# AdaptManip: Learning Adaptive Whole-Body Object Lifting and Delivery with Online Recurrent State Estimation

Morgan Byrd[1*], Donghoon Baek[1], Kartik Garg[1], Hyunyoung Jung[1],
Daesol Cho[1], Maks Sorokin[1], Robert Wright[2], and Sehoon Ha[1]

*Abstract*—This paper presents Adaptive Whole-body Loco-Manipulation, AdaptManip, a fully autonomous framework for humanoid robots to perform integrated navigation, object lifting, and delivery. Unlike prior imitation learning-based approaches that rely on human demonstrations and are often brittle to disturbances, AdaptManip aims to train a robust loco-manipulation policy via reinforcement learning without human demonstrations or teleoperation data. The proposed framework consists of three coupled components: (1) a recurrent object state estimator that tracks the manipulated object in real time under limited field-of-view and occlusions; (2) a whole-body base policy for robust locomotion with residual manipulation control for stable object lifting and delivery; and (3) a LiDAR-based robot global position estimator that provides drift-robust localization. All components are trained in simulation using reinforcement learning and deployed on real hardware in a zero-shot manner. Experimental results show that AdaptManip significantly outperforms baseline methods, including imitation learning-based approaches, in adaptability and overall success rate, while accurate object state estimation improves manipulation performance even under occlusion. We further demonstrate fully autonomous real-world navigation, object lifting, and delivery on a humanoid robot.

*Index Terms*—Humanoid, Learning-based whole-body loco manipulation, State estimation

## I. INTRODUCTION

Humanoid robots are a promising platform for human-centric environments due to their ability to execute human-like whole-body capabilities [1], [2]. However, achieving reliable whole-body loco-manipulation on humanoids remains fundamentally challenging. Such tasks require the robot to simultaneously coordinate high-dimensional whole-body dynamics, maintain balance under changing contact conditions, and regulate complex multi-contact interactions with external objects. While recent work [3], [4], [5], [6] has enabled dynamic whole-body skills such as jumping and parkour, autonomous contact-rich humanoid whole-body loco manipulation remains largely unsolved.

To address these challenges, recent advances in imitation learning—leveraging motion capture demonstrations, teleoperation, and large-scale human video datasets—have emerged as a powerful paradigm for learning whole-body humanoid control, which offers an alternative to traditional model-based approaches [6], [7], [8], [9]. While such methods have shown strong performance, they rely heavily on motion capture systems and curated demonstrations that provide privileged global robot and object state information [10], [3], [4], [5]. Moreover,
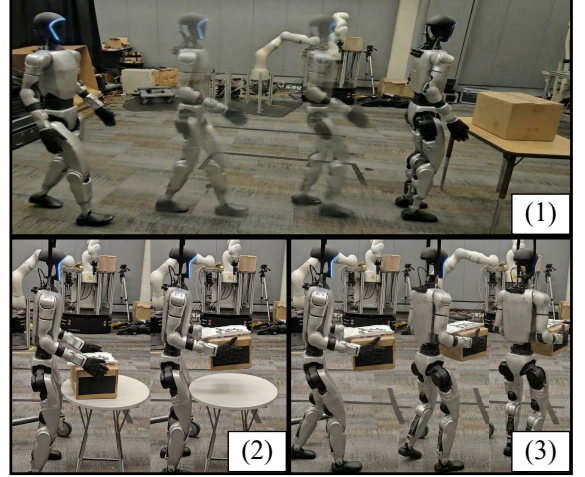


Fig. 1: **Fully autonomous humanoid loco-manipulation using online recurrent state estimation.** (1) navigating toward the object, (2) lifting the object through coordinated whole-body motion, and (3) delivering the object to the target location. Our method relies solely on onboard sensing and does not require teleoperation data or an external mocap system.

recovery and adaptability under natural failure cases, such as object slippage or drops, remain underexplored because these methods tend to follow the fixed reference motion for the given time window.

Our key insight for robust and adaptive loco-manipulation is to *jointly learn online object state estimation and control by training a recurrent state estimator alongside reinforcement learning that encounters and adapts to failure cases.* This formulation tightly couples perception and control, and allows the humanoid to continuously reason about object pose and contact evolution without relying on privileged global information or pre-recorded demonstrations. By integrating object state estimation into training, the robot naturally learns to adjust its whole-body coordination when interactions deviate from nominal execution, such as during object slippage or partial loss of contact. Inspired by human behavior, our design leverages complementary sensing modalities—vision and proprioception—to enable robust state inference even when visual observations are partial or occluded.

