensure high-quality retargeting data.

3.2 Dual-level Controller

JAEGER aims to track the reference pose and root velocity more precisely and robustly. To this end, we implement a dual-level controller that separately manages the upper and lower bodies. Intuitively, this approach decomposes the action space, thus reducing the learning complexity. In addition, the independent controllers offer greater flexibility in selecting their individual network architectures.

3.2.1 Setting Formulation

We formulate the humanoid whole-body control problem as a Decentralized Partially Observable Markov Decision Process (Dec-POMDP) [17] with two agents. Specifically, one agent controls the lower body using the policy π^{lower}, and the other one controls the upper body using the policy π^{upper}. The observation at timestep t consists of proprioceptive state o_t, previous action $a_{t-1} = [a_{t-1}^{\text{lower}}, a_{t-1}^{\text{upper}}]$, root velocity tracking target v_t, and the pose tracking target g_t. Both policies take as input a n-step observation history $\boldsymbol{x}_{t,n} = \{\boldsymbol{o}_{t-n:t}, \boldsymbol{a}_{t-n-1:t-1}, \boldsymbol{v}_{t-n:t}, \boldsymbol{g}_{t-n:t}\}$, and output action a_t^{lower} and a_t^{upper}, respectively. The predicted actions are then concatenated to form the target joint positions of joint proportional derivative (PD) controllers.

3.2.2 Lower-body Controller

The task of the lower body is particularly challenging, as it not only needs to track the root velocity and reference pose, but also must maintain overall body balance simultaneously. Recent works

Specifically, we adopt the Gated Transformer-XL architecture [18], which consists of three layers of transformer blocks. This Transformer variant is designed to better suit RL tasks by replacing standard residual connections with gating mechanisms within the transformer block. These gates can substantially improve the learning stability and speed in RL. Note that the lower-body controller takes as input the 15-step observation history $\boldsymbol{x}_{t,15}$.

3.2.3 Upper-body Controller

As the primary task of the upper-body controller is to track the reference pose, we implement it as a lightweight 3-layer MLP, reducing computational resource usage. The

upper-body controller only takes as input the concatenation of the 3-step observation history $\boldsymbol{x}_{t,3}$.

3.3 Curriculum Learning

Training a root-based WBC controller from scratch with RL is relatively straightforward. However, when simultaneously training both lower-body root velocity tracking and upper-body pose tracking, we observe significant instability. This is likely due to the distinct objectives of the upper and lower bodies, which interfere with each other during training. It highlights the necessity of curriculum learning. **While curriculum learning is a well-established strategy in RL, our contribution lies in a task-specific, structured initialization designed for dual-level WBC.**

3.3.1 Base policy

Specifically, as training a lower-body controller independently is much easier, we first train it while keeping the upper-body joints fixed, resulting in a base policy π^{base}. The base policy can either be a locomotion policy or a joint angle tracking policy, depending on whether we aim to train the root-based mode or pose-based mode controller.

3.3.2 Supervised Initialization

We initially attempted to continue training WBC with RL from the base policy, but encountered two problems. The first one is that the upper-body controller is prone to being overly conservative, failing to accurately track the reference pose. The other one is that the unconvergent upper-body controller disrupted the training of the lower-body controller, causing it to deviate from the base policy and compromising its ability to maintain balance. To address these, we propose a supervised initialization process. Firstly, supervised learning is applied to the upper body to directly mimic the reference pose: $L_{\text{SL}}(\pi^{\text{upper}}) = \|\pi^{\text{upper}}(\boldsymbol{x}_{t,3}) - \boldsymbol{g}_t^{\text{upper}}\|$. Meanwhile, we impose a KL regularization on the lower-body controller to prevent a large deviation from the base policy: $L_{\text{KL}}(\pi^{\text{lower}}) = D_{\text{KL}}\left[\pi^{\text{lower}}(\cdot \mid \boldsymbol{x}_{t,15}) \| \pi^{\text{base}}(\cdot \mid \boldsymbol{x}_{t,15})\right]$. While these two objectives may result in a whole-body controller with suboptimal performance, they provide a solid initialization, facilitating faster learning during subsequent RL training and reducing the risk of converging to overly conservative local optima. In addition, to further simplify the task, we reduce the pose amplitude to half of its original value.

