AMO: Adaptive Motion Optimization for Hyper-Dexterous Humanoid Whole-Body Control

Jialong Li* Xuxin Cheng* Tianshu Huang* Shiqi Yang Ri-Zhao Qiu Xiaolong Wang
UC San Diego
https://amo-humanoid.github.io/

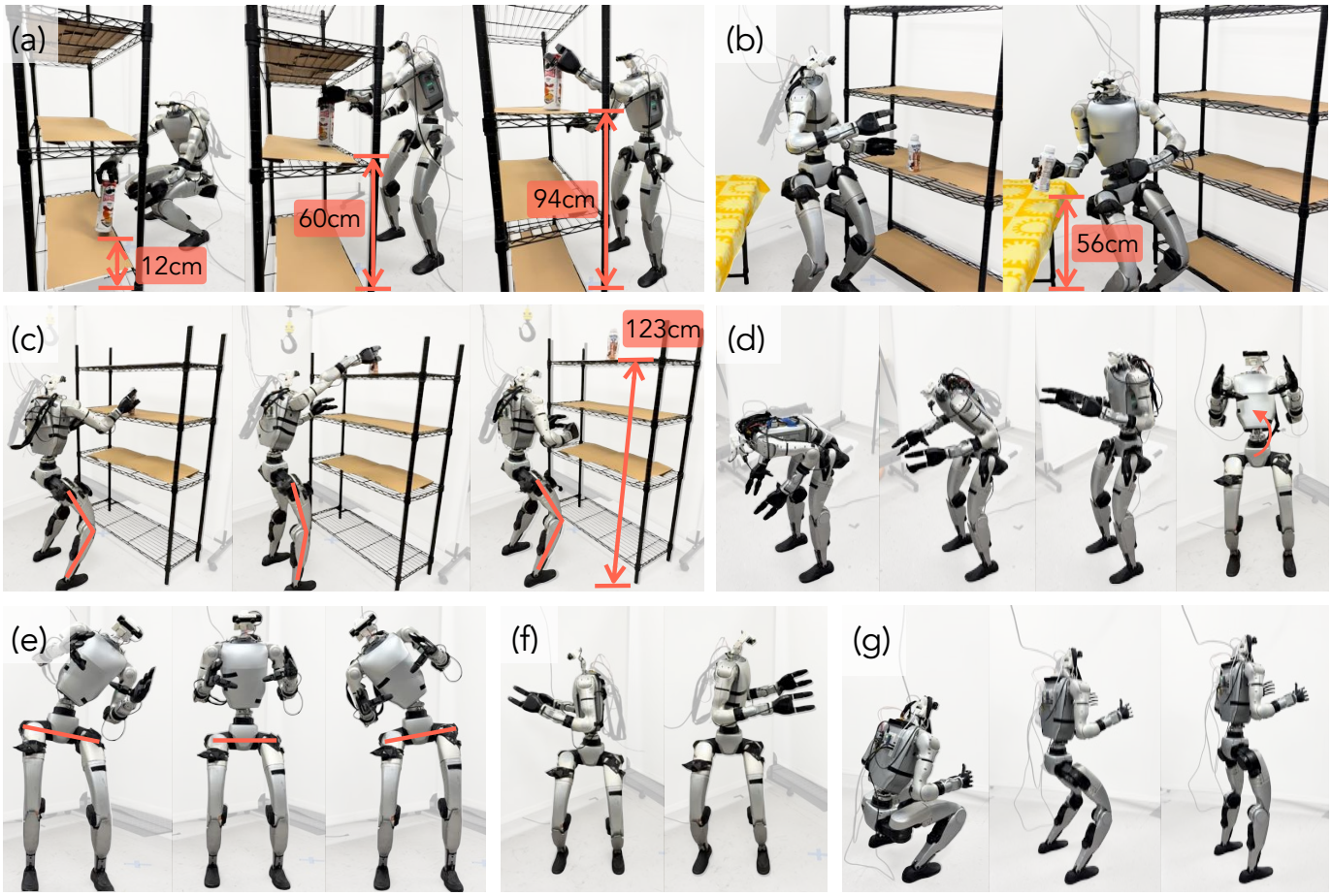


Fig. 1: AMO enables hyper-dexterous whole-body movements for humanoid robots. (a): The robot picks and places a can on platforms of different heights. (b): The robot picks a bottle from the higher shelf on the left and puts it on the lower table on the right. (c): The robot stretches its legs to put the bottle on a high shelf. (d-g): The robot demonstrates a wide range of torso pitch, roll, yaw, and height adjustments. (e): The robot utilizes hip motors to compensate waist joint limits to achieve larger roll rotation.

***Abstract*—Humanoid robots derive much of their dexterity from hyper-dexterous whole-body movements, enabling tasks that require a large operational workspace—such as picking objects off the ground. However, achieving these capabilities on real humanoids remains challenging due to their high degrees of freedom (DoF) and nonlinear dynamics. We propose Adaptive Motion Optimization (AMO), a framework that integrates sim-to-real reinforcement learning (RL) with trajectory optimization for real-time, adaptive whole-body control. To mitigate distribution bias in motion imitation RL, we construct a hybrid AMO dataset and train a network capable of robust, on-demand adaptation to potentially O.O.D. commands. We validate AMO in simulation and on a 29-DoF Unitree G1 humanoid robot, demonstrating superior stability and an expanded workspace compared to strong baselines. Finally, we show that AMO's consistent performance supports autonomous task execution via imitation learning, underscoring the system's versatility and robustness.**

* Equal contributions. Special thanks to Jiajian Fu for designing and manufacturing of the Active Head 2.0.

I. Introduction

Humans can expand their workspace of hands using whole-body movements. The joint configurations of humanoid robots closely mimic humans' functionality and degree of freedom while facing challenges of achieving similar movements with

Metrics	AMO (Ours)	HOVER [30]	Opt2Skill [41]
Ref. Type	Hybrid	MoCap	Traj. Opt.
SO(3) + Height Dex. Torso	✓	✗	✗
SE(3) Task Space	✓	✗	✗
No Ref. Deploy	✓	✓	✗
O.O.D.	✓	✗	✗

TABLE I: Comparisons with two recent representative humanoid motion imitation works. *Dex. Torso* means if the robot is able to adjust its' torso's orientation and height to expand the workspace. *Task Space* means the end effector. *No Ref. Deploy* means if the robot needs reference motion during deployment. *O.O.D.* means if the work has evaluated O.O.D. performance, which is a typical case when the robot is controlled by a human operator and the control signal is highly unpredictable.

real-time control. This is due to the dynamic humanoid whole-body control's high-dimensional, highly non-linear, and contact-rich nature. Traditional model-based optimal control methods require precise modeling of the robot and environment, high computational power, and reduced-order models for realizable computation results, which is not feasible for the problem of utilizing all DoFs (29) of an overactuated humanoid robot in the real world.

Recent advances in reinforcement learning (RL) combined with sim-to-real have demonstrated significant potential in enabling humanoid loco-manipulation tasks in real-world settings [42]. While these approaches achieve robust real-time control for high-degree-of-freedom (DoF) humanoid robots, they often rely on extensive human expertise and manual tuning of reward functions to ensure stability and performance. To address this limitation, researchers have integrated motion imitation frameworks with RL, leveraging retargeted human motion capture (MoCap) trajectories to define reward objectives that guide policy learning [10, 28]. However, such trajectories are typically kinematically viable but fail to account for the dynamic constraints of target humanoid platforms, introducing an embodiment gap between simulated motions and hardware-executable behaviors. An alternative approach combines trajectory optimization (TO) with RL to bridge this gap [38, 41].

While these methodologies advance humanoid loco-manipulation capabilities, current approaches remain constrained to simplified locomotion patterns rather than achieving true whole-body dexterity. Motion capture-driven methods suffer from inherent kinematic bias: their reference datasets predominantly feature bipedal locomotion sequences (e.g., walking, turning) while lacking coordinated arm-torso movements essential for hyper-dexterous manipulation. Conversely, trajectory optimization TO-based techniques face complementary limitations—their reliance on a limited repertoire of motion primitives and computational inefficiency in real-time applications precludes policy generalization. This critically hinders deployment in dynamic scenarios requiring rapid adaptation to unstructured inputs, such as reactive teleoperation or environmental perturbations.

To bridge this gap, we present Adaptive Motion Optimization (AMO)—a hierarchical framework for real-time whole-body control of humanoid robots through two synergistic innovations: (i) Hybrid Motion Synthesis: We formulate hybrid upper-body command sets by fusing arm trajectories from motion capture data with probabilistically sampled torso orientations, systematically eliminating kinematic bias in the training distribution. These commands drive a dynamics-aware trajectory optimizer to produce whole-body reference motions that satisfy both kinematic feasibility and dynamical constraints, thereby constructing the AMO dataset—the first humanoid motion repository explicitly designed for dexterous loco-manipulation. (ii) Generalizable Policy Training: While a straightforward solution would map commands to motions via discrete look-up tables, such methods remain fundamentally constrained to discrete, in-distribution scenarios. Our AMO network instead learns continuous mappings, enabling robust interpolation across both continuous input spaces and out-of-distribution (O.O.D.) teleoperation commands while maintaining real-time responsiveness.