Motivated by this insight, we propose AdaptManip, a learning-based framework for whole-body humanoid loco-manipulation that jointly integrates online object state estimation and control to enable fully autonomous navigation, lifting, and delivery using only onboard sensing. By coupling multi-modal object state estimation from vision and proprioception with LiDAR-based robot pose odometry, AdaptManip achieves robust and recovery-capable loco-manipulation—including re-

[1] Georgia Institute of Technology, Atlanta, GA, 30308, USA
[2] Georgia Tech Research Institute, Atlanta, GA, 30308, USA
*Correspondence to abyrd45@gatech.edu

TABLE I: Comparison of representative methods across key aspects: onboard-only sensing (Onbd), absence of human demonstrations (NoHumRef), locomotion–manipulation capability (LocoMan), absence of future references (NoFutRef), and no teleoperation (NoTeleOp).

| Method | Onbd | NoHumRef | LocoMan | NoFutRef | NoTeleOp |
|---|---|---|---|---|---|
| TWIST [7] | ✗ | ✗ | ✓ | ✓ | ✓ |
| ResMimic [4] | ✗ | ✗ | ✓ | ✓ | ✓ |
| VisualMimic [3] | ✓ | ✗ | ✓ | ✓ | ✗ |
| HDMI [10] | ✗ | ✗ | ✓ | ✓ | ✓ |
| PhysHSI [11] | ✗ | ✗ | ✓ | ✓ | ✓ |
| OmniRetarget [5] | ✗ | ✗ | ✓ | ✓ | ✓ |
| GMT [6] | ✓ | ✗ | ✓ | ✗ | ✓ |
| BoxLocoManip [12] | ✓ | ✓ | ✓ | ✓ | ✗ |
| **Ours** | ✓ | ✓ | ✓ | ✓ | ✓ |

grasping after object drops—without relying on motion capture systems or teleoperation. See Table I for a qualitative comparison to existing methods. To realize these capabilities, our framework integrates three key components into a unified system: (1) a reinforcement learning-based locomotion policy for stable bipedal mobility, (2) a residual upper-body manipulation policy for contact-rich object interaction, and (3) a fully onboard object state estimator that provides real-time perception for control. Concretely, the robot operates through three coordinated stages: navigation, lifting, and delivery. During the navigation stage, the humanoid approaches the target object using LiDAR-based robot pose odometry and proprioceptive feedback, enabling fully onboard localization. In the lifting stage, an online object state estimator guides grasping and coordinated whole-body manipulation. In the delivery stage, the robot transports the object to the target location while maintaining balance under changing contact conditions. Overall, AdaptManip achieves higher task success rates than prior baselines by learning robust and adaptive behaviors through reinforcement learning, including recovery actions under failure conditions. This is realized in a fully autonomous system through the integration of onboard sensing and recurrent state estimation.

We summarize our key contributions as follows. First, we introduce AdaptManip, a learning-based framework for whole-body humanoid loco-manipulation that autonomously accomplishes navigation, lifting, and delivery through a structured three-stage strategy. Second, we develop an online, recurrent object state estimation module that fuses LiDAR, vision, and proprioceptive sensing, enabling robust and recovery-capable loco-manipulation using only onboard sensors, without teleoperation data or external motion capture systems. Finally, we validate the effectiveness of AdaptManip through extensive simulation studies and real-world experiments on physical humanoid hardware.

## II. RELATED WORK

### A. Humanoid Whole-Body Control

Over the past decades, model-based control, such as Model Predictive Control, has advanced significantly across a wide range of humanoid platforms, including quadrupedal, bipedal, and wheeled robots [13], [14], [15], [16]. These approaches are typically developed through hierarchical and decomposed control architectures that separate trajectory planning and whole-body control, often integrating model-based predictive optimization at different levels and time scales [17], [18], [19]. A key advantage of this paradigm is its reliance on physics-based models, which provide strong interpretability, stability guarantees, and smooth, dynamics-aware continuous control actions. However, the performance is inherently sensitive to assumptions and often requires manual human effort to model accurate loco-manipulation behaviors.

In light of these challenges, learning-based methods —particularly reinforcement learning (RL)— have become increasingly influential in humanoid robotics, supporting the synthesis of complex whole-body behaviors [20]. End-to-end RL approaches, often augmented with domain randomization, have shown encouraging sim-to-real transfer and demonstrated successful deployment on hardware platforms [21], [22], [23], [24], [25]. However, reliable and fully autonomous object-centric whole-body manipulation remains challenging, as physical interaction with diverse objects entails complex, task-dependent dynamics that are difficult to model and generalize.

### B. Learning Humanoid Loco-Manipulation

Learning-based methods have recently advanced from isolated locomotion and manipulation to integrated whole-body humanoid loco-manipulation [9], [6], [26], [27]. This progress has been accelerated by the availability of large-scale open-source motion datasets (e.g., AMASS [28] and LAFAN1 [29]), together with advances in imitation learning, which enable humanoid robots to reproduce natural human motions such as jumping, running, dancing, and kicking [6], [26], [30], [27], [31]. Despite these successes, existing imitation-based humanoid controllers largely emphasize kinematic motion reproduction, with limited ability to handle object contacts and interaction dynamics.

Recent works have begun to address object-centric whole-body humanoid loco-manipulation, where robots must jointly reason about physical interaction with objects. Existing approaches explore this problem using motion imitation with external tracking [10], [4], vision-based reinforcement learning [3], or hybrid perception pipelines combining long-range sensing and close-range visual feedback [11]. While these methods demonstrate promising results on tasks such as box lifting, many rely on strong assumptions, including motion capture supervision, human-in-the-loop navigation or skill transitions [12], and continuous visual access to the target object. Unlike existing works, our approach relies solely on fully onboard sensing and enables fully autonomous, robust, and adaptive whole-body behaviors to accomplish contact-rich manipulation tasks.

