

# Mobile-TeleVision: Predictive Motion Priors for Humanoid Whole-Body Control

Chenhao Lu<sup>\*1</sup>, Xuxin Cheng<sup>\*1</sup>, Jialong Li<sup>\*1</sup>, Shiqi Yang<sup>1</sup>, Mazeyu Ji<sup>1</sup>, Chengjing Yuan<sup>1</sup>  
Ge Yang<sup>2</sup>, Sha Yi<sup>1</sup>, Xiaolong Wang<sup>1,3</sup>

<sup>1</sup>UC San Diego, <sup>2</sup>MIT, <sup>3</sup>NVIDIA

<https://mobile-tv.github.io/>

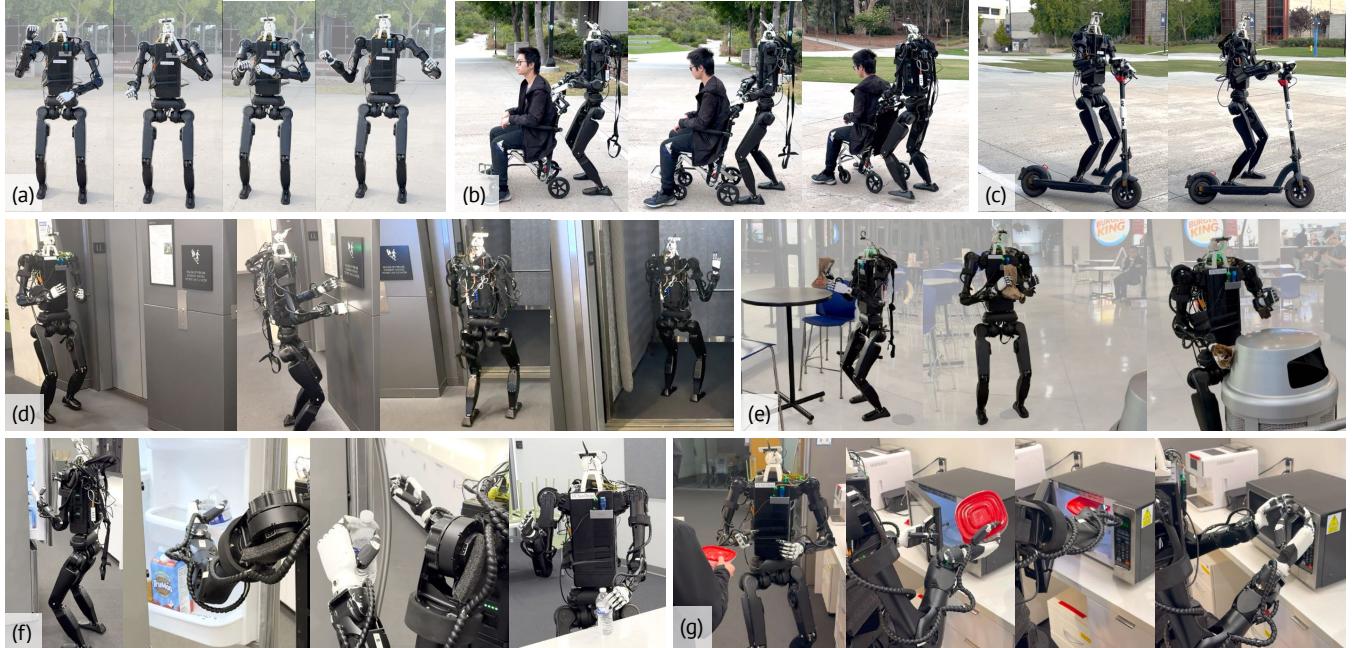


Fig. 1: Humanoid robot doing whole-body tasks that require both precise manipulation and robust locomotion. The robot can a) dance with its arms while the lower body maintains balance, b) push a wheelchair with a human, c) walk while carrying a scooter, d) use an elevator, e) collect and throw away trash, f) open the fridge, take and deliver a bottle of water, and g) put a food storage container into the microwave.

**Abstract**—Humanoid robots require both robust lower-body locomotion and precise upper-body manipulation. While recent Reinforcement Learning (RL) approaches provide whole-body loco-manipulation policies, they lack precise manipulation with high DoF arms. In this paper, we propose decoupling upper-body control from locomotion, using inverse kinematics (IK) and motion retargeting for precise manipulation, while RL focuses on robust lower-body locomotion. We introduce *PMP* (*Predictive Motion Priors*), trained with Conditional Variational Autoencoder (CVAE) to effectively represent upper-body motions. The locomotion policy is trained conditioned on this upper-body motion representation, ensuring that the system remains robust with both manipulation and locomotion. We show that CVAE features are crucial for stability and robustness, and significantly outperforms RL-based whole-body control in precise manipulation. With precise upper-body motion and robust lower-body locomotion control, operators can remotely control the humanoid to walk around and explore different environments, while performing diverse manipulation tasks.

## I. INTRODUCTION

Our ability to precisely grasp objects while being on the move is essential to our daily lives. Looking at humanoid control, what we demand of its arms is quite different from those of the legs: We want the arms to be versatile enough to apply variable forces, to hold objects at a fixed position and to move rapidly, such as when catching a ball. Meanwhile, its legs must also maintain balance. It should be able to walk while carrying a box, or to run while it dribbles. Current state-of-the-art humanoid whole-body control is unable to do this. The key missing piece is a way to integrate expressive and load-bearing movements of the upper body with control of the legs that fully captures common ranges of motion.

Recent advances in whole-body humanoid control using Reinforcement Learning (RL) have demonstrated the ability of robots to robustly execute a wide range of motions [1]–[3] from large-scale human motion dataset [4] or teleoperation [5], [6]. RL commonly exposes the robot in simulation to a

\* Equal contribution.

variety of target motions and failure modes. Such setups are able to produce performant policies that can directly deploy in the real world, but at the expense of expressiveness, due to Reinforcement Learning’s tendency to seek and overfit to particular modes. RL policy is also not well-suited for high DoF positional and orientation control [7], [8] and can output unpredictable actions for arms, especially in dynamic walking scenarios. It brings additional complexity to training the policy as well. These experimental results share a consensus with our observations that different mechanisms are required to control arms and legs.

The upper body’s movements can be efficiently computed using inverse kinematics (IK), without needing to address complex balance challenges. Based on this, we propose removing the upper-body tracking during RL training, leaving the RL controller responsible solely for robust walking based on velocity commands. This approach reduces exploration and training costs while ensuring high precision for the upper body using direct joint position control.