During deployment we first extract the sparse poses from a VR teleoperation system and output upper-body goals with multi-target inverse kinematics. The trained AMO network and RL policy together output the robot's control signal. We list a brief comparison in Table. I to show the major advantages of our method with two recent representative works. To summarize, our contributions are as follows:

- A novel adaptive control method AMO that substantially expands the workspace of humanoid robots. AMO works in real-time with sparse task-space targets and shows O.O.D. performance previous methods have not achieved.
- A new landmark of humanoid hyper-dexterous WBC controller with orders larger workspace that enables a humanoid robot to pick up objects from the ground.
- Comprehensive experiments both in simulation and the real world with teleoperation and autonomous results showing the effectiveness of our method and ablations of core components.

II. Related Work

Humanoid Whole-body Control. Whole-body control for humanoid robots remains a challenging problem due to their high DoFs and non-linearity. This is previously primarily achieved by dynamic modeling and model-based control [14, 16, 18, 19, 31, 32, 34, 35, 51, 52, 56, 72, 75]. More recently, deep reinforcement learning methods have shown promise in achieving robust locomotion performance for legged robots [3, 7, 8, 20, 21, 22, 23, 26, 37, 38, 39, 40, 46, 47, 61, 67, 73, 74, 79]. Researchers have studied whole-body control from high-dimensional inputs for quadrupeds [7, 8, 22, 27] and humanoids [9, 24, 29, 33]. [24] trains one transformer for control and another transformer for imitation learning. [9] only encourages the upper body to imitate the motion, while the lower body control is decoupled. [29] trains goal-conditioned policies for downstream tasks. All [9, 24, 29] show only limited whole-body control ability, which enforcing

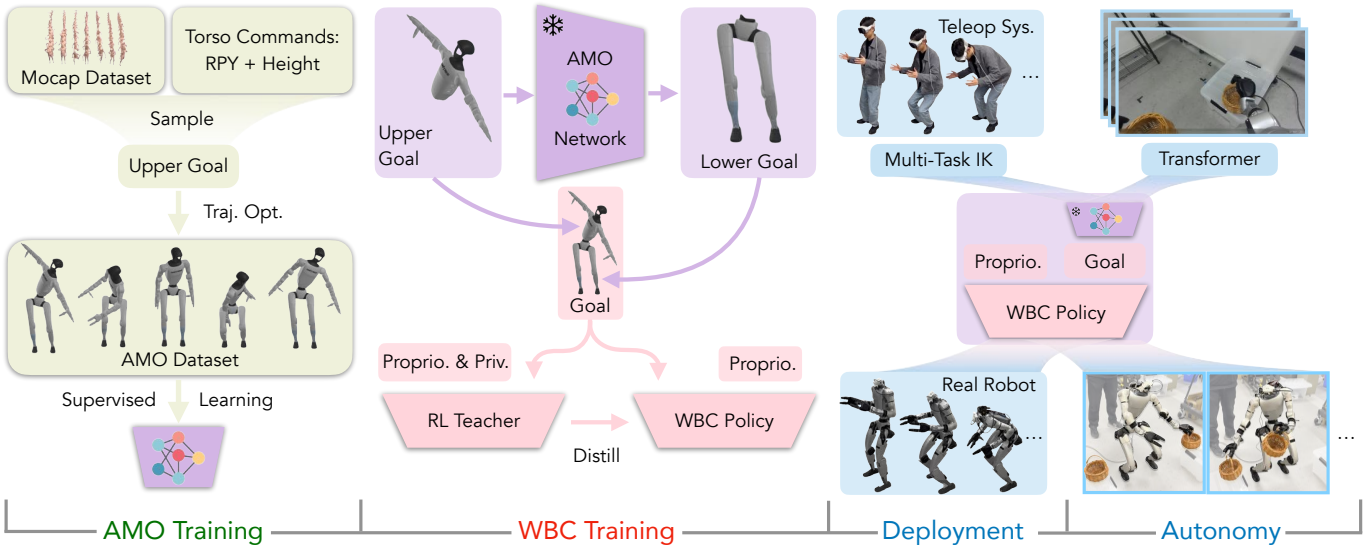


Fig. 2: System overview. The system is decomposed into four stages: 1. AMO module training by collecting AMO dataset using trajectory optimization; 2. RL policy training by teacher-student distillation in simulation; 3. real robot teleoperation by IK and retargeting; 4. real robot autonomous policy training by imitation learning (IL) with a transformer.

the torso and pelvis of humanoid robots to stay motionless. [33] shows expressive whole-body action for humanoid robots, but it does not emphasize leveraging whole-body control to extend robots' loco-manipulation task space.

Teleoperation of Humanoids. Teleoperation of humanoids is crucial for real-time control and robot data collection. Prior efforts in humanoid teleoperation include [11, 24, 28, 29, 42, 64]. For example, [24, 29] uses a third-person RGB camera to obtain key points of the human teleoperator. Some works use VR to provide ego-centric observation for teleoperators. [11] uses Apple VisionPro to control the active head and upper body with dexterous hands. [42] uses Vision Pro for the head and upper body while using pedals for locomotion control. Humanoid whole-body control requires a teleoperator to provide physically achievable whole-body coordinates for the robots.

Loco-manipulation Imitation Learning. Imitation learning has been studied to help the robot complete the task autonomously. Identifying existing work with demonstration sources can be classified as learning from real robot expert data [4, 5, 12, 13, 25, 36, 54, 55, 65, 66, 76, 78], learning from play data [15, 50, 70], and learning from human demonstrations [9, 21, 24, 26, 29, 57, 58, 59, 69, 71, 73]. Those imitation learning studies are limited to manipulation skills, while there are very few imitation learning studies for loco-manipulation. [25] studies imitation learning for loco-manipulation, using a wheel-based robot. This paper uses imitation learning to enable humanoid robots to complete loco-manipulation tasks autonomously.

III. Adaptive Motion Optimization

We present AMO: Adaptive Motion Optimization, a framework that achieves seamless whole-body control as illustrated in Figure 2. Our system is decomposed into four different components. We introduce the notations and overall framework at first, and then elaborate on the learning and realization of these components in the following sections separately.

A. Problem Formulation and Notations

We address the problem of humanoid whole-body control and focus on two distinct settings: **teleoperation** and **autonomous**.

In the teleoperation setting, the whole-body control problem is formulated as learning a goal-conditioned policy $\pi' : \mathcal{G} \times \mathcal{S} \to \mathcal{A}$, where $\mathcal{G}$ represents the goal space, $\mathcal{S}$ the observation space, and $\mathcal{A}$ the action space. In the autonomous setting, the learned policy $\pi : \mathcal{S} \to \mathcal{A}$ generates actions solely based on observations, without human input.

The goal-conditioned teleoperation policy receives a control signal $\mathbf{g} \in \mathcal{G}$ from the teleoperator, where $\mathbf{g} = [\mathbf{p}_{\text{head}}, \mathbf{p}_{\text{left}}, \mathbf{p}_{\text{right}}, \mathbf{v}]$. $\mathbf{p}_{\text{head}}, \mathbf{p}_{\text{left}}, \mathbf{p}_{\text{right}}$ represent the operator's head and hand keypoint poses, while $\mathbf{v} = [v_x, v_y, v_{\text{yaw}}]$ specifies the base velocity. Observations $\mathbf{s} \in \mathcal{S}$ include visual and proprioceptive data: $\mathbf{s} = [\text{img}_{\text{left}}, \text{img}_{\text{right}}, \mathbf{s}_{\text{proprio}}]$. Actions $\mathbf{a} \in \mathcal{A}$ consist of joint angle commands for the upper and lower body: $\mathbf{a} = [\mathbf{q}_{\text{upper}}, \mathbf{q}_{\text{lower}}]$.

a) Goal-conditioned teleoperation policy: The goal-conditioned policy adopts a hierarchical design: $\pi' = [\pi'_{\text{upper}}, \pi'_{\text{lower}}]$. The upper policy $\pi'_{\text{upper}}(\mathbf{p}_{\text{head}}, \mathbf{p}_{\text{left}}, \mathbf{p}_{\text{right}}) = [\mathbf{q}_{\text{upper}}, \mathbf{g}']$ outputs actions for the upper body and an intermediate control signal $\mathbf{g}' = [\mathbf{rpy}, h]$, where $\mathbf{rpy}$ command the torso orientation and h commands the base height. The lower policy $\pi'_{\text{lower}}(\mathbf{v}, \mathbf{g}', \mathbf{s}_{\text{proprio}}) = \mathbf{q}_{\text{lower}}$ generates actions for the lower body using this intermediate control signal, the velocity commands, and proprioceptive observations.

b) Autonomous policy: The autonomous policy $\pi = [\pi_{\text{upper}}, \pi_{\text{lower}}]$ shares the same hierarchical design as the teleoperation policy. The lower policy is identical: $\pi_{\text{lower}} = \pi'_{\text{lower}}$, while the upper policy generates actions and intermediate control independently from human input: $\pi_{\text{upper}}(\text{img}_{\text{left}}, \text{img}_{\text{right}}, \mathbf{s}_{\text{proprio}}) = [\mathbf{q}_{\text{upper}}, \mathbf{v}, \mathbf{g}']$.