### C. Object State Estimation

Accurate perception and state estimation are critical components of humanoid loco-manipulation. Motion capture systems are commonly used to obtain accurate global robot and object poses in laboratory environments [10], [4]. However, such systems are inherently restricted to controlled environments and are impractical for deployment beyond the lab. Visual pose estimation offers a more flexible alternative, either

through fiducial markers [32] or direct pixel-based methods [33], enabling operation in less structured settings. Despite their effectiveness, vision-based approaches typically require continuous and reliable visual observations, making them vulnerable to failures caused by occlusions, limited view frustums, or dynamic viewpoints during whole-body motion. To address these limitations, we rely exclusively on fully onboard sensing—camera, LiDAR, and robot proprioception—to, similar to [34], recurrently estimate object state, robot–object contact forces, and the robot's global pose, and deploy the resulting system directly on hardware for fully autonomous loco-manipulation.

## III. TASK DESCRIPTION

To robustly complete our whole-body loco-manipulation tasks, we decompose it into three stages: (1) moving the robot from an initial pose to the box, (2) grasping and lifting the box, and (3) transporting it to the target location. Each stage is designed to address the distinct requirements (Fig. 2).

**Navigation.** In the first stage, the humanoid navigates from its initial location to a goal location in front of the target object, which is given as input. Therefore, this stage requires locomotion rather than object manipulation skills. Because the object is initially outside the camera's field of view, the relative object location is estimated using LiDAR-inertial odometry based on the onboard LiDAR sensor, implemented with FAST-LIO [35].

**Grasping and Lifting.** When the robot approaches sufficiently close to the object ($\leq$ 0.5m), it transitions to the second stage to grasp and lift the target object. The robot employs unarticulated rubber hands; therefore, the object is supported purely through frictional contact rather than form closure. In this phase, vision-based sensing is activated to refine the object pose for accurate grasping and lifting. We implement the vision-based sensing using an AprilTag [32], although it can be replaced with any vision-based 6D object pose estimator. An effective policy for this stage must be robust to inaccurate or unavailable 6D object pose information.

**Carrying to Destination.** Once the object is securely lifted, the system transitions to the third stage, which involves transporting the object from its initial location to the target location. During this phase, the object is frequently occluded by the robot's body and hands and visual information becomes unreliable.

## IV. LEARNING LOCO-MANIPULATION WITH RECURRENT STATE ESTIMATION

We present a learning-based humanoid loco-manipulation policy with a recurrent object state estimator to enable robust object grasping, lifting, and delivery under unreliable visual observations. The proposed policy allows a humanoid robot to adaptively manipulate an object by integrating onboard visual and proprioceptive information. However, learning such a policy is inherently difficult, as humanoid loco-manipulation requires simultaneously maintaining stable bipedal locomotion, interpreting noisy and intermittent visual inputs, and
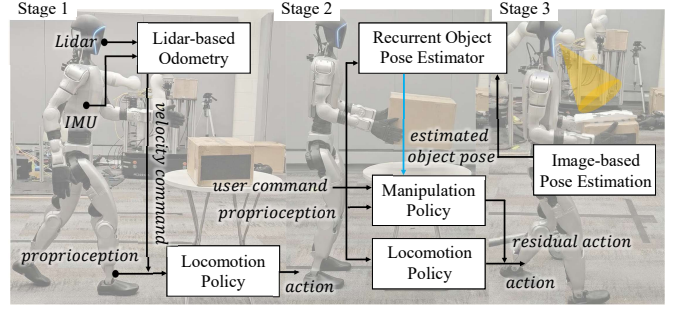


Fig. 2: **Three-stage AdaptManip experiment plan and deployment.** Stage 1: LiDAR odometry and proprioception enable autonomous navigation. Stage 2: Recurrent multimodal object-pose estimation supports coordinated lifting. Stage 3: Image-based refinement and residual policies ensure stable delivery. All stages operate using only onboard sensing.

executing adaptive whole-body manipulation under contact-rich interactions. This challenge substantially increases the complexity of policy learning and reduces the effectiveness of a naive end-to-end approach.

To address these challenges, we structure the learning process into three components. First, we train a base whole-body locomotion policy that provides stable and periodic bipedal walking. Then, we learn a whole-body residual manipulation policy on top of the base policy, which enables adaptive object grasping and lifting while preserving locomotion stability. Concurrently, we train a recurrent object state estimator that fuses visual observations and proprioceptive signals to infer the object pose online, which aims to achieve robust manipulation under partial or missing visual inputs, similar to [13]. The details of each stage are described in the following subsections.