While decoupling upper and lower bodies brings benefits to the manipulation precision, prior studies [2], [3], [8] also show that fully decoupling the control of the upper and lower body may lead to instability or loss of balance, as it fails to account for the interaction between them. The goal of this work is to develop an integrated system that combines the precision and expressiveness of classical motion planning for the upper body, with the real-world robustness of locomotion control trained via deep reinforcement learning, in a single, whole-body control system. We do so by treating this as a representation learning problem, that starts with learning *Predictive Motion Priors (PMP)* of the upper body movement using a conditional variational autoencoder (CVAE) to encode upper-body motions as the observation of the policy. Specifically, we first train a CVAE on a motion dataset. Conditioned on past motion frames, the decoder of the CVAE predicts future motion frames with a randomly sampled latent vector. Next, we train an RL locomotion policy that observes the latent vector as an observation. The output of the RL policy only involves motors for the lower body and the upper body motor angles can be set in a general way such as IK or motion retargeting, resulting in higher precision.

This decoupled structure is particularly well-suited for teleoperation systems designed for loco-manipulation tasks, where the operator can control the robot’s mobility using simple velocity commands while simultaneously performing precise arm and hand operations. By leveraging inverse kinematics (IK) and hand re-targeting, the teleoperator can efficiently perform complex manipulation tasks without the need to focus on balancing or lower-body coordination. As shown in Figure 1, this enables us to perform fine, dynamic, and load-bearing manipulation tasks more efficiently, even when the robot is on the move. We compare our method and previous approaches in detail in Table I.

To test the generalizability of our method, we experiment with both the Fourier GR1 and the Unitree H1 robots in simulation, and evaluate the Unitree controller in the real world. We test its robustness by performing large motions in

| Metrics        | PMP<br>(Ours) | ExBody<br>[2] | OmniH2O<br>[3] | HumanPlus<br>[1] |
|----------------|---------------|---------------|----------------|------------------|
| Arm DoFs       | 7             | 4             | 5              | 5                |
| PrManip        | ✓             | ✗             | ✗              | ✗                |
| Robust Walking | ✓             | ✓             | ✓              | ✗                |

TABLE I: Comparisons with whole-body control methods. PrManip means whether arm motions are directly controlled and do not go through RL mapping. Robust walking means whether the work shows at least 10 seconds of robot walking.

a teleoperation setup that involves the robot bearing various loads (Figure 1). Our result showcases PMP’s ability to realize accurate upper-body control for a pair of high-DoF arms while remaining robust, a new capability previously unseen by prior methods.

## II. PROBLEM FORMULATION

We focus on providing a lower-body controller that tracks a given body velocity while maintaining stability when the upper body performs dynamic actions.

We consider this problem as a goal-conditioned motor policy  $\pi : \mathcal{G} \times \mathcal{S} \rightarrow \mathcal{A}$ . The goal space  $\mathcal{G}$  includes the target behavior of both the root and the upper body.  $\mathcal{S}$  consists of proprioception and auxiliary information obtained from observations. The auxiliary information in our work is the motion prior, which we will go into detail in the next section.  $\mathcal{A}$  is the action space, which contains the PD torque control input for each joint.

a) *Goal Space for Locomotion Control and Upper Body Motion*: The goal space  $\mathcal{G}$  contains two parts: the locomotion goal  $\mathcal{G}^m = \langle \mathbf{v}, r, p, y, h \rangle$  for the lower body, and the upper body motion goal  $\mathcal{G}^u = \langle \mathbf{q}^{\text{upper}} \rangle$ . The base locomotion is to track the given linear velocity  $\mathbf{v} \in \mathbb{R}^3$ , orientation described by Euler angles roll  $r$ , pitch  $p$ , and yaw  $y$ , and the base height  $h$ .  $\mathbf{q}^{\text{upper}}$  is the target position of upper body joints including the waist, shoulders, elbows, wrists, and hands. For the target platforms of this paper, H1 (with dexterous hands) and GR1 contain 27 and 19 upper-body joints, respectively.

b) *Decoupled control policy*: We decouple the control policy by  $\pi = [\pi^{\text{upper}}, \pi^{\text{lower}}]$ .  $\pi^{\text{upper}}$  is the open loop control, directly outputting the upper body motion goal  $\mathbf{q}^{\text{upper}}$ .  $\pi^{\text{lower}}$  is trained by RL. It observes proprioception  $\mathbf{s}_t = [\mathbf{q}_t, \dot{\mathbf{q}}_t, \boldsymbol{\omega}_t, \mathbf{g}_t, \mathbf{a}_{t-1}]$  and the locomotion command where  $\mathbf{q}_t, \dot{\mathbf{q}}_t$  is current joint positions and velocities,  $\boldsymbol{\omega}_t \in \mathbb{R}^3$  is current base angular velocity,  $\mathbf{g}_t$  is current projected gravity,  $\mathbf{a}_{t-1} \in \mathbb{R}^{12}$  is the previous action. Both GR1 and H1 have 12 joints in the lower body. Our method takes auxiliary information as input including gait periodic signals and motion prior.

## III. METHOD

As illustrated in Figure 2, our training pipeline consists of three stages. The first stage involves preprocessing a given human motion dataset, where we apply a local rotation mapping to convert the local rotation matrices into joint angles on the robots. We follow the data filtering and retargeting process described in ExBody [2] to get a retargeted

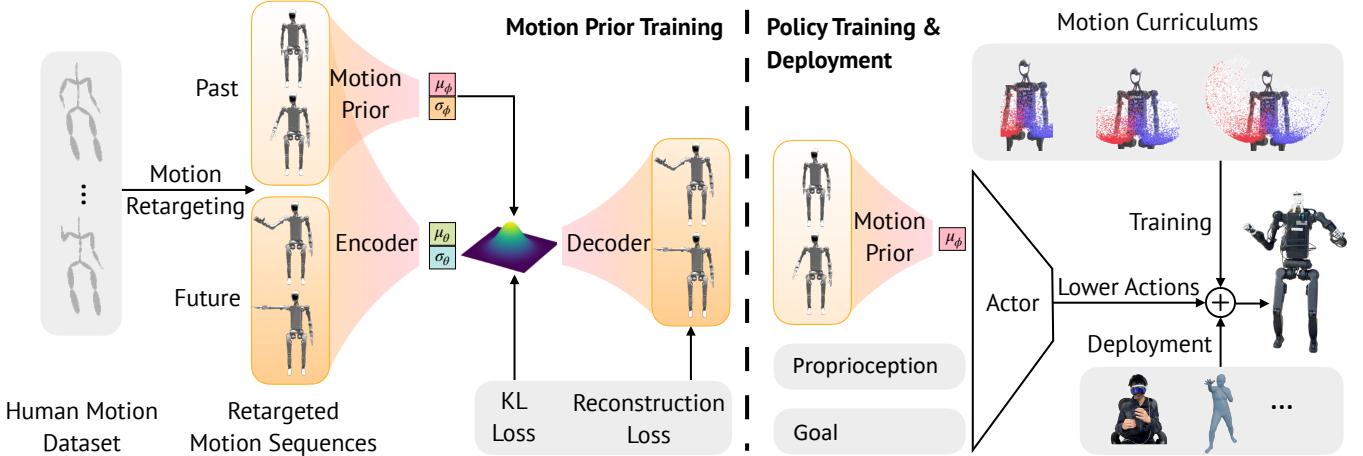


Fig. 2: The training pipeline consists of three stages: (a) preprocessing of the motion dataset by mapping local rotation, (b) training a CVAE to capture prior knowledge of upper body human motion, and (c) RL training where the upper body is controlled using sampled target joint positions and the lower body is trained using prior motion representations.

motion dataset  $\mathcal{M}$ . In the second stage, we train the model to autoregressively generate future motions based on the retargeted dataset. The generated motion sequence is then used to extract motion priors, which we introduce in detail in section III-A. In the third stage, we directly control the upper body using sampled target joint positions in a curriculum schedule. This is then combined with the lower body control for whole-body movements.