3.3.3 RL Training

Based on the initialized policy introduced in the previous section, we further improve its performance with RL. At this stage, we adjust the pose magnitude back to its original range. We use the PPO

Our reward function, as listed in Table 1, is carefully designed to improve both the performance and real-world applicability of the humanoid's motion. It primarily includes rewards for tracking the root's velocity and orientation, along with precise joint angle tracking. Additionally, we introduce several regularization terms to enhance the robot's stability and improve the transferability from simulation to real-world applications. The key components of our tracking reward are outlined in Table 1.

Table 1: Rewards Specification for JAEGER.

Term	Expression	Weight
	Upper Body	
DoF Position	$\exp(-4.0 * \lvert \boldsymbol{q}_{\text{ref}} - \boldsymbol{q} \rvert)$	20.0
	Lower Body	
DoF Position	$\exp(-4.0 * \lvert \boldsymbol{q}_{\text{ref}} - \boldsymbol{q} \rvert)$	100.0
Feet Height	$\exp(-10.0 * \lvert \boldsymbol{h}_{\text{target}} - \boldsymbol{h} \rvert)$	100.0
Linear Velocity	$\exp(-4.0 * \lvert \boldsymbol{v}_{\text{lin_cmd}} - \boldsymbol{v}_{\text{lin}} \rvert)$	20.0
Angular Velocity	$\exp(-4.0 * \lvert \boldsymbol{v}_{\text{ang_cmd}} - \boldsymbol{v}_{\text{ang}} \rvert)$	20.0
Roll & Pitch	$\exp(-4.0 * \lvert \boldsymbol{\Omega}_{\text{ref}}^{\phi\theta} - \boldsymbol{\Omega}^{\phi\theta} \rvert)$	1.0

After training the models for both modes separately, we distill them into a single network via imitation learning. Note that, for the root-based mode, we only need to set the lower-body reference pose command to zero.

4 Experiment

4.1 Experimental Setup

We conduct our experiments in Isaac-Gym

To examine the effectiveness of our method, we evaluate three *open-sourced SOTA* baselines: **HumanPlus**

For the **root-based mode** evaluation, we use the upper body joint angles from the evaluation set $\hat{\mathcal{Q}}$ and randomly assign root velocity commands. We evaluate the performance of each method via linear velocity tracking error $E_{\text{root-lin-vel}}$ (m/s) and angular velocity tracking error $E_{\text{root-ang-vel}}$ (rad/s) for the lower body. For the upper-body pose tracking, we calculate the joint tracking error $E_{\text{upper-j}}$ (rad). Note that OmniH2O is excluded from this mode since it does not support root commands.

Table 2: Main results of JAEGER in root-based mode. The officially released ExBody checkpoint is used for comparison. For HumanPlus, we reproduced the results using the officially released code. For root tracking, we compare errors in tracking root linear and angular velocity. For the upper body, we compare errors in tracking joint angles.

		Root Tracking Error		Joint Angle Error
Method		$E_{\text{root-lin-vel}} \downarrow$	$E_{\text{root-ang-vel}} \downarrow$	$E_{\text{upper-j}} \downarrow$
H1	ExBody	0.2417 ± 0.0004	0.4176 ± 0.0002	0.1689 ± 0.0006
	HumanPlus	0.1696 ± 0.0013	0.3105 ± 0.0021	0.2187 ± 0.0041
	JAEGER (Ours)	0.1504 ± 0.0264	0.0937 ± 0.0142	0.1093 ± 0.0010
H1-2	HumanPlus	0.4026 ± 0.0008	0.6209 ± 0.0015	0.2547 ± 0.0009
	JAEGER (Ours)	0.2224 ± 0.0150	0.2184 ± 0.0062	0.0933 ± 0.0001

For the **pose-based mode** evaluation, we use the whole-body poses from the evaluation set $\hat{\mathcal{Q}}$ and evaluate the tracking ability of each method. Specifically, we compare the methods based on upper-body and lower-body joint tracking errors, $E_{\text{upper-j}}$ and $E_{\text{lower-j}}$ (rad), as well as root velocity tracking error $E_{\text{root-vel}}$ (m/s) and root orientation tracking error, $E_{\text{root-r}}$, $E_{\text{root-p}}$ and $E_{\text{root-y}}$ (rad). In this mode, the target root