### A. Base Whole-Body Locomotion Policy

For robust whole-body walking, we train a base locomotion policy using RL to generate stable bipedal locomotion across a range of commanded velocities. This policy serves as a fixed foundation for subsequent whole-body manipulation learning.
**Observation and Action Space.** We employ an asymmetric actor–critic architecture during training. The actor observation at time $t$ is defined as

$$\mathbf{o}_t^{\text{actor}} = \left[ \, \boldsymbol{\theta}_j, \, \dot{\boldsymbol{\theta}}_j, \, \boldsymbol{\omega}, \, \mathbf{g}_{\text{proj}}, \, \bar{\mathbf{v}}, \, \bar{\omega}_z, \, \bar{h}, \, \mathbf{a}_{t-1} \, \right],$$

where $\boldsymbol{\theta}_j \in \mathbb{R}^{29}$ and $\dot{\boldsymbol{\theta}}_j \in \mathbb{R}^{29}$ denote the joint positions and velocities, $\boldsymbol{\omega} \in \mathbb{R}^3$ is the base angular velocity, $\mathbf{g}_{\text{proj}} \in \mathbb{R}^3$ is the gravity vector projected into the torso frame, $\bar{\mathbf{v}} = [\bar{v}_x, \bar{v}_y]^\top \in \mathbb{R}^2$ denotes the commanded planar velocity of the base, $\bar{\omega} \in \mathbb{R}$ is the commanded base yaw rate, $\bar{h} \in \mathbb{R}$ is the commanded base height, and $\mathbf{a}_{t-1} \in \mathbb{R}^{29}$ denotes the action applied at the previous timestep. The critic receives the same observations as the actor, augmented with the base linear velocity $\mathbf{v}_t \in \mathbb{R}^3$, and is provided with a three-step temporal history of all inputs, i.e., $\mathbf{o}_t^{\text{critic}} = [\, \mathbf{o}_{t-2:t}^{\text{actor}}, \, \mathbf{v}_{t-2:t} \,]$.

The actions for the policy, $\mathbf{a}_t \in \mathbb{R}^{29}$, are joint-space position targets relative to a default configuration which are mapped to joint torques via a PD controller, $\boldsymbol{\tau}_t = \mathbf{k}_p\big((\mathbf{a}_t + \mathbf{q}_{\text{def}}) - \mathbf{q}_t\big) - \mathbf{k}_d \dot{\mathbf{q}}_t$.
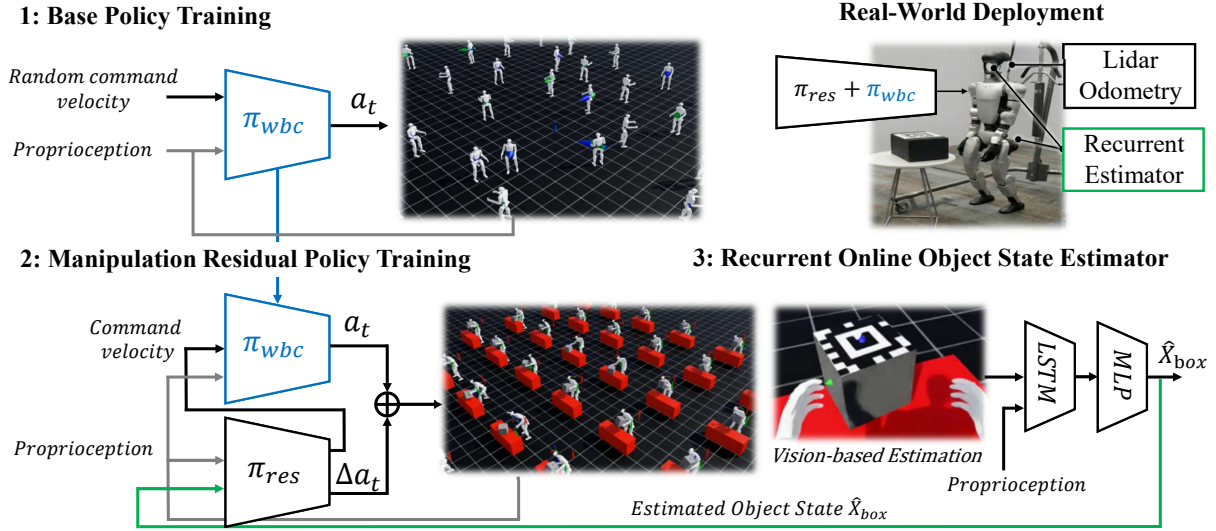
Fig. 3: **Overview of the training and deployment pipeline.** (1) A base whole-body control policy $\pi_{\mathrm{wbc}}$ is trained in IsaacLab to generate base whole-body behavior such as walking. (2) A manipulation residual policy $\pi_{\mathrm{res}}$ is trained on top of the base policy, taking proprioception and the estimated object state $\hat{X}_{\mathrm{box}}$ to produce residual actions $\Delta a_t$. The residual action aims to adaptively lift a 3D object. (3) A recurrent online object state estimator fuses vision and proprioceptive cues using a V-LSTM and MLP to infer $\hat{X}_{\mathrm{box}}$, and is trained jointly with the residual manipulation policy. During real-world deployment, the robot uses onboard estimators and LiDAR odometry and executes the combined policies $\pi_{\mathrm{wbc}}$ and $\pi_{\mathrm{res}}$ to complete the whole-body loco-manipulation task.