### A. Motion Prior Training

Since our control policy decouples the upper and lower body, the robustness of the lower body control becomes critical when the upper body controller has separate policies. The key to addressing this challenge lies in informing the lower body control with prior knowledge of upper body movements. Specifically, we estimate the future motions of the upper body from previous motions based on the retargeted dataset, and train a representation to be incorporated in the state space of the lower body control. We define this representation as *Predictive Motion Priors* (PMP).

Conditional variational autoencoder (CVAE) has been shown as an effective motion representation [9], [10]. A CVAE architecture contains a prior model  $R$ , an encoder  $E$ , and a decoder  $D$ . Given an upper-body motion sequence within a time frame of  $t - W$  to  $t + W$ , where  $W$  is a time window, the prior model  $R$  calculates a prior distribution for a latent vector  $z_t$ . It allows us to generate the essential future upper-body motion from  $z_t \in \mathbb{R}^H$ .  $z_t$  is the desired motion prior for our policy.

Formally, we model the encoder and prior distribution as diagonal Gaussian:

$$R(\cdot|M_t^0) = \mathcal{N}(\cdot|\mu_\phi(M_t^0), \sigma_\phi(M_t^0)) \quad (1)$$

$$E(\cdot|M_t^0, M_t^1) = \mathcal{N}(\cdot|\mu_\theta(M_t^0, M_t^1), \sigma_\theta(M_t^0, M_t^1)), \quad (2)$$

where  $M_t^0 = \mathbf{q}_{t-W:t-1}^{\text{upper}}$  and  $M_t^1 = \mathbf{q}_{t:t+W-1}^{\text{upper}}$  are two consecutive motion sequences.

During training,  $(M_t^0, M_t^1)$  are sampled from the dataset  $\mathcal{M}$ . The objective in the evidence lower bound is as follows:

$$\begin{aligned} \log P(M_t^1|M_t^0) \geq & \mathbb{E}_{z_t \sim E} [D(M_t^1|z_t, M_t^0)] \\ & - \text{KL}(E(z_t|M_t^0, M_t^1) \| R(z_t|M_t^0)). \end{aligned}$$

In the specific implementation, the structures of  $R$ ,  $E$ , and  $D$  are three-layer MLPs, and the size  $H$  of  $z_t$  is 64. The decoder  $D$  is responsible for reconstructing the future motion sequence  $M_t^1$  from the latent vector  $z_t$ . The decoder learns to map  $z_t$  back into the space of motion sequences, aiming to generate a future motion sequence  $M_t^1$  that closely resembles the original future motion sequence. The window length  $W$  corresponds to a real time of 1s. Due to different control frequencies,  $W$  is 50 for H1 and 100 for GR1. In the next stage, we use the prior model  $R$  to calculate motion prior  $z_t$ , which is as state input for the control policy.

### B. Decoupling Policy Training

We use PPO [11] to train the lower-body policy. During training, the environment will randomly sample a motion sequence  $M_t = \mathbf{q}_{1:T}^{\text{upper}}$  from the retargeted motion dataset  $\mathcal{M}$  at the beginning of a new episode. At step  $t$ ,  $\mathbf{q}_{1:T}^{\text{upper}}$  will be directly set as the PD target of the humanoid's upper body.

To overcome the exploration burden for RL with hard motions, we introduce an adaptive difficulty factor  $\alpha_i$  for each motion  $i$ , adjusting the amplitude of the upper body motions in a curriculum schedule. Given a sequence of upper-body motion  $i$ , denote the target joint position to be  $\mathbf{q}_{\text{target}}^{\text{upper}}$ . For the PD controller of the joint position during the curriculum, we follow

$$\mathbf{q}_{\text{curriculum}}^{\text{upper}} = \mathbf{q}_0 + \alpha_i(\mathbf{q}_{\text{target}}^{\text{upper}} - \mathbf{q}_0) \quad (3)$$

where  $\mathbf{q}_0$  is the default joint position for the robot.

The difficulty factor  $\alpha_i$  changes when the environment is reset with the following rule:

$$\alpha_i \leftarrow \begin{cases} \alpha_i + 0.05 & \text{survival time} \geq 90\% \text{ total time} \\ \alpha_i - 0.01 & \text{otherwise} \end{cases} \quad (4)$$

For reward design, we follow ExBody [2] with additional explicit gait periodicity rewards, as seen in previous work [12], [13].

#### IV. EXPERIMENTS

We use the Unitree H1 robot for our real-world experiments and an additional Fourier GR-1 robot for simulation experiments. The details of the H1 robot are shown in Figure 4 right. We aim to answer the following questions by both simulation and real-world experiments:

- 1) How well does **PMP** perform compared to those methods in learning-based upper-body control?
- 2) How does the tracking precision and the stability of **PMP** perform when a disturbance occurs?
- 3) How does **PMP** perform in real-world scenarios requiring both robust locomotion and precise manipulation?

##### A. Simulation Results

To answer **Q1** in simulation, the performance of **PMP** is evaluated by comparing it with the following baseline:

- **Exbody:** This baseline's upper body control policy is RL-trained with tracking rewards for upper body target keypoints and joint positions. The code for H1 is sourced from the Exbody's [2] codebase. The code for GR1 is sourced from our own reimplementation.
- **Exbody (Whole):** This baseline's upper body control policy is RL-trained with tracking rewards for whole-body target keypoints and joint positions. The H1 code is sourced from Exbody's codebase, and similar methods include HumanoidPlus [1] and OmniH2O [3] in terms of reward settings and RL training.
- **PMP without motion prior:** This baseline uses decoupled policy but without motion prior.
- **PMP without motion curriculum:** This is an ablation study on motion curriculum. In this baseline, the  $\alpha$  in the motion curriculum is always set to 1.