Table 3: Main results of JAEGER in pose-based mode. For HumanPlus and OmniH2O, we reproduced the results using the officially released code. We calculate the upper-body joint tracking error $E_{\text{upper-j}}$ (rad), lower-body joints $E_{\text{lower-j}}$ (rad), root velocity tracking error $E_{\text{root-vel}}$ (m/s), and root orientation tracking error $E_{\text{root-rpy}}$ (rad). Mean absolute error (MAE) is utilized as the metric. JAEGER demonstrates significantly lower joint tracking errors than baselines.

		Root Tracking Error					Joint Angle Error	
Method		$E_{\text{root-vel}} \downarrow$	$E_{\text{root-height}} \downarrow$	$E_{\text{root-r}} \downarrow$	$E_{\text{root-p}} \downarrow$	$E_{\text{root-y}} \downarrow$	$E_{\text{upper-j}} \downarrow$	$E_{\text{lower-j}} \downarrow$
H1	HumanPlus	0.4220 ± 0.0047	0.1015 ± 0.0004	0.0766 ± 0.0006	0.1137 ± 0.0003	0.6147 ± 0.0092	0.2199 ± 0.0047	0.1284 ± 0.0002
	OmniH2O	0.5424 ± 0.0028	0.0853 ± 0.0022	0.0481 ± 0.0024	0.1301 ± 0.0041	0.6547 ± 0.0067	0.2217 ± 0.0024	0.2404 ± 0.0021
	JAEGER (Ours)	0.3181 ± 0.0009	0.1002 ± 0.0001	0.0415 ± 0.0001	0.0595 ± 0.0001	0.5452 ± 0.0028	0.0686 ± 0.0001	0.0658 ± 0.0001
H1-2	HumanPlus	0.3078 ± 0.0056	0.0673 ± 0.0003	0.0672 ± 0.0047	0.1270 ± 0.0012	0.8293 ± 0.0064	0.2277 ± 0.0004	0.1722 ± 0.0627
	OmniH2O	0.4509 ± 0.0029	0.0688 ± 0.0021	0.0314 ± 0.0024	0.0515 ± 0.0056	0.6540 ± 0.0141	0.2552 ± 0.0032	0.2790 ± 0.0078
	JAEGER (Ours)	0.3418 ± 0.0004	0.0608 ± 0.0000	0.0398 ± 0.0001	0.0931 ± 0.0001	0.8051 ± 0.0089	0.1192 ± 0.0002	0.1000 ± 0.0001

velocity and orientation are extracted from each trajectory in the evaluation set. In addition, since OmniH2O is the keypoint position tracking method, we convert its reference keypoint positions to corresponding joint angles to ensure a fair comparison on the same dimensional basis. Note that ExBody is excluded in this mode since it does not support lower-body pose tracking.

4.2 Main Experiments

4.2.1 Root-based mode

The evaluation results of the root-based mode are presented in Table 2. Specifically, we compare JAEGER with two SOTA baselines, HumanPlus and ExBody, that support root command and release the code. For the root velocity training and evaluation range, we set the linear forward velocity as $[-0.5, 1]$ m/s, linear sideways velocity as $[-0.5, 0.5]$ m/s, and the angular velocity as $[-0.5, 0.5]$ rad/s. Note that we randomly sample root commands from the evaluation range and the upper-body reference pose from the evaluation set.

For both H1 and H1-2, JAEGER significantly outperforms the baseline in both velocity tracking error and upper-body joint angles tracking error. In more detail, for H1, the average error of the upper body is approximately 0.1 radians (5.7 degrees), while the errors of the other methods are at least 10 degrees. For velocity tracking error, JAEGER also shows a significant improvement, especially in angular velocity tracking, with an error of only 5 degrees per second. HumanPlus, due to its single-network structure, exhibits noticeable deficiencies in both upper-body angle tracking and angular velocity tracking. Surprisingly, ExBody, using the official checkpoint, performs the worst. This may be because its training set consists of only a small subset of AMASS. However, both JAEGER and HumanPlus utilize the entire dataset, resulting in stronger generalization capabilities. For H1-2, we can draw similar experimental conclusions as H1. However, in velocity tracking, H1-2 performs worse than H1 due to

its heavier upper body (with heavier arms and computing modules), which increases the difficulty of lower-body training.