**Reward Design.** The locomotion reward follows a weighted structure consisting of command tracking, gait shaping, motion regularization, and constraint violation penalties, as summarized in Eq. (1), following prior work [36].

$$
\begin{aligned}
r_{\mathrm{loco}} = & \underbrace{\omega_{\mathrm{tr}}\left(e^{-\|v-\bar{v}\|} + e^{-\|\omega-\bar{\omega}\|} + e^{-|z-\bar{z}|}\right)}_{\text{Command tracking}} \\
& + \underbrace{\omega_{\mathrm{gait}}\left(\sum_f \mathbb{I}_{v>0.1}(t_f - 0.4) + e^{-0.5\sum_f |z_f - 0.05|}\right)}_{\text{Gait shaping}} \\
& - \underbrace{\omega_{\mathrm{reg}}\sum_j \left(\tau_j^2 + \omega_j^2 + \dot{\omega}_j^2 + a_j^2 + \dot{a}_j^2 + \ddot{a}_j^2\right)}_{\text{Motion regularization}} \\
& - \underbrace{\omega_{\mathrm{vio}}\left(\sum_f \mathbb{I}_c\|v_f\| + \sum_f v_r f_c + \|g_p\|^2 + \omega_o + \tau_o\right)}_{\text{Constraint violation penalties}}.
\end{aligned}
\tag{1}
$$

Here, $v$, $\omega$, and $z$ denote the base linear velocity, angular velocity, and height, with corresponding commands $\bar{v}$, $\bar{\omega}$, and $\bar{z}$. The index $j$ denotes joints and $f$ denotes feet. Terms $\tau_j$, $\omega_j$, and $\dot{\omega}_j$ are joint torque, velocity, and acceleration, while $a_j$ is the policy action. $\dot{a}_j$ and $\ddot{a}_j$ denote the first- and second-order finite differences of actions. $v_f$ and $f_c$ are the foot velocity and contact force, $t_f$ is the foot air-time, and $z_f$ is the foot height. $g_p$ denotes the gravity vector projected onto the body frame. $\omega_o$ and $\tau_o$ indicate joint velocity and torque limit violations. $\mathbb{I}_c$ is an indicator function for foot contact.

### B. Whole-Body Residual Manipulation Policy

For whole-body loco-manipulation, we train a **residual** policy on top of the frozen base policy. This residual policy

learns task-specific adaptations for object grasping, lifting, and stabilization.

**Observation and Action Space.** The actor receives

$$
\mathbf{o}_t^{\mathrm{actor}} = \left[\boldsymbol{\theta}_j,\ \dot{\boldsymbol{\theta}}_j,\ \boldsymbol{\omega},\ \mathbf{g}_{\mathrm{proj}},\ \tilde{\mathbf{X}}_{\mathrm{box}},\ \bar{\mathbf{X}}_{\mathrm{box}},\ \mathbf{a}_{t-1}\right],
$$

which includes the same proprioceptive state as the base locomotion policy (joint positions $\boldsymbol{\theta}_j$, velocities $\dot{\boldsymbol{\theta}}_j$, base angular velocity $\boldsymbol{\omega}$, and projected gravity $\mathbf{g}_{\mathrm{proj}}$), augmented with the estimated 6D box pose $\tilde{\mathbf{X}}_{\mathrm{box}}$, the commanded box pose $\bar{\mathbf{X}}_{\mathrm{box}}$, and the previous action $\mathbf{a}_{t-1}$. As in the locomotion controller, we employ an asymmetric actor–critic architecture. The critic augments the actor observations with privileged information:

$$
\mathbf{o}_t^{\mathrm{critic}} = \left[\mathbf{o}_{t-2:t}^{\mathrm{actor}},\ \mathbf{o}_{t-2:t}^{\mathrm{priv}}\right],
$$

where $\mathbf{o}_t^{\mathrm{priv}}$ denotes privileged information available only during training, including the ground-truth 6D pose $\mathbf{X}_{\mathrm{box}}$, the linear velocity $\mathbf{v}_{\mathrm{box}}$, the angular velocity $\omega_{\mathrm{box}}$, as well as hand and box contact forces $\mathbf{f}_{\mathrm{hand}}$ and $\mathbf{f}_{\mathrm{box}}$.

The action space consists of two components. First, the command vector provided to the frozen low-level locomotion policy is $\bar{\mathbf{u}}_{\mathrm{loco}} = [\bar{v}_x, \bar{v}_y, \bar{\omega}, \bar{h}] \in \mathbb{R}^4$, which corresponds to the base motion and height commands defined in (IV-A). These commands regulate planar locomotion, including forward and lateral motion, turning, and base height control. Second, the upper-body action $a_{\mathrm{upper}} \in \mathbb{R}^{17}$ corresponds to residual PD joint position targets for the upper body, primarily affecting the waist and arm joints.