Our metrics are as follows:

- 1) *Precision:* upper joint position error  $E_{\text{jpe}}^{\text{upper}}$ , upper key point position error  $E_{\text{kpe}}^{\text{upper}}$ .
- 2) *Smoothness:* upper joint acceleration  $E_{\text{acc}}^{\text{upper}}$ , upper action difference  $E_{\text{action}}^{\text{upper}}$ .
- 3) *Locomotion:* command linear velocity error  $E_{\text{vel}}$ , command angular velocity error  $E_{\text{ang}}$ .
- 4) *Stability:* lower joint acceleration  $E_{\text{acc}}^{\text{lower}}$ , lower action difference  $E_{\text{action}}^{\text{lower}}$ , projected gravity  $E_g$ .

Results are shown in Table II. We observe that the control policy employing the decoupling method proposed in this paper demonstrates significantly higher *precision* and *smoothness* in tracking upper-body motion targets compared to the use of a learnable policy for the upper body. For *locomotion* and *stability*, results from the H1 reveal that the decoupling strategy leads to a decline in performance metrics for motion command tracking and lower-body stability, due to the common trade-offs in loco-manipulation. However, PMP, by leveraging motion prior and motion curriculum, improves lower-body stability, achieving performance metrics comparable to the best baseline.

To answer **Q2**, we conduct robustness tests under perturbations such as pushing the robot and varying the playback speed of upper-body motions. We assess precision by the average upper DoF error  $E_{\text{jpe}}^{\text{upper}}$  and stability by the average projected gravity  $E_g$ . The results are presented in Figure 3. As disturbances increase, the precision comparison shows that PMP consistently exhibits superior tracking accuracy compared to methods like ExBody by about 40%, showcasing the advantage of decoupled upper-body control. Regarding stability, we observe that the introduction of motion prior and motion curriculum significantly improves the policy's robustness by about 25% in terms of projected gravity. Overall, our results demonstrate that *PMP* achieves precise upper-body control while maintaining robustness.

##### B. Teleoperation Setup

To test the performance of our method in the real world, we set up a real-world teleoperation system with the Unitree H1 robot. The teleoperation of the active neck and upper body follows the pipeline described in [5]. The operator uses the Apple Vision Pro to track hand and head movements, which are then translated into the robot's joint angles through inverse kinematics (IK) and motion retargeting. The stereo camera's stream is transmitted to the VR headset as an immersive display in real-time.

For the lower body control, since our policy receives  $\langle \mathbf{v}, rpy, h \rangle$  as its locomotion goal, it is compatible with any input device capable of providing these commands. Our system offers two different teleoperation setups for locomotion control. The first involves separate control: one operator controls the upper body using the VR headset, while another operator controls the lower body using a remote controller with joysticks. The second setup features a unified control, where a single user who controls the upper body also controls the lower body using pedals, as depicted in Figure 4 left. This unified setup allows the entire humanoid to be teleoperated by one person. Our teleoperation system is capable of performing a diverse set of loco-manipulation tasks, as shown in Figure 1.

##### C. Real-World Results

To answer **Q3**, we conduct experiments to evaluate the locomotion robustness and manipulation precision.

1) *Robustness Test:* To demonstrate that our method maintains stable locomotion control, we conduct a robustness test, as in Figure 5. The robot's robustness is evaluated by applying a pushing force to its torso and measuring the maximum pitch velocity, the maximum pitch deviation, and the recovery time. The policies tested include *Ours* (PMP) and *Ours* (PMP) w/o motion prior. We present the robustness testing results for both walking and standing modes separately, as shown in Table III.

When the robot is pushed while walking, both policies perform equally well at recovering to stable states. When pushed while standing, the policy trained with motion prior exhibits less deviation from the stable pose and requires less time to recover. The recovery process in the standing

| Method                                | $E_{acc}^{upper} \downarrow$ | $E_{action}^{upper} \downarrow$ | $E_{kpe}^{upper} \downarrow$ | $E_{jpe}^{upper} \downarrow$ | $E_{vel} \downarrow$ | $E_{ang} \downarrow$ | $E_{acc}^{lower} \downarrow$ | $E_{action}^{lower} \downarrow$ | $E_g \downarrow$ |
|---------------------------------------|------------------------------|---------------------------------|------------------------------|------------------------------|----------------------|----------------------|------------------------------|---------------------------------|------------------|
| <b>(a) Simulation results for H1</b>  |                              |                                 |                              |                              |                      |                      |                              |                                 |                  |
| Ours (PMP)                            | <b>5.07</b>                  | <b>0.00280</b>                  | 0.02161                      | <b>0.02355</b>               | 0.01296              | <b>0.01947</b>       | 13.28                        | 0.01651                         | <b>1.301</b>     |
| Exbody                                | 11.93                        | 0.01595                         | 0.02303                      | 0.03125                      | 0.00784              | 0.02105              | 14.69                        | 0.01651                         | 1.367            |
| Exbody (Whole)                        | 9.97                         | 0.01576                         | 0.02474                      | 0.03788                      | <b>0.00782</b>       | 0.02051              | 14.76                        | <b>0.01553</b>                  | 1.454            |
| Ours w/o motion prior                 | 5.24                         | 0.00282                         | <b>0.02146</b>               | 0.02425                      | 0.01349              | 0.02081              | <b>13.27</b>                 | 0.01623                         | 1.585            |
| Ours w/o curriculum                   | 5.39                         | 0.00289                         | 0.02150                      | 0.02440                      | 0.01481              | 0.02057              | 13.54                        | 0.01645                         | 1.597            |
| <b>(b) Simulation results for GR1</b> |                              |                                 |                              |                              |                      |                      |                              |                                 |                  |
| Ours (PMP)                            | <b>3.08</b>                  | <b>0.00054</b>                  | <b>0.02165</b>               | <b>0.00730</b>               | 0.00409              | <b>0.00886</b>       | <b>7.83</b>                  | 0.00784                         | <b>0.420</b>     |
| Exbody (Reimplementation)             | 7.47                         | 0.00200                         | 0.02282                      | 0.01006                      | 0.00978              | 0.01448              | 9.74                         | <b>0.00425</b>                  | 0.835            |
| Ours w/o motion prior                 | 3.11                         | <b>0.00054</b>                  | 0.02166                      | 0.00730                      | <b>0.00393</b>       | 0.00899              | 8.07                         | 0.00773                         | 0.442            |
| Ours w/o curriculum                   | 3.38                         | 0.00055                         | 0.02166                      | 0.00737                      | 0.00424              | 0.01105              | 9.24                         | 0.00916                         | 0.546            |

TABLE II: Comparisons with baselines. We sample 5 trajectories for each motion from the dataset in simulation and report their mean episode metrics. The locomotion commands are from the base state of the corresponding motion.