4.2.2 Pose-based mode

The evaluation results of the pose-based mode are presented in Table 3. Specifically, we compare JAEGER with two SOTA baselines, HumanPlus and OmniH2O, which support tracking reference pose.

For both H1 and H1-2, JAEGER achieves higher tracking accuracy in both upper-body and lower-body pose tracking compared to baselines, HumanPlus and OmniH2O. In detail, the joint angle tracking errors of JAEGER are only 30-60% of those observed in other methods. Meanwhile, JAEGER also demonstrates comparable accuracy in root tracking. For example, JAEGER exhibits the lowest tracking errors for root roll, pitch, and yaw on H1. This suggests that our method does not sacrifice other tracking capabilities to achieve precise joint angle tracking.

Consistent with the results in the root-based mode, we also observe that H1-2 exhibits larger tracking errors due to its heavier body across all methods. While both HumanPlus and JAEGER are designed to track joint angles, JAEGER achieves lower tracking errors across most evaluation metrics, suggesting the effectiveness of our proposed dual-level controller and curriculum learning process. Furthermore, JAEGER exhibits low tracking errors in tracking root rotation and velocity, demonstrating its ability to maintain balance effectively while achieving superior lower-body pose tracking performance compared to the baselines. OmniH2O demonstrates certain advantages in root tracking, as it is trained to follow global keypoint positions, whereas HumanPlus and JAEGER are designed to track joint angles. As a result, OmniH2O shows the worst tracking ability in joint angles.

4.2.3 Stability

We conducted stability tests by applying lateral forces roughly ranging from 70N to 210N (1–3m/s^2 equivalent acceleration). The robot consistently maintained balance or recovered after perturbations in over 90 % of trials. We conducted 5 hardware trials for each demo motion. The demo motions are in the supplementary material.

4.3 Component Ablations

We also do the ablation studies to evaluate the effectiveness of each component. In more detail, we compare JAEGER with two ablation baseline: the one without supervised initialization, denoted as **JAEGER w.o. Init**, and the one that uses one transformer network to control the whole body, denoted as **JAEGER single**.

We conduct ablation studies to evaluate the effectiveness of each component. Specifically, we compare JAEGER with two ablation baselines: one without supervised initialization, referred to as **JAEGER w.o. Init**, and another replacing the dual-level controller with a single transformer network to control the whole body, referred to as **JAEGER single**.

For the root-based mode, we found that while velocity tracking of JAEGER is slightly better than those of the two ablation methods overall, there is a significant

Table 4: Ablation studies of JAEGER in pose-based mode. JAEGER w.o. Init refers to the controller trained directly without supervised initialization. JAEGER Single refers to the controller using one transformer network to control the whole body.