B. Adaptation Module Pre-Training

In our system specification, the lower policy follows commands in the form of $[v_x, v_y, v_{\text{yaw}}, \mathbf{rpy}, h]$. The locomotion ability of following velocity commands $[v_x, v_y, v_{\text{yaw}}]$ can be easily learned by randomly sampling directed vectors in the simulation environment following the same strategy as [8, 9]. However, it is non-trivial to learn the torso and height tracking skills as they require whole-body coordination. Unlike in the locomotion task, where we can design feet-tracking rewards based on Raibert Heuristic [62] to facilitate skill learning, there lacks such a heuristic to guide the robot to complete whole-body control. Some works [28, 33] train such policies by tracking human references. However, their strategy does not build a connection between human poses and whole-body control directives.

To address this issue, we propose an **adaptive motion optimization (AMO)** module. The AMO module is represented as $\phi(\mathbf{q}_{\text{upper}}, \mathbf{rpy}, h) = \mathbf{q}^{\text{ref}}_{\text{lower}}$. Upon receiving whole-body control commands $\mathbf{rpy}, h$ from the upper, it converts these commands into joint angle references for all lower-body actuators for the lower policy to track explicitly. To train this adaptive module, first, we collect an AMO Dataset by randomly sampling upper commands and performing model-based trajectory optimization to acquire lower body joint angles. The trajectory optimization can be formulated as a multi-contact optimal control problem (MCOP) with the following cost fuction:

$$
\begin{aligned}
\mathcal{L} &= \mathcal{L}_{\mathbf{x}} + \mathcal{L}_{\mathbf{u}} + \mathcal{L}_{\text{CoM}} + \mathcal{L}_{\mathbf{rpy}} + \mathcal{L}_h \\
\mathcal{L}_{\mathbf{x}} &= \|\mathbf{x}_t - \mathbf{x}_{\text{ref}}\|^2_{\mathbf{Q}_x} \\
\mathcal{L}_{\mathbf{u}} &= \|\mathbf{u}_t\|^2_{\mathbf{R}} \\
\mathcal{L}_{\text{CoM}} &= \|\mathbf{c}_t - \mathbf{c}_{\text{ref}}\|^2_{\mathbf{Q}_{\text{CoM}}} \\
\mathcal{L}_{\mathbf{rpy}} &= \|\mathbf{R}_{\text{torso}} - \mathbf{R}_{\text{ref}}(\mathbf{rpy})\|^2_{\mathbf{Q}_{\text{torso}}} \\
\mathcal{L}_h &= w_h(h_t - h)^2
\end{aligned}
$$

Which includes regularization for both state $\mathbf{x}$ and control $\mathbf{u}$, target tracking terms $\mathcal{L}_{\mathbf{rpy}}$ and $\mathcal{L}_h$, and a center-of-mass (CoM) regularization term that ensures balance when performing whole-body control. Upon collecting dataset, we first randomly select upper body motions from the AMASS dataset [43] and sample random torso commands. Then, we perform trajectory optimizations to track torso objectives while maintaining a stable CoM and adhering to wrench cone constraints to generate dynamically feasible reference joint angles. Since we do not take the walking scenario into consideration, both feet of the robot are considered to be making contact with the ground. The references are generated using control-limited feasibility-driven differential dynamic programming (BoxFDDP) through Crocoddyl [48, 49]. These data are collected to train an AMO module that converts torso commands into reference lower poses. The AMO module is a three-layer multi-layer perceptron (MLP) and is frozen during the later stage of lower policy training.

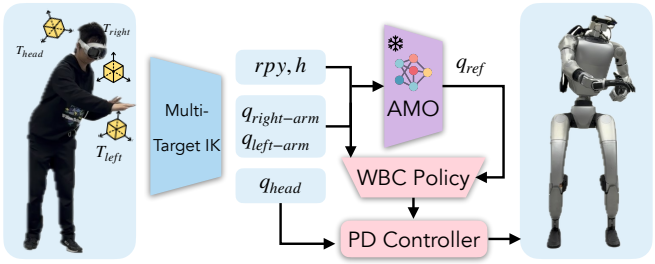


Fig. 3: Teleoperation system overview. The operator provides three end-effector targets: head, left wrist, and right wrist poses. A multi-target IK computes upper goals and intermediate goals by matching three weighted targets simultaneously. The intermediate goals $(\mathbf{rpy}, h)$ are fed to AMO and converted to lower goals.

C. Lower Policy Training

We use massively parallel simulation to train our lower policy with IsaacGym[44]. The lower policy aims at tracking $\mathbf{g}'$ and $\mathbf{v}$, while utilizing proprioceptive observations $\mathbf{s}_{\text{proprio}}$, which is defined as:

$$
[\boldsymbol{\theta}_t, \boldsymbol{\omega}_t, \mathbf{q}_t^{\text{whole-body}}, \dot{\mathbf{q}}_t^{\text{whole-body}}, \mathbf{a}_{t-1}^{\text{whole-body}}, \boldsymbol{\phi}_t, \mathbf{q}^{\text{ref}}_{\text{lower}}]
$$

The above formulation contains base orientation $\boldsymbol{\theta}_t$, base angular velocity $\boldsymbol{\omega}_t$, current position, velocity, and last position targets. It is noticeable that the lower policy's observation includes the states of upper body actuators for better upper-lower body coordination. $\boldsymbol{\phi}_t$ is the gait cycling signal defined in similar ways as [45, 77]. $\mathbf{q}^{\text{ref}}_{\text{lower}}$ is the reference lower body joint angles generated by the AMO module. The lower action space $\mathbf{q}_{\text{lower}} \in \mathbb{R}^{15}$ is a vector of dimension 15, which consists of $2 * 6$ target joint positions for both legs and 3 target joint positions for the waist motors.

We opt to use a teacher-student framework to train our lower policy. We first train a teacher policy that can observe privileged information in simulation, using off-the-shelf PPO[63]. We then distill the teacher policy into a student policy using supervised learning. The student policy only observes information available in real and can be deployed for teleoperation and autonomous tasks.

The teacher policy can be formulated as $\pi_{\text{teacher}}(\mathbf{v}, \mathbf{g}', \mathbf{s}_{\text{proprio}}, \mathbf{s}_{\text{priv}}) = \mathbf{q}_{\text{lower}}$. The additional privilege observation $\mathbf{s}_{\text{priv}}$ is defined as:

$$
[\mathbf{v}_t^{\text{gt}}, \mathbf{rpy}_t^{\text{gt}}, h_t^{\text{gt}}, \mathbf{c}_t]
$$

Which includes ground-truth values of: base velocity $\mathbf{v}_t^{\text{gt}}$, torso orientation $\mathbf{rpy}_t^{\text{gt}}$, and base height h_t^{gt}, which are not readily available when tracking their corresponding targets

in the real world. c_t is the contact signal between feet and the ground. The teacher RL training process is detailed in Appendix.B.

The student policy can be written as $\pi_{\text{student}}(\mathbf{v}, \mathbf{g}', \mathbf{s}_{\text{proprio}}, \mathbf{s}_{\text{hist}}) = \mathbf{q}_{\text{lower}}$. To compensate for $\mathbf{s}_{\text{proprio}}$ with observations accessible in the real-world, the student policy utilizes a 25-step history of proprioceptive observations as an additional input signal: $\mathbf{s}_{\text{hist},t} = \mathbf{s}_{\text{proprio},t-1\sim t-25}$.

D. Teleoperation Upper Policy Implementation

The teleoperation upper policy generates a series of commands for whole-body control, including arm and hand movements, torso orientation, and base height. We opt to implement this policy using optimization-based techniques to achieve the precision required for manipulation tasks. Specifically, hand movements are generated via retargeting, while other control signals are computed using inverse kinematics (IK). Our implementation of hand retargeting is based on dex-retargeting [60]. More details about our retargeting formulation are presented in Appendix.A.