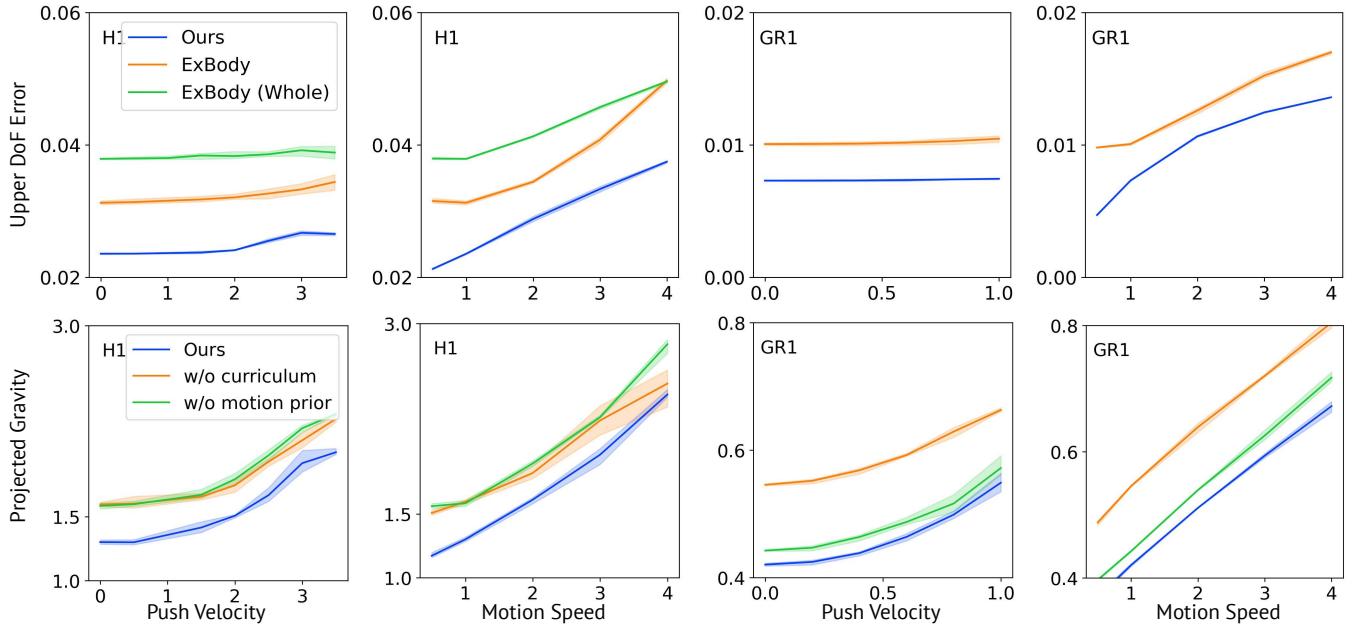


Fig. 3: Evaluation for precision (Top) and stability (Bottom) under disturbance (lower is better for all figures). We sample 5 trajectories for each motion from the motion dataset in simulation and report their mean episode metrics for each type of disturbance. The locomotion commands are from the base state of the corresponding motion. The robot is pushed every 5 seconds by a sudden increase in velocity to a value of `push_vel`. Motion speed is changed by multiplying the frame rate of retargeted motion by the corresponding factor. PMP shows lower values, which correspond to better performance.

mode generally takes longer than in walking mode, as the robot may sometimes need to take additional steps to adjust and stabilize itself if it cannot find a stationary leg position initially. The policy trained with motion prior significantly reduces both the occurrence and duration of this scenario.

2) *Precision Evaluation:* We evaluate the precision of our method in arm tracking by measuring the distance between the tracking objectives (left & right wrist poses tracked by the Apple Vision Pro) and the actual poses of the robot wrists in the real world. Since directly measuring the real-world poses is intractable without accurate motion capture devices, we instead use poses computed through forward kinematics based on arm motor readings. The tracking distances are

reported in Table IV. The results are computed along trajectories collected while the robot performs the four tabletop tasks as described in [5].

The results suggest that our system closely tracks the movement of human wrists. Notably, there are two layers of smoothing operations between the human wrist poses and robot joint angles: a linear filter applied on the wrist poses, and PD controllers. The tracking errors are also exaggerated by interactions with objects. Despite all these factors, our system consistently maintains a low tracking error.

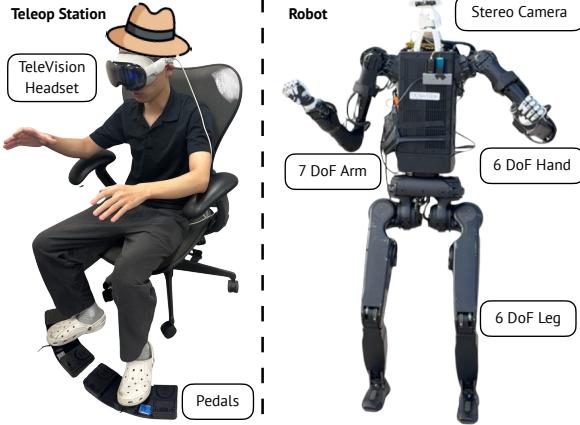


Fig. 4: Left: Unified teleoperation setup, where a single user uses Apple Vision Pro to control the upper body and uses pedals to send lower body locomotion commands. Right: H1 robot with customized neck and head. We mount a 6-DoF dexterous hand for each arm. The active neck and stereo camera are for immersive teleoperation following [5].

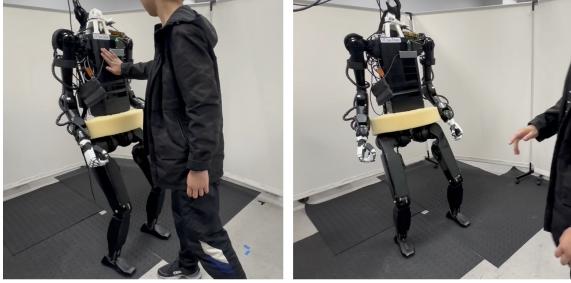


Fig. 5: Robustness testing. Left: robot being pushed while standing; Right: robot recovers to the stable standing pose.

## V. RELATED WORK

**Whole-Body Control for Humanoid Robots.** For humanoid robots to be useful, they need to be both robust in mobility and precise in manipulation. This is previously primarily achieved by dynamics modeling and control [14]–[27]. More recently, deep reinforcement learning methods have shown promise in learning complex locomotion skills for legged robots [8], [28]–[46]. Whole-body locomotion from high-dimensional sensory inputs has been studied for quadrupeds [8], [47]–[49] and humans [1]–[3], [32], [50], [51]. In [1]–[3], authors train whole-body policies that output the actions for the entire body for robustness. [2] decouples the upper and lower bodies’ objectives to compensate for morphological differences between humans and robots. [1] trains one transformer for whole-body control of the humanoid robot and another transformer for imitation learning. [3] trains goal-conditioned policies for various downstream tasks. However, these works focus on whole-body control with smaller arm DoFs (4 or 5) and do not study how the precision of the upper body affects manipulation.