		Root Tracking Error					Joint Angle Error	
Method		$E_{\text{root-vel}} \downarrow$	$E_{\text{root-height}} \downarrow$	$E_{\text{root-r}} \downarrow$	$E_{\text{root-p}} \downarrow$	$E_{\text{root-y}} \downarrow$	$E_{\text{upper-j}} \downarrow$	$E_{\text{lower-j}} \downarrow$
H1	JAEGER w.o. Init	0.3059 $\pm$ 0.0009	0.0982 $\pm$ 0.0000	0.0419 $\pm$ 0.0001	0.0549 $\pm$ 0.0001	0.5718 $\pm$ 0.0040	0.0780 $\pm$ 0.0001	0.0667 $\pm$ 0.0001
	JAEGER Single	0.3142 $\pm$ 0.0009	0.0958 $\pm$ 0.0001	0.0408 $\pm$ 0.0001	0.0550 $\pm$ 0.0001	0.5601 $\pm$ 0.0010	0.2469 $\pm$ 0.0001	0.0654 $\pm$ 0.0001
	JAEGER (Ours)	0.3181 $\pm$ 0.0009	0.1002 $\pm$ 0.0001	0.0415 $\pm$ 0.0001	0.0595 $\pm$ 0.0001	0.5452 $\pm$ 0.0028	0.0686 $\pm$ 0.0001	0.0658 $\pm$ 0.0001
H1-2	JAEGER w.o. Init	0.4108 $\pm$ 0.0037	0.0664 $\pm$ 0.0045	0.0577 $\pm$ 0.0005	0.0781 $\pm$ 0.0002	1.5397 $\pm$ 0.0089	0.4088 $\pm$ 0.0010	0.1137 $\pm$ 0.0004
	JAEGER Single	0.6634 $\pm$ 0.0028	0.0643 $\pm$ 0.0003	0.0755 $\pm$ 0.0002	0.0989 $\pm$ 0.0002	1.1690 $\pm$ 0.0099	0.2477 $\pm$ 0.0005	0.1112 $\pm$ 0.0003
	JAEGER (Ours)	0.3418 $\pm$ 0.0004	0.0608 $\pm$ 0.0000	0.0398 $\pm$ 0.0001	0.0931 $\pm$ 0.0001	0.8051 $\pm$ 0.0089	0.1192 $\pm$ 0.0002	0.1000 $\pm$ 0.0001

Table 6: The comparison of different retargeting methods in terms of real-time performance (s/frame) and retargeting error (m/frame).

Method	**Real-time**	**Error**
HumanPlus	0.00200	0.16908
H2O	9.05409	0.14476
JAEGER (Ours)	0.00064	0.13567

Table 7: The smoothness of the joint angle sequence is evaluated using the Root Mean Square of angular acceleration, the second difference of the angle (rad/s^2), where lower values indicate higher smoothness.

	Left Shoulder			Right Shoulder		
Method	**Pitch**	**Roll**	**Yaw**	**Pitch**	**Roll**	**Yaw**
H2O	0.0143	0.0105	0.0147	0.0100	0.0065	0.0118
JAEGER (Ours)	0.0069	0.0033	0.0052	0.0070	0.0031	0.0054

difference in upper-body tracking. As shown in Table 5, JAEGER performs better in upper body tracking. In contrast, JAEGER Single performs the worst in upper body tracking. We speculate that when a single policy controls the entire body, the upper and lower body movements may interfere with each other. Specifically, larger upper-body movements can increase the difficulty of maintaining lower-body stability. During training, the policy can easily sacrifice the performance of one task to satisfy the tracking ability of the other. For example, on H1-2, JAEGER Single demonstrates best performance on root tracking error but the worst joint angle tracking ability. This validates the advantage of the dual-level controller, where separate controllers can focus more on their respective tasks. In addition, JAEGER's better performance than JAEGER w.o. Init also highlights the importance of supervised initialization. A good initialization can influence the sub-optimality where the model can converge.

In the root-based mode, we found that while JAEGER slightly outperforms the two ablation methods in velocity tracking overall, there is a significant difference in upper-body tracking. As shown in Table 5, JAEGER performs the best in upper-body tracking, whereas JAEGER Single performs the worst.

Table 5: Ablation studies of JAEGER in root-based mode.

		Root Tracking Error		Joint Angle Error
Method		$E_{\text{root-lin-vel}} \downarrow$	$E_{\text{root-ang-vel}} \downarrow$	$E_{\text{upper-j}} \downarrow$
H1	JAEGER w.o. Init	0.1953 $\pm$ 0.0152	0.2015 $\pm$ 0.0075	0.1627 $\pm$ 0.0001
	JAEGER Single	0.2039 $\pm$ 0.0010	0.2776 $\pm$ 0.0402	0.5377 $\pm$ 0.0003
	JAEGER (Ours)	0.1504 $\pm$ 0.0264	0.0937 $\pm$ 0.0142	0.1093 $\pm$ 0.0010
H1-2	JAEGER w.o. Init	0.2488 $\pm$ 0.0236	0.2780 $\pm$ 0.0185	0.1286 $\pm$ 0.0091
	JAEGER Single	0.2026 $\pm$ 0.0273	0.1865 $\pm$ 0.0230	0.4693 $\pm$ 0.0070
	JAEGER (Ours)	0.2224 $\pm$ 0.0150	0.2184 $\pm$ 0.0062	0.0933 $\pm$ 0.0001