In our whole body control framework, we extend the conventional IK to multi-target weighted IK, minimizing the 6D distances to three key targets: the head, the left wrist, and the right wrist. The robot mobilizes all upper-body actuators to match these three targets simultaneously. Formally, our objective is:

$$
\begin{aligned}
&\min_{\mathbf{q}} \quad \mathcal{L}_{\text{head}} + \mathcal{L}_{\text{left}} + \mathcal{L}_{\text{right}} \\
\mathcal{L}_{\text{head}} &= \|\mathbf{p}_{\text{head}} - \mathbf{p}_{\text{head-link}}\|^2 + \lambda \|\mathbf{R}_{\text{head}} - \mathbf{R}_{\text{head-link}}\|_F^2 \\
\mathcal{L}_{\text{left}} &= \|\mathbf{p}_{\text{left}} - \mathbf{p}_{\text{left-link}}\|^2 + \lambda \|\mathbf{R}_{\text{left}} - \mathbf{R}_{\text{left-link}}\|_F^2 \\
\mathcal{L}_{\text{right}} &= \|\mathbf{p}_{\text{right}} - \mathbf{p}_{\text{right-link}}\|^2 + \lambda \|\mathbf{R}_{\text{right}} - \mathbf{R}_{\text{right-link}}\|_F^2 \\
\mathbf{q} &= [\mathbf{q}_{\text{head}}, \mathbf{q}_{\text{left-arm}}, \mathbf{q}_{\text{right-arm}}, \mathbf{rpy}, h]
\end{aligned}
$$

As illustrated in Fig. 3. The optimization variable $\mathbf{q}$ includes all actuated degrees of freedom (DoFs) in the robot's upper body: $\mathbf{q}_{\text{head}}$, $\mathbf{q}_{\text{left-arm}}$, and $\mathbf{q}_{\text{right-arm}}$. In addition to the motor commands, it also solves for an intermediate command to enable whole-body coordination: $\mathbf{rpy}$ and h for torso orientation and height control. To ensure smooth upper body control, posture costs are weighted differently across different components of the optimization variable $\mathbf{q}$: $\mathbf{W}_{\mathbf{q}_{\text{head}}, \mathbf{q}_{\text{left-arm}}, \mathbf{q}_{\text{right-arm}}} < \mathbf{W}_{\mathbf{rpy}, h}$. This encourages the policy to prioritize upper-body actuators for simpler tasks. However, for tasks requiring whole-body movement, such as bending over to pick up or reaching distant targets, additional control signals $[\mathbf{rpy}, h]$ are generated and sent to the lower policy. The lower policy coordinates its motor angles to fulfill the upper policy's requirements, enabling whole-body target reaching. Our IK implementation employs Levenberg-Marquardt (LM) algorithm [68] and is based on Pink [6].

E. Autonomous Upper Policy Training

We learn the autonomous upper policy via imitation learning. First, the human operators teleoperate with the robot using our goal-conditioned policy, recording observations and actions as demonstrations. We then employ ACT [78] with a DinoV2 [17, 53] visual encoder as the policy backbone. The visual observation includes two stereo images img_{left} and $\text{img}_{\text{right}}$. DinoV2 divides each image into 16×22 patches and produces a 384-dimensional visual token for each patch, yielding a combined visual token of shape $2 \times 16 \times 22 \times 384$. This visual token is concatenated with a state token obtained by projecting $\mathbf{o}_t = [\mathbf{s}^{\text{upper}}_{\text{proprio},t}, \mathbf{v}_{t-1}, \mathbf{rpy}_{t-1}, h_{t-1}]$. Here, $\mathbf{s}^{\text{upper}}_{\text{proprio},t}$ is the upper-body proprioceptive observation, and $[\mathbf{v}_{t-1}, \mathbf{rpy}_{t-1}, h_{t-1}]$ constitutes the last command sent to the lower policy. Due to our decoupled system design, the upper policy observes these lower-policy commands instead of direct lower-body proprioception. The policy's output is represented as:

$$[\mathbf{q}_t^{\text{head}}, \mathbf{q}_t^{\text{dual-arm}}, \mathbf{q}_t^{\text{dual-hand}}, \mathbf{v}_t, \mathbf{rpy}_t, h_t]$$

comprising all upper-body joint angles and the intermediate control signals for the lower policy.

Metrics (↓)	E_y	E_p	E_r	E_h	$E_{\mathbf{v}}$
Stand					
Ours(AMO)	0.1355	**0.1259**	**0.0675**	**0.0151**	**0.0465**
w/o AMO	0.3137	0.1875	0.0681	0.1168	0.0477
w/o priv	0.1178	0.1841	0.0757	0.0228	0.1771
w rand arms	**0.1029**	0.1662	0.0716	0.0200	0.1392
Walk					
Ours(AMO)	0.1540	**0.1519**	**0.0735**	**0.0182**	0.1779
w/o AMO	0.3200	0.1927	0.0797	0.1253	**0.1539**
w/o priv	0.1226	0.1879	0.0779	0.0276	0.2616
w rand arms	**0.1200**	0.1837	0.0790	0.0240	0.2596

TABLE II: Comparison of tracking errors with baselines. Each tracking error is averaged over 4096 environments and 500 steps. E_y, E_p, E_r, E_h, $E_{\mathbf{v}}$ represent tracking errors in torso yaw, torso pitch, torso roll, base height, and base linear velocity, respectively.

IV. Evaluation

In this section, we aim to address the following questions by conducting experiments in both simulation and in the real world:

- How well does AMO performs on tracking locomotion commands $\mathbf{v}$ and torso commands $\mathbf{rpy}, h$?
- How does AMO compare to other WBC strategies?
- How well does the AMO system perform in the real-world setting?

We conduct our sim experiments in IsaacGym simulator [44]. Our real robot setup is as shown in 3, which is modified from Unitree G1 [1] with two Dex3-1 dexterous hands. This platform features 29 whole-body DoFs and 7 DoFs on each hand. We customized an active head with three actuated DoFs to map the human operator's head movement and mounted a ZED Mini [2] camera for stereo streaming.

Metrics	R_y	R_p	R_r
Ours(AMO)	(**-1.5512**, **1.5868**)	(-0.4546, **1.5745**)	(**-0.4723**, **0.4714**)
w/o AMO	(-0.9074, 0.8956)	(**-0.4721**, 1.0871)	(-0.3921, 0.3626)
Waist Tracking	(-1.5274, 1.5745)	(-0.4531, 0.5200)	(-0.3571, 0.3068)
ExBody2	(-0.1005, 0.0056)	(-0.2123, 0.4637)	(-0.0673, 0.0623)

TABLE III: Comparison of maximum torso control ranges. R_y, R_p, R_r is the maximum range the torso can achieve while following in-distribution (I.D.) tracking commands in yaw, pitch, roll directions.

A. How well does AMO perform on tracking locomotion commands $\mathbf{v}$ *and torso commands* $\mathbf{rpy}, h$*?*

Table II presents the evaluation of AMO's performance by comparing it against the following baselines:

- **w/o AMO:** This baseline follows the same RL training recipe as **Ours(AMO)**, with two key modifications. First, it excludes the AMO output $\mathbf{q}^{\text{ref}}_{\text{lower}}$ from the observation space. Second, instead of penalizing deviations from $\mathbf{q}^{\text{ref}}_{\text{lower}}$, it applies a regularization penalty based on deviations from a default stance pose.
- **w/o priv:** This baseline is trained without additional privileged observations $\mathbf{s}_{\text{priv}}$.
- **w rand arms:** In this baseline, arm joint angles are not set using human references sampled from a MoCap dataset. Instead, they are randomly assigned by uniformly sampling values within their respective joint limits.

The performance is evaluated using the following metrics:

1) **Torso Orientation Tracking Accuracy:** Torso orientation tracking is measured by E_y, E_p, E_r. The results indicate that *AMO* achieves superior tracking accuracy in roll and pitch directions. The most notable improvement is in pitch tracking, where other baselines struggle to maintain accuracy, whereas our model significantly reduces tracking error. *w rand arms* exhibits the lowest yaw tracking error, potentially because random arm movements enable the robot to explore a broader range of postures. However, *AMO* is not necessarily expected to excel in yaw tracking, as torso yaw rotation induces minimal CoM displacement compared to roll and pitch. Consequently, yaw tracking accuracy may not fully capture AMO's capacity for generating adaptive and stable poses. Nonetheless, it is worth noting that *w/o AMO* struggles with yaw tracking, suggesting that AMO provides critical reference information for achieving stable yaw control.
2) **Height Tracking Accuracy:** The results show that *AMO* achieves the lowest height tracking error. Notably, *w/o AMO* reports a significantly higher error than all other baselines, indicating that it barely tracks height command. Unlike torso tracking, where at least one waist motor angle is directly proportional to the command, height tracking requires coordinated adjustments across multiple lower-body joints. Without reference information from AMO, the policy fails to learn the transformation relationship between height commands and corresponding motor angles, making it difficult to master this skill.

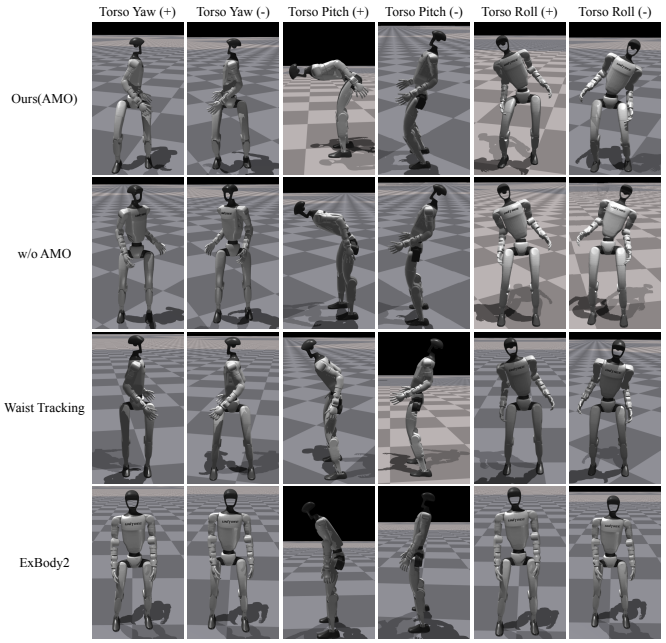


Fig. 4: Comparison of torso orientation ranges.