**Loco-Manipulation.** Quadrupedal robot can either use their legs for manipulation [48], [52]–[55], or with an additional mounted manipulator [7], [8], [49], [56], [57]. Hu-

| Metrics                                   | Ours                    | Ours w/o motion prior |
|---|-------------------------|-----------------------|
| Max (abs) pitch vel (rad/s)               | <b>0.253</b>            | 0.296                 |
| Pitch deviation range (rad)               | (-0.152, <b>0.035</b> ) | (-0.138, 0.050)       |
| Recovery time (s)                         | <b>0.59</b>             | 0.66                  |
| (a) Robustness testing results - walking  |                         |                       |
| Metrics                                   | Ours                    | Ours w/o motion prior |
| Max (abs) pitch vel (rad/s)               | <b>0.241</b>            | 0.331                 |
| Pitch deviation range (rad)               | (-0.150, <b>0.029</b> ) | (-0.161, 0.051)       |
| Recovery time (s)                         | <b>4.46</b>             | 5.38                  |
| (b) Robustness testing results - standing |                         |                       |

TABLE III: Robustness testing results; each number is averaged over ten pushes. Recovery time represents the time elapsed before the robot can steadily walk or stand in a stationary location again.

| Task      | x(m)  | y(m)   | z(m)  | roll(rad) | pitch(rad) | yaw(rad) |
|-----------|-------|--------|-------|-----------|------------|----------|
| Sorting   | 0.011 | -0.032 | 0.008 | 0.002     | 0.040      | 0.016    |
| Insertion | 0.027 | -0.018 | 0.006 | -0.021    | -0.009     | 0.036    |
| Folding   | 0.044 | -0.018 | 0.002 | -0.013    | -0.010     | 0.171    |
| Unloading | 0.027 | -0.028 | 0.008 | -0.045    | 0.040      | 0.093    |

TABLE IV: Arm tracking errors. For each task, the errors are averaged over twenty trajectories, the equivalency of roughly 30 minutes of arm movement under 50 Hz control frequency.

manoid robots naturally have two arms for manipulation and two legs for locomotion. [58] uses hierarchical visual-motor policy with implicit-hierarchical whole-body control to do fine manipulation. However, no robust walking is shown in the real world and the focus is on static manipulation. [59] studies how to complete box loco-manipulation task with sim-to-real techniques and lacks the ability to generalize to a wide variety of tasks.

**Motion Representation Learning.** Human and robot motions are in high-dimensional temporal-spatial space. A good representation of motion facilitates motion understanding, generation, and imitation. Adversarial imitation methods such as [60]–[63] work well with small motion datasets but suffer from mode collapse as the motions scale. [64]–[67] learn reusable motion representations via conditional variational encoders in simulation to generate novel motions. However, how CVAE can coordinate whole-body movements for real humanoid robots has not been well studied.

## VI. DISCUSSION AND LIMITATION

This paper introduces a novel whole-body controller for humanoid robots, which separately models the upper body with IK and retargeting, and the lower body with RL controllers. To bridge upper and lower bodies for robust control, we introduce Predictive Motion Priors to encode upper-body information for lower-body RL training. We show superior performance in both simulation and real-robot whole-body control, as well as teleoperation experiments.

While our method allows the humanoid robot to conduct a variety of loco-manipulation tasks, the separation of upper-body and lower-body controls limits the robot from certain agile tasks. However, the current humanoid hardware still

has too few degrees of freedom compared to humans, which makes RL tracking agile whole-body motion very challenging from the practical perspective. The whole-body policy accepts various input commands such as forward/lateral, and yaw velocity, bringing burden for the human teleoperator since the arm and hand movements are mapped to control the robot's arms and hands. We need a better human-robot interaction interface to address these challenges.