We speculate that when a single policy controls the entire body, upper-body and lower-body movements may interfere with each other. Specifically, larger upper-body movements can make it more difficult to maintain lower-body stability. During training, the policy may sacrifice the performance of one task to improve the tracking ability of the other. For example, on H1-2, JAEGER Single achieves the best performance in root tracking error but the worst in joint angle tracking. This highlights the advantage of the dual-level controller, where separate controllers can focus more effectively on their respective tasks. Additionally, JAEGER's superior performance compared to JAEGER w.o. Init emphasizes the importance of supervised initialization, as good initialization influences the model's ability to converge to an optimal solution.

As depicted in Table 4, we can observe a similar phenomenon for pose-based mode: the dual-level controller performs better than the single controller, especially for the upper-body joint angle tracking, and supervised initialization can help the model converge to a better performance. In the pose-based mode, the difference is not as significant as the root-based mode. The reason is that the velocity is generally lower, which results in less interference from the upper body with the lower-body's balance. In summary, these results again verify the effectiveness of each component.

As shown in Table 4, a similar trend is observed in the pose-based mode: the dual-level controller outperforms the single controller, especially in upper-body joint angle tracking, and supervised initialization improves model convergence and performance. Similar to the root-based mode, JAEGER Single struggles with upper-body joint tracking but performs well in lower-body joint tracking, maintaining low errors. Additionally, JAEGER w.o. Init exhibits significant failure in upper-body tracking for H1-2. This is due to poor initialization, which may cause some joints in the upper body to converge to a suboptimal solution, resulting in a higher average error. As for lower-body tracking, the three methods demonstrate comparable performance. Differences in root tracking errors among the three methods are less significant compared to the root-based mode. This can be attributed to the generally lower root velocity in the pose-based mode, which reduces the interference of upper-body movements with lower-body balance. In conclusion, these findings further confirm the effectiveness of each component.

4.4 Retargeting Analysis

In this part, we evaluate our retargeting method and baselines from the perspectives of real-time performance, accuracy, and smoothness. Since retargeting does not need to consider whether the pose is feasible for the humanoid, we use the entire AMASS dataset as the evaluation set. We conduct the following experiments on a platform equipped with an RTX 4060 GPU.

For real-time performance, each retargeting method processes motion sequences frame by frame, with the metric being the average time taken to process a single frame. The results are shown in Table 6. Our method significantly outperforms the others.

It turns out that the MLP enables control rates over 1000Hz, which is ideal for low-latency teleoperation applications, *e.g.,* surgical robotics. For accuracy, the targets are joint positions of the human original pose in terms of its shape. We compare the average error between the joint positions of the humanoid robot obtained by different methods and the targets. As shown in Table 6, our method achieves accuracy comparable to H2O and outperforms HumanPlus. The subpar performance of HumanPlus is due to directly copying the angles without considering the differences in DoF structures and geometric sizes between the humanoid and the human model.

We also want to emphasize that MLP-based retargeting method offers advantages over IK approaches in terms of global continuity and computational efficiency. We randomly select a sequence consisting of 2000 frames. This sequence is then separately processed through our MLP-based method and the IK method adopted in H2O for the retargeting task. Table 7 shows the smoothness of joint angles of both the left and right shoulders. Specifically, the smoothness is evaluated by the Root Mean Square of angular acceleration. The results verify the superior output smoothness achieved by our MLP-based method, which is essential for real-world deployment.

5 Conclusion

In this paper, we present a novel dual-level humanoid whole-body controller that combines both root velocity tracking and local joint angle tracking. By decoupling the upper and lower bodies into separate controllers, we reduce interference and improve the efficiency of training. Experimental results demonstrate that our method outperforms existing baselines in simulation. In addition, we also deploy our controller to real-world humanoid robot scenarios and show promising performance.

6 Declarations

6.1 Ethics, Consent to Participate, and Consent to Publish declarations

Not applicable.

6.2 Funding

This work was supported by the Beijing Academy of Artificial Intelligence (BAAI).

6.3 Competing Interests

The authors declare no competing interests.