3) **Linear Velocity Tracking Accuracy:** The AMO module generates reference poses based on whole-body control in a double-support stance, meaning it does not account for pose variations due to foot swing during locomotion. Despite this limitation, *AMO* remains capable of performing stable locomotion with a low tracking error, demonstrating its robustness.

B. How does AMO compare to other WBC strategies?

To address this question, we compare the range of torso control across the following baselines:

- **w/o AMO:** This baseline is the same as described in the previous section.
- **Waist Tracking:** This baseline explicitly commands the yaw, roll, and pitch angles of the waist motors instead of controlling the torso orientation.
- **ExBody2:** [33] is a representative work in leveraging human reference motions to guide robot whole-body control in RL. It achieves torso orientation control by modifying the reference joint angles of the waist.

Table. III presents the quantitative measurements of torso control ranges, while Fig. 4 illustrates the qualitative differences. For *ExBody2* and other methods that rely on human motion tracking, the range of torso control is inherently constrained by the diversity of the human motion dataset used for training. If a particular movement direction is underrepresented in the dataset, the learned policy struggles to generalize beyond those sparse examples. As shown in our results, *ExBody2* exhibits only minor control over torso pitch and is largely incapable of tracking torso yaw and roll movements.

Waist Tracking is fundamentally restricted by the robot's waist joint limits, as it relies exclusively on waist motors for

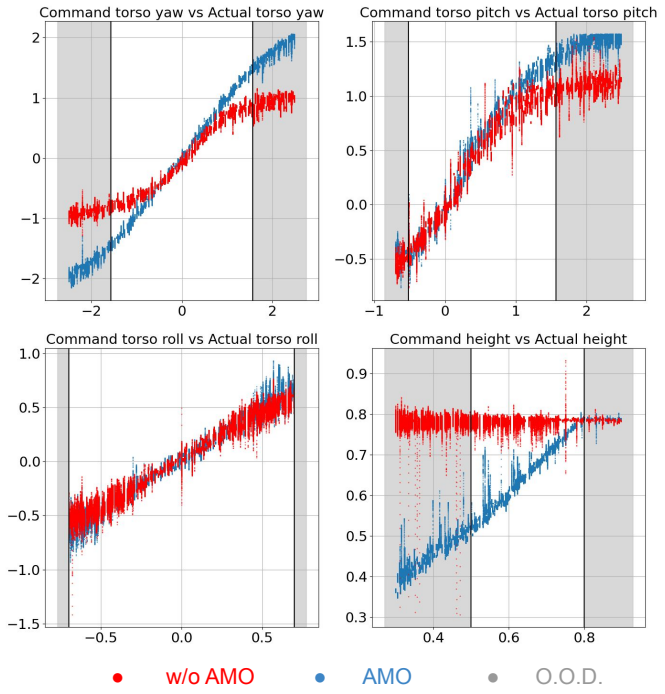
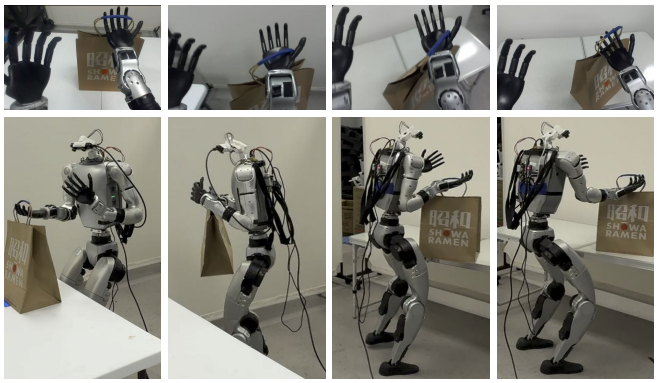


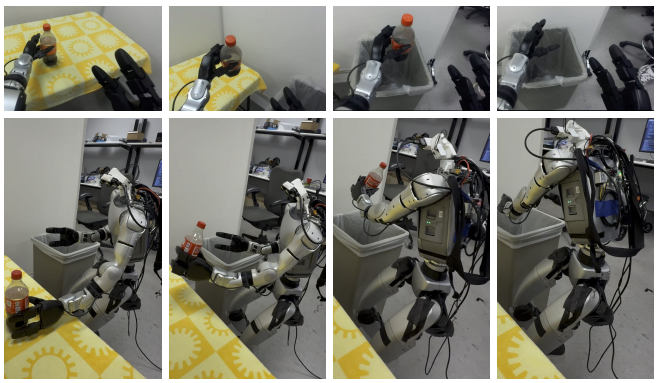
Fig. 5: Evaluation of in-distribution (I.D.) and out-of-distribution (O.O.D.) tracking results. Each figure shows the target vs. the actual commanded direction. The white area indicates I.D., meaning the command is used in both trajectory optimization and RL training. The grey area indicates O.O.D., meaning the command is not used in either trajectory optimization or RL training. The red and blue curves represent *w.o. AMO* and *AMO*, respectively.

torso orientation control rather than utilizing the entire lower body. For example, Unitree G1's waist pitch motor's positional limit is only ± 0.52 radians. In contrast, *AMO* achieves a significantly larger range of torso motion compared to other baselines, particularly in torso pitch, where it allows the robot to bend its upper body completely flat. Moreover, the policy demonstrates adaptive behaviours by leveraging leg motors to adjust the lower posture for stability. This is evident in torso roll control, where the robot slightly bends one leg to tilt its pelvis. Such adaptive behaviour is not observed in *w/o AMO*. Overall, by incorporating AMO, the policy not only expands the operational range of torso control but also improves torso stability by dynamically adapting to the commanded orientation.

AMO's advantage lies not only in accurately tracking in-distribution (I.D.) torso commands but also in its ability to effectively adapt to out-of-distribution (O.O.D.) commands. In Fig.5, we compare *AMO* and *w/o AMO* by evaluating their performance on both I.D. and O.O.D. commands. It is evident that *w/o AMO* struggles with O.O.D. commands: it fails to track torso pitch and yaw commands before they reach the sampled training ranges, and it does not track height commands at all, as discussed in the previous section. In contrast, *AMO* exhibits remarkable adaptability in tracking O.O.D. commands. It successfully tracks torso yaw command



(a) **Paper Bag Picking:** The task begins with the robot adjusting its torso to align its rubber hand with the handle. Then, the robot should turn and move to an appropriate distance to the destination table, stand still, and use its upper body joints coherently to leave the hand out of the handle.



(b) **Trash Bottle Throwing:** The task begins with the robot stooping and turning its upper body to the left to grab the trash bottle. The robot then turns its waist about 90 degrees to the right and throws the trash bottle into the trash bin.

Fig. 6: Autonomous tasks performed in the real-world setting. For each task, we collect 50 episodes using the teleoperation system and train an ACT to complete it autonomously.

up to ± 2, despite being trained only within the range of ± 1.57. Similarly, for height tracking, although the training distribution was limited to a range of $0.5m$ to $0.8m$, the policy generalizes well and accurately tracks height as low as $0.4m$. These results indicate that both the AMO module trained via imitation and the RL policy exhibit strong generalization capabilities, demonstrating AMO's robustness in adapting to whole-body commands beyond the training distribution.

C. How well does the AMO system perform in the real-world setting?

To showcase our policy's capability in torso orientation and base height control, we have performed a series of hyper-dexterous manipulation tasks using the AMO teleoperation framework. These teleoperation experiments are presented in Fig.1 and the supplementary videos.

To further highlight the robustness and hyper-dexterity of the AMO system, We select several challenging tasks that require adaptive whole-body control and perform imitation

Paper Bag Picking			
Setting	Picking	Moving	Placing
Stereo Input; Chunk Size 120	8/10	8/10	9/10
Mono Input; Chunk Size 120	9/10	7/10	10/10
Stereo Input; Chunk Size 60	7/10	7/10	6/10
Trash Bottle Throwing			
Setting		Picking	Placing
Stereo Input; Chunk Size 120		7/10	10/10
Mono Input; Chunk Size 120		4/10	10/10
Stereo Input; Chunk Size 60		5/10	10/10

TABLE IV: Training settings and success rate of individual stages of each task. Each training setting includes: using stereo (both left-right-eye images) or mono (single image) visual input; chunk size used in training action-chunking-transformer.

learning by collecting demonstrations, as illustrated in Fig.6. The performance and evaluations of these tasks are detailed below:

Paper Bag Picking: This task requires the robot to perform a loco-manipulation task in the absence of an end-effector, making it particularly demanding in terms of precision. We evaluate the impact of various training settings on task performance, as shown in Table.IV. With the most complete setting, the policy achieves a near-perfect success rate. While using only a single image slightly reduces the success rate, this task is particularly susceptible to shorter chunk sizes. When the robot places its hand under the bag handle and attempts to reorient, a shorter action memory often leads to confusion.