## REFERENCES

- [1] Z. Fu, Q. Zhao, Q. Wu, G. Wetzstein, and C. Finn, "Humanplus: Humanoid shadowing and imitation from humans," 2024. [Online]. Available: <https://arxiv.org/abs/2406.10454> 1, 2, 4, 6
- [2] X. Cheng, Y. Ji, J. Chen, R. Yang, G. Yang, and X. Wang, "Expressive whole-body control for humanoid robots," 2024. [Online]. Available: <https://arxiv.org/abs/2402.16796> 1, 2, 4, 6
- [3] T. He, Z. Luo, X. He, W. Xiao, C. Zhang, W. Zhang, K. Kitani, C. Liu, and G. Shi, "Omnih2o: Universal and dexterous human-to-humanoid whole-body teleoperation and learning," 2024. [Online]. Available: <https://arxiv.org/abs/2406.08858> 1, 2, 4, 6
- [4] N. Mahmood, N. Ghorbani, N. F. Troje, G. Pons-Moll, and M. J. Black, "Amass: Archive of motion capture as surface shapes," in *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2019. [Online]. Available: <https://amass.is.tue.mpg.de> 1
- [5] X. Cheng, J. Li, S. Yang, G. Yang, and X. Wang, "Open-television: Teleoperation with immersive active visual feedback," *CoRL*, 2024. 1, 4, 5, 6
- [6] Y. Park and P. Agrawal, "Using apple vision pro to train and control robots," 2024. [Online]. Available: <https://github.com/Improbable-AI/VisionProTeleop> 1
- [7] M. Liu, Z. Chen, X. Cheng, Y. Ji, R. Qiu, R. Yang, and X. Wang, "Visual whole-body control for legged loco-manipulation," *CoRL*, 2024. 2, 6
- [8] J. Fu, Y. Song, Y. Wu, F. Yu, and D. Scaramuzza, "Learning deep sensorimotor policies for vision-based autonomous drone racing," 2022. 2, 6
- [9] D. Rempe, T. Birdal, A. Hertzmann, J. Yang, S. Sridhar, and L. J. Guibas, "Humor: 3d human motion model for robust pose estimation," in *International Conference on Computer Vision (ICCV)*, 2021. 3
- [10] Z. Luo, J. Cao, J. Merel, A. Winkler, J. Huang, K. Kitani, and W. Xu, "Universal humanoid motion representations for physics-based control," *arXiv preprint arXiv:2310.04582*, 2023. 3
- [11] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017. 3
- [12] Q. Zhang, P. Cui, D. Yan, J. Sun, Y. Duan, A. Zhang, and R. Xu, "Whole-body humanoid robot locomotion with human reference," *arXiv preprint arXiv:2402.18294*, 2024. 4
- [13] G. B. Margolis and P. Agrawal, "Walk these ways: Tuning robot control for generalization with multiplicity of behavior," in *Conference on Robot Learning*. PMLR, 2023, pp. 22–31. 4
- [14] H. Miura and I. Shimoyama, "Dynamic walk of a biped," *IJRR*, 1984. 6
- [15] K. Yin, K. Loken, and M. Van de Panne, "Simbicon: Simple biped locomotion control," *ACM Transactions on Graphics*, 2007. 6
- [16] M. Hutter, C. Gehring, D. Jud, A. Lauber, C. D. Bellicoso, V. Tsounis, J. Hwangbo, K. Bodie, P. Fankhauser, M. Bloesch *et al.*, "Anymal-a highly mobile and dynamic quadrupedal robot," in *IROS*, 2016. 6
- [17] F. L. Moro and L. Sentis, "Whole-body control of humanoid robots," *Humanoid Robotics: A reference*, Springer, Dordrecht, 2019. 6
- [18] B. Dariush, M. Gienger, B. Jian, C. Goerick, and K. Fujimura, "Whole body humanoid control from human motion descriptors," in *2008 IEEE International Conference on Robotics and Automation*. IEEE, 2008, pp. 2677–2684. 6
- [19] S. Kajita, F. Kanehiro, K. Kaneko, K. Yokoi, and H. Hirukawa, "The 3d linear inverted pendulum mode: A simple modeling for a biped walking pattern generation," in *Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the the Next Millennium (Cat. No. 01CH37180)*, vol. 1. IEEE, 2001, pp. 239–246. 6
- [20] E. R. Westervelt, J. W. Grizzle, and D. E. Koditschek, "Hybrid zero dynamics of planar biped walkers," *IEEE transactions on automatic control*, vol. 48, no. 1, pp. 42–56, 2003. 6
- [21] I. Kato, "Development of wabot 1," *Biomechanism*, vol. 2, pp. 173–214, 1973. 6
- [22] K. Hirai, M. Hirose, Y. Haikawa, and T. Takenaka, "The development of honda humanoid robot," in *Proceedings. 1998 IEEE international conference on robotics and automation (Cat. No. 98CH36146)*, vol. 2. IEEE, 1998, pp. 1321–1326. 6
- [23] M. Chignoli, D. Kim, E. Stanger-Jones, and S. Kim, "The mit humanoid robot: Design, motion planning, and control for acrobatic behaviors," in *2020 IEEE-RAS 20th International Conference on Humanoid Robots (Humanoids)*. IEEE, 2021, pp. 1–8. 6
- [24] A. Dallard, M. Benallegue, F. Kanehiro, and A. Kheddar, "Synchronized human-humanoid motion imitation," *IEEE Robotics and Automation Letters*, vol. 8, no. 7, pp. 4155–4162, 2023. 6
- [25] K. Darvish, Y. Tirupachuri, G. Romualdi, L. Rapetti, D. Ferigo, F. J. A. Chavez, and D. Pucci, "Whole-body geometric retargeting for humanoid robots," in *2019 IEEE-RAS 19th International Conference on Humanoid Robots (Humanoids)*, 2019, pp. 679–686. 6
- [26] L. Penco, N. Scianca, V. Modugno, L. Lanari, G. Oriolo, and S. Ivaldi, "A multimode teleoperation framework for humanoid locomotion manipulation: An application for the icub robot," *IEEE Robotics and Automation Magazine*, vol. 26, no. 4, pp. 73–82, 2019. 6
- [27] J. Ramos and S. Kim, "Dynamic locomotion synchronization of bipedal robot and human operator via bilateral feedback teleoperation," *Science Robotics*, vol. 4, no. 35, p. eaav4282, 2019. [Online]. Available: <https://www.science.org/doi/abs/10.1126/scirobotics.aav4282> 6
- [28] G. B. Margolis, G. Yang, K. Paigwar, T. Chen, and P. Agrawal, "Rapid locomotion via reinforcement learning," *arXiv preprint arXiv:2205.02824*, 2022. 6
- [29] A. Kumar, Z. Fu, D. Pathak, and J. Malik, "Rma: Rapid motor adaptation for legged robots," *arXiv preprint arXiv:2107.04034*, 2021. 6
- [30] Z. Fu, A. Kumar, J. Malik, and D. Pathak, "Minimizing energy consumption leads to the emergence of gaits in legged robots," *Conference on Robot Learning (CoRL)*, 2021. 6
- [31] A. Escontrela, X. B. Peng, W. Yu, T. Zhang, A. Iscen, K. Goldberg, and P. Abbeel, "Adversarial motion priors make good substitutes for complex reward functions. 2022 ieee," in *International Conference on Intelligent Robots and Systems (IROS)*, vol. 2, 2022. 6
- [32] Z. Li, X. Cheng, X. B. Peng, P. Abbeel, S. Levine, G. Berseth, and K. Sreenath, "Reinforcement learning for robust parameterized locomotion control of bipedal robots," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 2811–2817. 6
- [33] J. Siekmann, K. Green, J. Warila, A. Fern, and J. Hurst, "Blind bipedal stair traversal via sim-to-real reinforcement learning," *arXiv preprint arXiv:2105.08328*, 2021. 6
- [34] A. Agarwal, A. Kumar, J. Malik, and D. Pathak, "Legged locomotion in challenging terrains using egocentric vision," in *Conference on Robot Learning*. PMLR, 2023, pp. 403–415. 6
- [35] R. Yang, G. Yang, and X. Wang, "Neural volumetric memory for visual locomotion control," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 1430–1440. 6
- [36] G. B. Margolis, T. Chen, K. Paigwar, X. Fu, D. Kim, S. Kim, and P. Agrawal, "Learning to jump from pixels," *arXiv preprint arXiv:2110.15344*, 2021. 6
- [37] H. Duan, B. Pandit, M. S. Gadde, B. J. van Marum, J. Dao, C. Kim, and A. Fern, "Learning vision-based bipedal locomotion for challenging terrain," *arXiv preprint arXiv:2309.14594*, 2023. 6
- [38] Z. Zhuang, Z. Fu, J. Wang, C. Atkeson, S. Schwertfeger, C. Finn, and H. Zhao, "Robot parkour learning," in *Conference on Robot Learning (CoRL)*, 2023. 6
- [39] X. Cheng, K. Shi, A. Agarwal, and D. Pathak, "Extreme parkour with legged robots," *arXiv preprint arXiv:2309.14341*, 2023. 6
- [40] X. B. Peng, E. Coumans, T. Zhang, T.-W. Lee, J. Tan, and S. Levine, "Learning agile robotic locomotion skills by imitating animals," Apr. 2020. 6
- [41] Y. Wang, Z. Jiang, and J. Chen, "Amp in the wild: Learning robust, agile, natural legged locomotion skills," *arXiv preprint arXiv:2304.10888*, 2023. 6
- [42] Y. Fuchioka, Z. Xie, and M. Van de Panne, "Opt-mimic: Imitation of optimized trajectories for dynamic quadruped behaviors," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 5092–5098. 6