References

[1] Agarwal A, Kumar A, Malik J, et al (2023) Legged locomotion in challenging terrains using egocentric vision. In: Conference on robot learning, PMLR, pp 403–415

[2] Cheng X, Ji Y, Chen J, et al (2024) Expressive whole-body control for humanoid robots

[3] Fu Z, Zhao Q, Wu Q, et al (2024) Humanplus: Humanoid shadowing and imitation from humans. In: Conference on Robot Learning (CoRL)

[4] Grizzle JW, Hurst J, Morris B, et al (2009) Mabel, a new robotic bipedal walker and runner. In: 2009 American Control Conference, IEEE, pp 2030–2036

[5] Gronauer S, Diepold K (2022) Multi-agent deep reinforcement learning: a survey. Artificial Intelligence Review 55(2):895–943

[6] He T, Luo Z, He X, et al (2024) Omnih2o: Universal and dexterous human-to-humanoid whole-body teleoperation and learning

[7] He T, Luo Z, Xiao W, et al (2024) Learning human-to-humanoid real-time whole-body teleoperation. arXiv preprint arXiv:240304436

[8] He T, Xiao W, Lin T, et al (2024) Hover: Versatile neural whole-body controller for humanoid robots. arXiv preprint arXiv:241021229

[9] Ho J, Ermon S (2016) Generative adversarial imitation learning. Advances in neural information processing systems 29

[10] Ji M, Peng X, Liu F, et al (2024) Exbody2: Advanced expressive humanoid whole-body control. arXiv preprint arXiv:241213196

[11] Li Z, Cheng X, Peng XB, et al (2021) Reinforcement learning for robust parameterized locomotion control of bipedal robots. In: 2021 IEEE International Conference on Robotics and Automation (ICRA), IEEE, pp 2811–2817

[12] Li Z, Peng XB, Abbeel P, et al (2024) Reinforcement learning for versatile, dynamic, and robust bipedal locomotion control. The International Journal of Robotics Research p 02783649241285161

[13] Long J, Ren J, Shi M, et al (2024) Learning humanoid locomotion with perceptive internal model. arXiv preprint arXiv:241114386

[14] Lowe R, Wu YI, Tamar A, et al (2017) Multi-agent actor-critic for mixed cooperative-competitive environments. Advances in neural information processing systems 30

[15] Mahmood N, Ghorbani N, Troje NF, et al (2019) AMASS: Archive of motion capture as surface shapes. In: International Conference on Computer Vision, pp 5442–5451

[16] Mayne DQ, Rawlings JB, Rao CV, et al (2000) Constrained model predictive control: Stability and optimality. Automatica 36(6):789–814

[17] Oliehoek FA, Amato C, et al (2016) A concise introduction to decentralized POMDPs, vol 1. Springer

[18] Parisotto E, Song F, Rae J, et al (2020) Stabilizing transformers for reinforcement learning. In: International conference on machine learning, PMLR, pp 7487–7498

[19] Radosavovic I, Xiao T, Zhang B, et al (2023) Real-world humanoid locomotion with reinforcement learning. URL https://arxiv.org/abs/2303.03381, arXiv:2303.03381

[20] Radosavovic I, Rajasegaran J, Shi B, et al (2024) Humanoid locomotion as next token prediction. In: The Thirty-eighth Annual Conference on Neural Information Processing Systems, URL https://openreview.net/forum?id=GrMczQGTlA

[21] Radosavovic I, Xiao T, Zhang B, et al (2024) Real-world humanoid locomotion with reinforcement learning. Science Robotics 9(89):eadi9579

[22] Rashid T, Samvelyan M, De Witt CS, et al (2020) Monotonic value function factorisation for deep multi-agent reinforcement learning. Journal of Machine Learning Research 21(178):1–51

[23] Siekmann J, Godse Y, Fern A, et al (2021) Sim-to-real learning of all common bipedal gaits via periodic reward composition. In: 2021 IEEE International Conference on Robotics and Automation (ICRA), IEEE, pp 7309–7315

[24] Yu C, Velu A, Vinitsky E, et al (2022) The surprising effectiveness of ppo in cooperative multi-agent games. Advances in Neural Information Processing Systems 35:24611–24624

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

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- 2.pdf