**Reward Design.** We use the locomotion reward $r_{\mathrm{loco}}$ defined in Eq. (1) and augment it with manipulation-specific objectives for robust bimanual grasping and stable box transport. Beyond the base locomotion reward in Eq. (1), the proposed reward

introduces additional manipulation-specific objectives that explicitly account for physical interaction with the object during bimanual grasping and transport. In particular, the contact-related terms penalize excessive relative motion between the robot and the box, encourage symmetric bimanual contact forces, reward proper hand–box contact orientation, penalize failures to establish contact, and discourage tangential hand motion indicative of slipping:

$$
\begin{aligned}
r = {} & r_{\text{loco}} \\
& + \underbrace{\omega_{\text{kin}}\Big(e^{-|\psi_{\text{robot}} - \psi_{\text{box}}|} + e^{-4\|p_{\text{hand}}^{\text{err}}\|} + e^{-1.5\|p_{\text{root}}^{\text{err}}\|}\Big)}_{\text{Kinematic tracking}} \\
& + \underbrace{\omega_{\text{box}}\Big(e^{-2\|p_{\text{box}} - p_{\text{des}}\|_1 - \|q_{\text{box}} - q_{\text{des}}\|_1} + e^{-\|v_{\text{root}} - v_{\text{box}}\|_2}\Big)}_{\text{Box stabilization}} \\
& + \underbrace{\omega_{\text{con}}\,\text{clamp}\Big(\sum_h \|f_{\text{con},h}\|\,\mathbb{I}_{\text{box}},\, 0,\, 1\Big)}_{\text{Contact force quality}} \\
& - \underbrace{\omega_{\text{con}} \sum_h \min\big(0,\, v_{\text{hand},z} - v_{\text{box},z}\big)}_{\text{Slip avoidance}} .
\end{aligned}
\tag{2}
$$

**Terminal Condition.** Episodes terminate after 20 seconds, if the robot tilts more than $60°$, if the root height drops below 0.15m, or if the box falls below 0.25m.

### C. Recurrent Object State Estimation

Accurate and robust object state estimation is essential for whole-body loco-manipulation, as manipulation performance directly depends on reliable object pose information. Although motion capture systems provide accurate measurements, they are restricted to laboratory settings, making vision-based perception essential for real-world operation. In this work, we employ a robust tag-based visual pose estimator [32]; however, visual observations are often incomplete or intermittent due to occlusions, limited camera field of view, and object motion during manipulation. Enforcing constant object visibility to mitigate these issues can induce unnatural behaviors.

We propose an online object state estimation approach that fuses visual observations with proprioceptive measurements and executed actions. The estimator follows a human-inspired strategy, using vision primarily for grasp initiation and maintaining the object state through proprioception during manipulation. By leveraging proprioceptive and action histories, the estimator remains reliable under partial or intermittent visual feedback.

We employ a recurrent object state estimator to infer the relative object pose $\tilde{\mathbf{X}}_{\text{box}}$ online during manipulation. The estimator takes as input the current vision-based pose measurement (set to zero when unavailable), robot proprioceptive observations, and the executed action, and outputs the estimated 6D pose in the robot frame. We implement the estimator using an LSTM with an MLP output head, as its internal memory enables robust state propagation under missing or unreliable visual observations.

The estimator is trained concurrently with the manipulation policy using supervised learning with the ground-truth object

| Parameter | Range | Operation |
|---|---|---|
| Base Mass [kg] | [-2.5, 2.5] | Add |
| $k_p$ | [ 0.8, 1.2] | Scale |
| $k_d$ | [ 0.8, 1.2] | Scale |
| Ground Static Friction | [ 0.3, 1.5] | Absolute |
| Ground Dynamic Friction | [ 0.3, 0.9] | Absolute |
| Base Force Disturbance [N] | [-4.0, 4.0] | Absolute |
| Base Torque Disturbance [Nm] | [-2.0, 2.0] | Absolute |
| Table Static Friction | [ 0.3, 1.3] | Absolute |
| Table Dynamic Friction | [ 0.3, 1.5] | Absolute |
| Table Restitution | [ 0.0, 0.5] | Absolute |
| Box Static Friction | [ 0.3, 1.3] | Absolute |
| Box Dynamic Friction | [ 0.3, 1.5] | Absolute |
| Box Restitution | [ 0.0, 0.5] | Absolute |
| Box Mass [kg] | [-0.88, 1.5] | Add |
| Box Scale $x$ | [ 0.75, 1.25] | Scale |
| Box Scale $y$ | [ 0.75, 1.25] | Scale |
| Box Center of Mass $x$ | [ 0.75, 1.25] | Add |
| Box Center of Mass $y$ | [ 0.75, 1.25] | Add |
| Box Center of Mass $z$ | [ 0.75, 1.25] | Add |

TABLE II: Domain randomization parameters for policy training.

poses available in simulation. To stabilize early training, we apply a curriculum that gradually replaces the ground-truth pose with the estimated pose in the policy input. Specifically, the pose provided to the policy is defined as

$$
\mathbf{X}_{\text{in}} = w\,\tilde{\mathbf{X}}_{\text{box}} + (1 - w)\,\mathbf{X}_{\text{box}}, \tag{3}
$$

where the weighting factor is given by $w = \min(t/T, 1)$ with $t$ denoting the current training iteration and $T$ the maximum iteration.