- [43] R. Yang, Z. Chen, J. Ma, C. Zheng, Y. Chen, Q. Nguyen, and X. Wang, “Generalized animal imitator: Agile locomotion with versatile motion prior,” *arXiv preprint arXiv:2310.01408*, 2023. [6](#)
- [44] Z. Li, X. B. Peng, P. Abbeel, S. Levine, G. Berseth, and K. Sreenath, “Reinforcement learning for versatile, dynamic, and robust bipedal locomotion control,” *arXiv preprint arXiv:2401.16889*, 2024. [6](#)
- [45] Q. Liao, B. Zhang, X. Huang, X. Huang, Z. Li, and K. Sreenath, “Berkeley humanoid: A research platform for learning-based control,” *arXiv preprint arXiv:2407.21781*, 2024. [6](#)
- [46] I. Radovovic, B. Zhang, B. Shi, J. Rajasegaran, S. Kamat, T. Darrell, K. Sreenath, and J. Malik, “Humanoid locomotion as next token prediction,” *arXiv:2402.19469*, 2024. [6](#)
- [47] X. Cheng, K. Shi, A. Agarwal, and D. Pathak, “Extreme parkour with legged robots,” 2023. [Online]. Available: <https://arxiv.org/abs/2309.14341> [6](#)
- [48] X. Cheng, A. Kumar, and D. Pathak, “Legs as manipulator: Pushing quadrupedal agility beyond locomotion,” in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023. [6](#)
- [49] H. Ha, Y. Gao, Z. Fu, J. Tan, and S. Song, “Umi on legs: Making manipulation policies mobile with manipulation-centric whole-body controllers,” *arXiv preprint arXiv:2407.10353*, 2024. [6](#)
- [50] T. He, Z. Luo, W. Xiao, C. Zhang, K. Kitani, C. Liu, and G. Shi, “Learning human-to-humanoid real-time whole-body teleoperation,” *arXiv preprint arXiv:2403.04436*, 2024. [6](#)
- [51] C. Zhang, W. Xiao, T. He, and G. Shi, “Wococo: Learning whole-body humanoid control with sequential contacts,” *arXiv e-prints*, pp. arXiv–2406, 2024. [6](#)
- [52] C. Lin, X. Liu, Y. Yang, Y. Niu, W. Yu, T. Zhang, J. Tan, B. Boots, and D. Zhao, “Locoman: Advancing versatile quadrupedal dexterity with lightweight loco-manipulators,” *arXiv preprint arXiv:2403.18197*, 2024. [6](#)
- [53] Y. Ji, G. B. Margolis, and P. Agrawal, “Dribblebot: Dynamic legged manipulation in the wild,” *arXiv preprint arXiv:2304.01159*, 2023. [6](#)
- [54] S. Jeon, M. Jung, S. Choi, B. Kim, and J. Hwangbo, “Learning whole-body manipulation for quadrupedal robot,” *arXiv preprint arXiv:2308.16820*, 2023. [6](#)
- [55] T. He, Z. Luo, W. Xiao, C. Zhang, K. Kitani, C. Liu, and G. Shi, “Learning human-to-humanoid real-time whole-body teleoperation,” in *arXiv*, 2024. [6](#)
- [56] Y. Ma, F. Farshidian, T. Miki, J. Lee, and M. Hutter, “Combining learning-based locomotion policy with model-based manipulation for legged mobile manipulators,” *RA-L*, 2022. [6](#)
- [57] G. Pan, Q. Ben, Z. Yuan, G. Jiang, Y. Ji, J. Pang, H. Liu, and H. Xu, “Roboduet: A framework affording mobile-manipulation and cross-embodiment,” *arXiv preprint arXiv:2403.17367*, 2024. [6](#)
- [58] M. Seo, S. Han, K. Sim, S. H. Bang, C. Gonzalez, L. Sentis, and Y. Zhu, “Deep imitation learning for humanoid loco-manipulation through human teleoperation,” in *2023 IEEE-RAS 22nd International Conference on Humanoid Robots (Humanoids)*. IEEE, 2023, pp. 1–8. [6](#)
- [59] J. Dao, H. Duan, and A. Fern, “Sim-to-real learning for humanoid box loco-manipulation,” in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 16930–16936. [6](#)
- [60] X. B. Peng, Z. Ma, P. Abbeel, S. Levine, and A. Kanazawa, “Amp: Adversarial motion priors for stylized physics-based character control,” *ACM Transactions on Graphics (ToG)*, vol. 40, no. 4, pp. 1–20, 2021. [6](#)
- [61] X. B. Peng, Y. Guo, L. Halper, S. Levine, and S. Fidler, “Ase: Large-scale reusable adversarial skill embeddings for physically simulated characters,” *ACM Trans. Graph.*, vol. 41, no. 4, Jul. 2022. [6](#)
- [62] C. Tessler, Y. Kasten, Y. Guo, S. Mannor, G. Chechik, and X. B. Peng, “Calm: Conditional adversarial latent models for directable virtual characters,” in *ACM SIGGRAPH 2023 Conference Proceedings*, ser. SIGGRAPH ’23. New York, NY, USA: Association for Computing Machinery, 2023. [Online]. Available: <https://doi.org/10.1145/3588432.3591541> [6](#)
- [63] M. Hassan, Y. Guo, T. Wang, M. Black, S. Fidler, and X. B. Peng, “Synthesizing physical character-scene interactions,” 2023. [Online]. Available: <https://doi.org/10.1145/3588432.3591525> [6](#)
- [64] Z. Luo, J. Cao, J. Merel, A. Winkler, J. Huang, K. M. Kitani, and W. Xu, “Universal humanoid motion representations for physics-based control,” in *The Twelfth International Conference on Learning Representations*, 2024. [Online]. Available: <https://openreview.net/forum?id=OrOd8PxOO2> [6](#)
- [65] H. Y. Ling, F. Zinno, G. Cheng, and M. Van De Panne, “Character controllers using motion vaes,” *ACM Transactions on Graphics (TOG)*, vol. 39, no. 4, pp. 40–1, 2020. [6](#)
- [66] H. Zhang, Y. Yuan, V. Makoviychuk, Y. Guo, S. Fidler, X. B. Peng, and K. Fatahalian, “Learning physically simulated tennis skills from broadcast videos,” *ACM Trans. Graph.*, vol. 42, no. 4, jul 2023. [Online]. Available: <https://doi.org/10.1145/3592408> [6](#)
- [67] J. Wang, J. Hodgins, and J. Won, “Strategy and skill learning for physics-based table tennis animation,” in *ACM SIGGRAPH 2024 Conference Papers*, 2024, pp. 1–11. [6](#)