Trash Bottle Throwing: This task involves no locomotion; however, to successfully grasp the bottle and throw it into a bin positioned at another angle, the robot must execute extensive torso movements. The evaluation results, as shown in Table.IV, indicate that our system can learn to complete this task autonomously. Both stereo vision and longer action chunks enhance task performance. The benefits of stereo vision likely stem from an increased field-of-view (FoV) and implicit depth information, which are essential for grasping.

To demonstrate the effectiveness of our system and its potential in executing complex loco-manipulation tasks, we conduct an IL experiment on a long-horizon task that demands hyper-dexterous whole-body control: **Basket Picking.** In this task, the robot must crouch to a considerably low height and adjust its torso orientation to grasp two baskets positioned on either side, close to the ground. A recorded autonomous rollout is presented as a case study in Fig.7, showcasing the learned policy's execution in real-world conditions.

To illustrate how the robot coordinates its whole-body actuators, we visualize the joint angles of the waist motors and the left knee motor. The left knee motor is selected as a representative joint for height changing and walking. As shown in Fig.7, the task begins with the robot crouching and bending forward to align its hands with the baskets' height. This motion is reflected by an increase in the knee and waist pitch motor angles. Next, the robot tilts left and right to grasp the baskets, as indicated by the variations in the waist roll curve. After successfully retrieving the targets, the robot stands up, which is

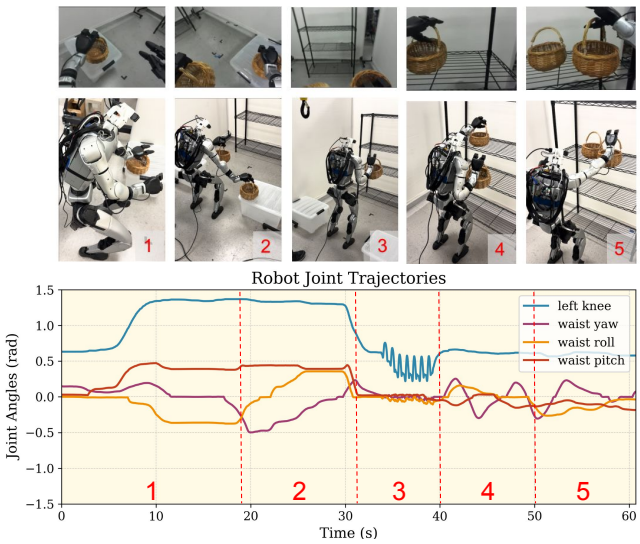


Fig. 7: **Basket Picking:** A complicated loco-manipulation task that also requires whole-body coordination. The task begins with the robot picking two baskets from left(1) and right(2), which are placed at low heights and close to the ground. Then, the robot stands up, moves forward(3), and puts two baskets on the shelf at eye level(4,5). The motor angle trajectories along the autonomous rollout are displayed and labeled to match the corresponding phases.

marked by a decrease in knee angle. The periodic fluctuations in the knee motor reading confirm that the robot is walking. Once stationary, the robot reaches to the left and right to place the basket on the shelf, with the waist yaw motor rotating back and forth to facilitate lateral torso movements. This case study clearly demonstrates our system's ability to leverage whole-body coordination to accomplish complex tasks effectively.

V. Conclusions and Limitations

We present **AMO**, a framework that integrates model-based trajectory optimization with model-free reinforcement learning to achieve whole-body control. Through extensive experiments in both simulation and real-robot platforms, we demonstrate the effectiveness of our approach. AMO enables real-world humanoid control with an unprecedented level of dexterity and precision. We hope that our framework provides a novel pathway for achieving humanoid whole-body control beyond conventional model-based whole-body methods and RL policies that rely on tracking human motion references.

While our decoupled approach introduces a new paradigm for whole-body control, its separated nature inherently limits the level of whole-body coordination it can achieve. For instance, in highly dynamic scenarios, humans naturally utilize not only their lower body and waist but also their arms to maintain balance. However, in our current setup, arm control is independent of the robot's base state. A balance-aware upper-body control mechanism could be explored by incorporating base state information into upper-limb joint-angle acquisition, potentially enhancing overall stability and adaptability.

VI. Acknowledgements

This project was supported, in part, by gifts from Amazon, Qualcomm and Meta.

References

[1] Unitree Robotics, G1, 2024, www.unitree.com/g1, [Online; accessed Jan. 2025].

[2] Zed Mini, Stereo Camera, 2024, www.robotis.us/dynamixel-xl330-m288-t, [Online, accessed Jun. 2024].

[3] Ananye Agarwal, Ashish Kumar, Jitendra Malik, and Deepak Pathak. Legged locomotion in challenging terrains using egocentric vision. In *Conference on Robot Learning*, pages 403–415. PMLR, 2023.

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

[5] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. *arXiv preprint arXiv:2307.15818*, 2023.

[6] Stéphane Caron, Yann De Mont-Marin, Rohan Budhiraja, Seung Hyeon Bang, Ivan Domrachev, and Simeon Nedelchev. Pink: Python inverse kinematics based on Pinocchio, 2024. URL https://github.com/stephane-caron/pink.

[7] Xuxin Cheng, Ashish Kumar, and Deepak Pathak. Legs as manipulator: Pushing quadrupedal agility beyond locomotion. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023.

[8] Xuxin Cheng, Kexin Shi, Ananye Agarwal, and Deepak Pathak. Extreme parkour with legged robots, 2023. URL https://arxiv.org/abs/2309.14341.

[9] Xuxin Cheng, Yandong Ji, Junming Chen, Ruihan Yang, Ge Yang, and Xiaolong Wang. Expressive whole-body control for humanoid robots, 2024. URL https://arxiv.org/abs/2402.16796.

[10] 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.

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

[12] Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. In *Proceedings of Robotics: Science and Systems (RSS)*, 2023.

[13] Cheng Chi, Zhenjia Xu, Chuer Pan, Eric Cousineau, Benjamin Burchfiel, Siyuan Feng, Russ Tedrake, and Shuran Song. Universal manipulation interface: In-the-wild robot teaching without in-the-wild robots. In *Proceedings of Robotics: Science and Systems (RSS)*, 2024.

[14] Matthew Chignoli, Donghyun Kim, Elijah Stanger-Jones, and Sangbae Kim. The mit humanoid robot: Design, motion planning, and control for acrobatic behaviors. In *2020 IEEE-RAS 20th International Conference on Humanoid Robots (Humanoids)*, pages 1–8. IEEE, 2021.

[15] Zichen Jeff Cui, Yibin Wang, Nur Muhammad Mahi Shafiullah, and Lerrel Pinto. From play to policy: Conditional behavior generation from uncurated robot data. *arXiv preprint arXiv:2210.10047*, 2022.

[16] Antonin Dallard, Mehdi Benallegue, Fumio Kanehiro, and Abderrahmane Kheddar. Synchronized human-humanoid motion imitation. *IEEE Robotics and Automation Letters*, 8(7):4155–4162, 2023. doi: 10.1109/LRA.2023.3280807.

[17] Timothée Darcet, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. Vision transformers need registers, 2023.

[18] Behzad Dariush, Michael Gienger, Bing Jian, Christian Goerick, and Kikuo Fujimura. Whole body humanoid control from human motion descriptors. In *2008 IEEE International Conference on Robotics and Automation*, pages 2677–2684. IEEE, 2008.

[19] Kourosh Darvish, Yeshasvi Tirupachuri, Giulio Romualdi, Lorenzo Rapetti, Diego Ferigo, Francisco Javier Andrade Chavez, and Daniele Pucci. Whole-body geometric retargeting for humanoid robots. In *2019 IEEE-RAS 19th International Conference on Humanoid Robots (Humanoids)*, pages 679–686, 2019. doi: 10.1109/Humanoids43949.2019.9035059.

[20] Helei Duan, Bikram Pandit, Mohitvishnu S Gadde, Bart Jaap van Marum, Jeremy Dao, Chanho Kim, and Alan Fern. Learning vision-based bipedal locomotion for challenging terrain. *arXiv preprint arXiv:2309.14594*, 2023.

[21] Alejandro Escontrela, Xue Bin Peng, Wenhao Yu, Tingnan Zhang, Atil Iscen, Ken Goldberg, and Pieter Abbeel. Adversarial motion priors make good substitutes for complex reward functions. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 25–32. IEEE, 2022.

[22] Jiawei Fu, Yunlong Song, Yan Wu, Fisher Yu, and Davide Scaramuzza. Learning deep sensorimotor policies for vision-based autonomous drone racing, 2022.

[23] Zipeng Fu, Ashish Kumar, Jitendra Malik, and Deepak Pathak. Minimizing energy consumption leads to the emergence of gaits in legged robots. *Conference on Robot Learning (CoRL)*, 2021.

[24] Zipeng Fu, Qingqing Zhao, Qi Wu, Gordon Wetzstein, and Chelsea Finn. Humanplus: Humanoid shadowing and imitation from humans, 2024. URL https://arxiv.org/abs/2406.10454.

[25] 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.

[26] Yuni Fuchioka, Zhaoming Xie, and Michiel Van de Panne. Opt-mimic: Imitation of optimized trajectories for dynamic quadruped behaviors. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5092–5098. IEEE, 2023.