To improve training efficiency, we do not directly incorporate visual inputs during RL training. Instead, we provide the policy with a noisy and randomly masked ground-truth object pose to model visual estimation errors and occlusions, which transfers robustly to real hardware.

### D. Domain Randomization

We employ domain randomization [37] during training to improve robustness and sim-to-real transfer by randomizing physical and control parameters, including base mass, ground friction, PD gains, and external disturbances (see Table II). For whole-body manipulation, we further introduce grasp-specific randomizations, such as box–table friction and restitution, box mass, scale, and center-of-mass location. Observation noise is injected for both the locomotion and manipulation policies to enhance robustness.

## V. EXPERIMENTAL RESULTS

In this section, we design simulation and hardware experiments to address the following research questions: (1) Can the proposed method manipulate objects more robustly compared to the baselines? (2) Can the learned object state estimator provide accurate pose estimates? (3) Can our policy be effectively transferred to a real humanoid robot?

### A. Implementation Details and Experimental Setup

*1) Training Details:* All policies are trained using PPO [38], augmented with a bilateral symmetry loss [39]. The base locomotion and whole-body manipulation policies

are trained separately, each for approximately one day on a single NVIDIA RTX 4090 GPU. The actor and critic are three-layer MLPs with hidden dimensions [512, 256, 128] and ELU activations. A recurrent object state estimator, implemented as an LSTM with a hidden dimension of 128, is trained jointly with the policy.

*2) Simulation Setup:* We conduct simulation experiments using two physics engines, IsaacLab [40] and MuJoCo [41], to enable cross-simulator validation. All experiments use a control frequency of 50 Hz, with the physics simulation running at 200 Hz. Policies are trained in IsaacLab and evaluated in MuJoCo without additional fine-tuning.

*3) Hardware Setup:* For real-world evaluation, we deploy on a Unitree G1 humanoid robot [42]. The robot is equipped with an Intel RealSense D435i RGB-D camera [43] and a Livox Mid-360 LiDAR [44]. All sensing and control are performed onboard, without reliance on external motion capture systems. Using a single AprilTag [32] placed at the center of the top surface of the box, we perform visual object pose estimation using the official Python bindings [45]. The policy is deployed on hardware without additional tuning following sim-to-sim validation, demonstrating effective sim-to-real transfer.

### B. Simulation Experiments

We first conducted comprehensive simulation experiments to demonstrate the effectiveness of AdaptManip, and compared it against the following baselines.

- **Pure RL.** The policy is trained with RL with the ground-truth 6D object pose, which corresponds to removing the estimator from our framework.
- **Pure RL + FK.** In addition to Pure RL, the policy is provided with the hand positions computed via forward kinematics, providing additional context.
- **Imitation Learning (IL).** The policy follows a predefined grasping motion, similar to recent motion-based methods [46].
- **AdaptManip (Ours).** Our full method combines an RL-based manipulation policy with a recurrent object state estimator.
- **Oracle.** The policy is identical to Pure RL but is given perfect ground-truth object state information at test time, providing an upper bound on achievable performance.

All policies were trained in IsaacLab without using vision, since training with visual inputs is computationally expensive and typically leads to weaker sim-to-real generalization. During evaluation, the object pose was obtained via an AprilTag in both IsaacLab and MuJoCo except for **Oracle**. MuJoCo was severed as an unseen simulator to perform sim-to-sim transfer evaluation and assess generalization beyond the training environment. We decomposed the task into three stages: navigation, grasping, and carrying, as illustrated in Fig. 2, and analyzed the results individually.

Overall, our results showed that **AdaptManip** not only achieved competitive performance in IsaacLab (the training environment) but also outperformed the baselines in the unseen MuJoCo environment (Table III). In IsaacLab, **AdaptManip**

TABLE III: Performance comparison between our method and the GT-Pose Oracle baseline in the no-occlusion setting across 135 trials in MuJoCo and IsaacLab.