[27] Huy Ha, Yihuai Gao, Zipeng Fu, Jie Tan, and Shuran Song. Umi on legs: Making manipulation policies mobile with manipulation-centric whole-body controllers. *arXiv preprint arXiv:2407.10353*, 2024.

[28] 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, 2024. URL https://arxiv.org/abs/2406.08858.

[29] 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.

[30] Tairan He, Wenli Xiao, Toru Lin, Zhengyi Luo, Zhenjia Xu, Zhenyu Jiang, Changliu Liu, Guanya Shi, Xiaolong Wang, Linxi Fan, and Yuke Zhu. Hover: Versatile neural whole-body controller for humanoid robots. *arXiv preprint arXiv:2410.21229*, 2024.

[31] Kazuo Hirai, Masato Hirose, Yuji Haikawa, and Toru Takenaka. The development of honda humanoid robot. In *Proceedings. 1998 IEEE international conference on robotics and automation (Cat. No. 98CH36146)*, volume 2, pages 1321–1326. IEEE, 1998.

[32] Marco Hutter, Christian Gehring, Dominic Jud, Andreas Lauber, C Dario Bellicoso, Vassilios Tsounis, Jemin Hwangbo, Karen Bodie, Peter Fankhauser, Michael Bloesch, et al. Anymal-a highly mobile and dynamic quadrupedal robot. In *IROS*, 2016.

[33] 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.

[34] Shuuji Kajita. A simple modeling for a biped walking pattern generation. In *Proceedings of the International Conference on Intelligent Robotics and Systems, Maui, HI, USA*, volume 29, 2001.

[35] Ichiro Kato. Development of wabot 1. *Biomechanism*, 2:173–214, 1973.

[36] Alexander Khazatsky, Karl Pertsch, Suraj Nair, Ashwin Balakrishna, Sudeep Dasari, Siddharth Karamcheti, Soroush Nasiriany, Mohan Kumar Srirama, Lawrence Yunliang Chen, Kirsty Ellis, Peter David Fagan, Joey Hejna, Masha Itkina, Marion Lepert, Yecheng Jason Ma, Patrick Tree Miller, Jimmy Wu, Suneel Belkhale, Shivin Dass, Huy Ha, Arhan Jain, Abraham Lee, Youngwoon Lee, Marius Memmel, Sungjae Park, Ilija Radosavovic, Kaiyuan Wang, Albert Zhan, Kevin Black, Cheng Chi, Kyle Beltran Hatch, Shan Lin, Jingpei Lu, Jean Mercat, Abdul Rehman, Pannag R Sanketi, Archit Sharma, Cody Simpson, Quan Vuong, Homer Rich Walke, Blake Wulfe, Ted Xiao, Jonathan Heewon Yang, Arefeh Yavary, Tony Z. Zhao, Christopher Agia, Rohan Baijal, Mateo Guaman Castro, Daphne Chen, Qiuyu Chen, Trinity Chung, Jaimyn Drake, Ethan Paul Foster, Jensen Gao, David Antonio Herrera, Minho Heo, Kyle Hsu, Jiaheng Hu, Donovon Jackson, Charlotte Le, Yunshuang Li, Kevin Lin, Roy Lin, Zehan Ma, Abhiram Maddukuri, Suvir Mirchandani, Daniel Morton, Tony Nguyen, Abigail O'Neill, Rosario Scalise, Derick Seale, Victor Son, Stephen Tian, Emi Tran, Andrew E. Wang, Yilin Wu, Annie Xie, Jingyun Yang, Patrick Yin, Yunchu Zhang, Osbert Bastani, Glen Berseth, Jeannette Bohg, Ken Goldberg, Abhinav Gupta, Abhishek Gupta, Dinesh Jayaraman, Joseph J Lim, Jitendra Malik, Roberto Martín-Martín, Subramanian Ramamoorthy, Dorsa Sadigh, Shuran Song, Jiajun Wu, Michael C. Yip, Yuke Zhu, Thomas Kollar, Sergey Levine, and Chelsea Finn. Droid: A large-scale in-the-wild robot manipulation dataset. 2024.

[37] Ashish Kumar, Zipeng Fu, Deepak Pathak, and Jitendra Malik. Rma: Rapid motor adaptation for legged robots. *arXiv preprint arXiv:2107.04034*, 2021.

[38] Zhongyu Li, Xuxin Cheng, Xue Bin Peng, Pieter Abbeel, Sergey Levine, Glen Berseth, and Koushil Sreenath. Reinforcement learning for robust parameterized locomotion control of bipedal robots. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2811–2817. IEEE, 2021.

[39] Zhongyu Li, Xue Bin Peng, Pieter Abbeel, Sergey Levine, Glen Berseth, and Koushil Sreenath. Reinforcement learning for versatile, dynamic, and robust bipedal locomotion control. *arXiv preprint arXiv:2401.16889*, 2024.

[40] Qiayuan Liao, Bike Zhang, Xuanyu Huang, Xiaoyu Huang, Zhongyu Li, and Koushil Sreenath. Berkeley humanoid: A research platform for learning-based control. *arXiv preprint arXiv:2407.21781*, 2024.

[41] Fukang Liu, Zhaoyuan Gu, Yilin Cai, Ziyi Zhou, Shijie Zhao, Hyunyoung Jung, Sehoon Ha, Yue Chen, Danfei Xu, and Ye Zhao. Opt2skill: Imitating dynamically-feasible whole-body trajectories for versatile humanoid loco-manipulation. *arXiv preprint arXiv:2409.20514*, 2024.

[42] 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.

[43] Naureen Mahmood, Nima Ghorbani, Nikolaus F. Troje, Gerard Pons-Moll, and Michael J. Black. Amass: Archive of motion capture as surface shapes. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2019. URL https://amass.is.tue.mpg.de.

[44] 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.

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

[46] Gabriel B Margolis, Tao Chen, Kartik Paigwar, Xiang Fu, Donghyun Kim, Sangbae Kim, and Pulkit Agrawal. Learning to jump from pixels. *arXiv preprint arXiv:2110.15344*, 2021.

[47] Gabriel B Margolis, Ge Yang, Kartik Paigwar, Tao Chen, and Pulkit Agrawal. Rapid locomotion via reinforcement learning. *arXiv preprint arXiv:2205.02824*, 2022.

[48] 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 *IEEE International Conference on Robotics and Automation (ICRA)*, 2020.

[49] Carlos Mastalli, Wolfgang Merkt, Josep Marti-Saumell, Henrique Ferrolho, Joan Solà, Nicolas Mansard, and Sethu Vijayakumar. A feasibility-driven approach to control-limited ddp. *Autonomous Robots*, 2022.

[50] Russell Mendonca, Shikhar Bahl, and Deepak Pathak. Alan : Autonomously exploring robotic agents in the real world. *ICRA*, 2023.

[51] Hirofumi Miura and Isao Shimoyama. Dynamic walk of a biped. *IJRR*, 1984.

[52] Federico L Moro and Luis Sentis. Whole-body control of humanoid robots. *Humanoid Robotics: A reference, Springer, Dordrecht*, 2019.

[53] Maxime Oquab, Timothée Darcet, Theo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Russell Howes, Po-Yao Huang, Hu Xu, Vasu Sharma, Shang-Wen Li, Wojciech Galuba, Mike Rabbat, Mido Assran, Nicolas Ballas, Gabriel Synnaeve, Ishan Misra, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision, 2023.

[54] Abhishek Padalkar, Acorn Pooley, Ajinkya Jain, Alex Bewley, Alex Herzog, Alex Irpan, Alexander Khazatsky, Anant Rai, Anikait Singh, Anthony Brohan, et al. Open x-embodiment: Robotic learning datasets and rt-x models. *arXiv preprint arXiv:2310.08864*, 2023.

[55] Jyothish Pari, Nur Muhammad Shafiullah, Sridhar Pandian Arunachalam, and Lerrel Pinto. The surprising effectiveness of representation learning for visual imitation. *arXiv preprint arXiv:2112.01511*, 2021.

[56] Luigi Penco, Nicola Scianca, Valerio Modugno, Leonardo Lanari, Giuseppe Oriolo, and Serena Ivaldi. A multimode teleoperation framework for humanoid loco-manipulation: An application for the icub robot. *IEEE Robotics and Automation Magazine*, 26(4):73–82, 2019. doi: 10.1109/MRA.2019.2941245.

[57] Xue Bin Peng, Erwin Coumans, Tingnan Zhang, Tsang-Wei Lee, Jie Tan, and Sergey Levine. Learning agile robotic locomotion skills by imitating animals. April 2020.

[58] Xue Bin Peng, Ze Ma, Pieter Abbeel, Sergey Levine, and Angjoo Kanazawa. AMP: adversarial motion priors for stylized physics-based character control. *ACM Trans. Graph.*, 40(4):1–20, July 2021.

[59] Xue Bin Peng, Ze Ma, Pieter Abbeel, Sergey Levine, and Angjoo Kanazawa. Amp: Adversarial motion priors for stylized physics-based character control. *ACM Transactions on Graphics (ToG)*, 40(4):1–20, 2021.

[60] Yuzhe Qin, Wei Yang, Binghao Huang, Karl Van Wyk, Hao Su, Xiaolong Wang, Yu-Wei Chao, and Dieter Fox. Anyteleop: A general vision-based dexterous robot arm-hand teleoperation system. In *Robotics: Science and Systems*, 2023.

[61] Ilija Radosavovic, Tete Xiao, Bike Zhang, Trevor Darrell, Jitendra Malik, and Koushil Sreenath. Learning humanoid locomotion with transformers. *arXiv preprint arXiv:2303.03381*, 2023.

[62] Marc H Raibert. *Legged robots that balance*. MIT press, 1986.

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

[64] Mingyo Seo, Steve Han, Kyutae Sim, Seung Hyeon Bang, Carlos Gonzalez, Luis Sentis, and Yuke Zhu. Deep imitation learning for humanoid loco-manipulation through human teleoperation. In *2023 IEEE-RAS 22nd International Conference on Humanoid Robots (Humanoids)*, pages 1–8. IEEE, 2023.

[65] Lucy Xiaoyang Shi, Zheyuan Hu, Tony Z. Zhao, Archit Sharma, Karl Pertsch, Jianlan Luo, Sergey Levine, and Chelsea Finn. Yell at your robot: Improving on-the-fly from language corrections. *arXiv preprint arXiv: 2403.12910*, 2024.

[66] Mohit Shridhar, Lucas Manuelli, and Dieter Fox. Perceiver-actor: A multi-task transformer for robotic manipulation. In *Conference on Robot Learning*, pages 785–799. PMLR, 2023.

[67] Jonah Siekmann, Kevin Green, John Warila, Alan Fern, and Jonathan Hurst. Blind bipedal stair traversal via sim-to-real reinforcement learning. *arXiv preprint arXiv:2105.08328*, 2021.

[68] Tomomichi Sugihara. Solvability-unconcerned inverse kinematics by the levenberg–marquardt method. *IEEE Transactions on Robotics*, 27(5):984–991, 2011. doi: 10.1109/TRO.2011.2148230. URL https://ieeexplore.ieee.org/document/5784347.

[69] Chen Tessler, Yoni Kasten, Yunrong Guo, Shie Mannor, Gal Chechik, and Xue Bin Peng. Calm: Conditional adversarial latent models for directable virtual characters. In *ACM SIGGRAPH 2023 Conference Proceedings*, SIGGRAPH '23, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9798400701597. doi:

10.1145/3588432.3591541. URL https://doi.org/10.1145/3588432.3591541.

[70] Chen Wang, Linxi Fan, Jiankai Sun, Ruohan Zhang, Li Fei-Fei, Danfei Xu, Yuke Zhu, and Anima Anandkumar. Mimicplay: Long-horizon imitation learning by watching human play. *arXiv preprint arXiv:2302.12422*, 2023.

[71] Tingwu Wang, Yunrong Guo, Maria Shugrina, and Sanja Fidler. Unicon: Universal neural controller for physics-based character motion, 2020.

[72] Eric R Westervelt, Jessy W Grizzle, and Daniel E Koditschek. Hybrid zero dynamics of planar biped walkers. *IEEE transactions on automatic control*, 48(1): 42–56, 2003.

[73] Ruihan Yang, Zhuoqun Chen, Jianhan Ma, Chongyi Zheng, Yiyu Chen, Quan Nguyen, and Xiaolong Wang. Generalized animal imitator: Agile locomotion with versatile motion prior. *arXiv preprint arXiv:2310.01408*, 2023.

[74] Ruihan Yang, Ge Yang, and Xiaolong Wang. Neural volumetric memory for visual locomotion control. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1430–1440, 2023.

[75] KangKang Yin, Kevin Loken, and Michiel Van de Panne. Simbicon: Simple biped locomotion control. *ACM Transactions on Graphics*, 2007.

[76] Yanjie Ze, Gu Zhang, Kangning Zhang, Chenyuan Hu, Muhan Wang, and Huazhe Xu. 3d diffusion policy: Generalizable visuomotor policy learning via simple 3d representations. In *Proceedings of Robotics: Science and Systems (RSS)*, 2024.

[77] Qiang Zhang, Peter Cui, David Yan, Jingkai Sun, Yiqun Duan, Arthur Zhang, and Renjing Xu. Whole-body humanoid robot locomotion with human reference. *arXiv preprint arXiv:2402.18294*, 2024.

[78] Tony Z Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. *arXiv preprint arXiv:2304.13705*, 2023.

[79] Ziwen Zhuang, Zipeng Fu, Jianren Wang, Christopher Atkeson, Sören Schwertfeger, Chelsea Finn, and Hang Zhao. Robot parkour learning. In *Conference on Robot Learning (CoRL)*, 2023.

APPENDIX A
RETARGETING

The hand retargeting problem is formulated as the following objective function:

$$\begin{aligned} \min_{\mathbf{q}_{\text{hand}}} \quad & \sum_{i=1}^{n} \|\mathbf{p}_i^{\text{human}} - \mathbf{p}_i^{\text{robot}}\|^2 \\ \mathbf{p}_i^{\text{robot}} &= FK(\mathbf{q}_{\text{hand}}) \\ \text{subject to} \quad & \mathbf{q}_{\min} \leq \mathbf{q}_{\text{hand}} \leq \mathbf{q}_{\max} \end{aligned}$$

This objective aims at minimizing the distances between corresponding keypoint vectors. Here, $\mathbf{p}_i^{\text{human}}$ denotes the i^{th} (scaled) keypoint vector of the human hand, while $\mathbf{p}_i^{\text{robot}}$ represents the corresponding vector of the robot hand. $\mathbf{p}_i^{\text{robot}} = FK(\mathbf{q}_{\text{hand}})$ is derived using forward kinematics from robot hand joint angles $\mathbf{q}_{\text{hand}}$. Since our system features a three-fingered robot hand,five vectors are employed for retargeting: three vectors ranging from the wrist to each fingertip and two vectors ranging from the thumb tip to the index and middle fingertips. Vectors originating from the wrist ensure that the overall robot hand pose closely resembles the human hand's pose, while vectors from the thumb tip facilitate precise and fine manipulations.

APPENDIX B
RL TRAINING DETAILS

Teacher Training Curriculum We use a carefully designed curriculum to regulate different tasks during RL training. Initially, the lower policy is treated as a pure locomotion policy with randomized target height. Then, we add delay to reduce sim-to-real gap. We then randomize target torso roll, pitch and yaw and activate related rewards. We finally sample arm actions from AMASS dataset and directly set the arm joint target position in the environment. Following the training schedule, the teacher policy gradually master complex whole-body control setting and be able to supervise a real world student.

Curriculum	Begin Global Step	Randomize Range
Target Height	100	0.5 $\sim$ 0.8
Target Torso Roll	2000	-0.7 $\sim$ 0.7
Target Torso Pitch	2000	-0.52 $\sim$ 1.57
Target Torso Yaw	2000	-1.57 $\sim$ 1.57
Target Arm	4000	AMASS

TABLE V: Curriculum Random Sample Steps

Table V shows the detail of our randomized curriculum schedule. When the global step is less than the curriculum begin global step, the target height is set to 0.8, the target torso roll, pitch, and yaw are set to 0, the target arm joint positions are set to the default arm joint positions. When the global step reaches the curriculum begin global step, target height, target torso roll, pitch, and yaw are randomly sampled in the ranges specified in the table, and the target arm joint positions are sampled from the AMASS dataset.

Curriculum	Begin Global Step	Final Global Step	Range
Gait frequency	5000	10000	1.3 $\sim$ 1.0
Stance Rate	1000	2000	0.2 $\sim$ 0.5

TABLE VI: Curriculum Linear Sample Steps

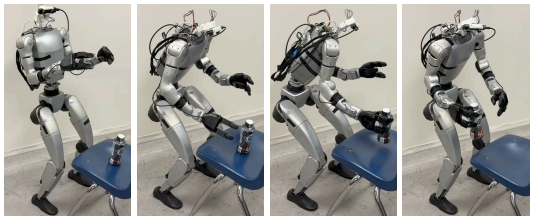


Fig. 8: Fully onboard whole-body teleoperation.

Table VI shows the detail of our linear sampled curriculum schedule. The gait frequency can be formulated as follows: The Gait Frequency and Stance Rate are defined as piecewise functions based on the Global Step:

$$\text{Gait Frequency}(s) = \begin{cases} 1.3 & s < 5k \\ 1.3 - \frac{0.3}{10k-5k}(s-5k), & 5k \leq s \leq 10k \\ 1.0 & s > 10k \end{cases}$$

$$\text{Stance Rate}(s) = \begin{cases} 0.2 & s < 5k \\ 0.2 + \frac{0.3}{2k-1k}(s-1k), & 1k \leq s \leq 2k \\ 0.5, & s > 2k \end{cases}$$

where s represents the global step.

APPENDIX C
DEPLOYMENT DETAILS

Some of our experiments are conducted with an external PC for easier debugging with an RTX 4090 GPU. Nonetheless, our entire teleoperation system and RL policy can be deployed onboard a Jetson Orin NX with 50Hz inference frequency, as shown in Fig. 8. We use OpenTelevision[11] to stream images and human poses between VR devices and the robot.