| Method | Whole | Stage1 | Stage2 | Drops (↓) | Regrasps (↑) |
|---|---|---|---|---|---|
| **IsaacLab** | | | | | |
| Pure RL | 0.62 (±0.48) | 0.98 (±0.14) | 0.88 (±0.32) | 1.79 (±2.39) | 5.74 (±3.91) |
| Pure RL + FK | 0.88 (±0.32) | 0.93 (±0.25) | 0.92 (±0.27) | 0.29 (±0.89) | 2.94 (±1.88) |
| Imitation Learning (IL) | 0.42 (±0.46) | 0.96 (±0.19) | 0.93 (±0.18) | 3.81 (±4.77) | 2.94 (±1.99) |
| AdaptManip (Ours) | 0.85 (±0.35) | 0.97 (±0.17) | 0.92 (±0.26) | 0.49 (±1.14) | 2.12 (±1.95) |
| Oracle | 0.91 (±0.28) | 0.98 (±0.14) | 0.97 (±0.15) | 0.46 (±1.16) | 2.47 (±1.48) |
| **MuJoCo** | | | | | |
| Pure RL | 0.37 (±0.48) | 0.84 (±0.36) | 0.80 (±0.40) | 2.11 (±1.01) | 4.20 (±3.78) |
| Pure RL + FK | 0.61 (±0.49) | 0.95 (±0.22) | 0.93 (±0.26) | 2.44 (±1.29) | 5.24 (±4.53) |
| Imitation Learning (IL) | 0.00 (±0.00) | 0.93 (±0.26) | 0.69 (±0.46) | 0.82 (±0.44) | 1.78 (±0.93) |
| AdaptManip (Ours) | 0.75 (±0.43) | 0.90 (±0.30) | 0.88 (±0.32) | 1.94 (±0.68) | 6.32 (±3.95) |
| Oracle | 0.79 (±0.41) | 0.98 (±0.12) | 0.97 (±0.16) | 2.16 (±1.01) | 6.37 (±4.94) |


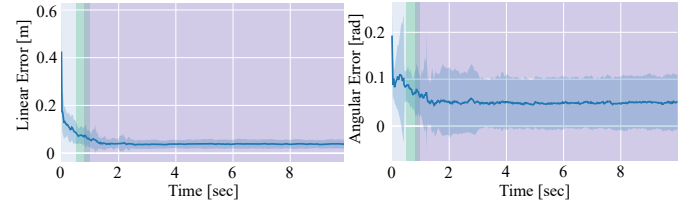
Fig. 4: State estimation error of our method. Shows mean ± 1 standard deviation across 50 episodes. The green region shows the area where vision is available, and the purple region shows the area where there is contact between the robot and box.

achieved an 85% success rate, which was comparable to **Pure RL + FK** and significantly better than **Pure RL** and the **Imitation Learning** baseline. Interestingly, **Pure RL + FK** achieved a success rate of 88%, substantially outperforming **Pure RL** at 62%, which highlighted the importance of informative state representations in reinforcement learning. In our experiments, **Imitation Learning** did not perform effectively, as it lacked the flexibility required to adapt to diverse interaction scenarios. Overall, the performance gap among **AdaptManip**, **Pure RL + FK**, and **Oracle** was relatively small in this setting, indicating that all three methods achieved near-optimal performance when reliable object state information was available.

However, when we evaluated sim-to-sim transfer, the differences became much more pronounced. **AdaptManip** achieved a 75% success rate, which was comparable to the 79% of **Oracle**, while also exhibiting a large number of regrasps (6.32), indicating its ability to actively recover from grasping failures. In contrast, **Pure RL + FK** showed a significantly lower success rate of 61%, which implied that simply providing end-effector positions was insufficient, as slippage between the hands and the object could not be properly captured. The other two methods, **Pure RL** and **Imitation Learning**, demonstrated relatively lower success rates, due to the lack of reliable object state information in the unseen environment.

We evaluate our policy on unseen objects in IsaacLab under a zero-shot setting. The policy achieves comparable performance across cylinders with different starting orientations (X-axis: 82 ± 14, Y-axis: 86 ± 8, Z-axis: 78 ± 11), while the performance is lower on the sphere (51 ± 21); nevertheless, the result still indicates encouraging generalization to significantly different object geometries.
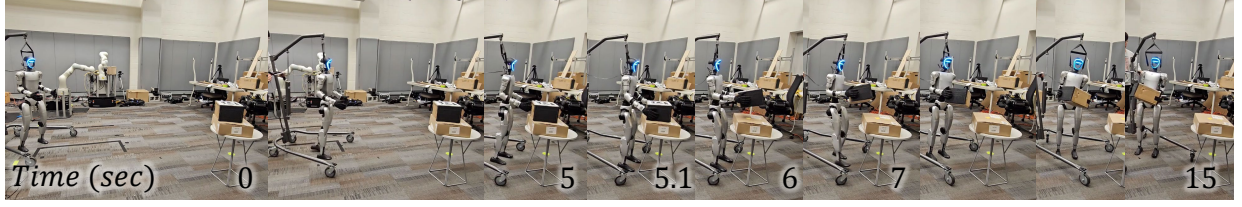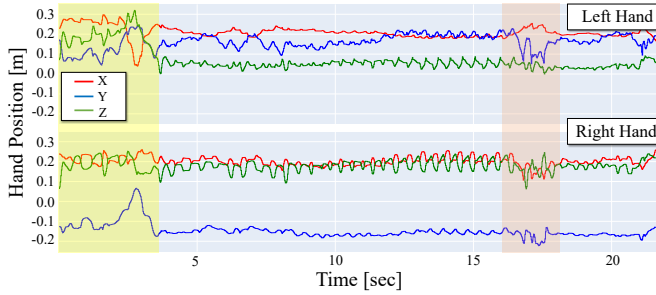
(a) Snapshots of Stage 1 (navigation to the box) and Stage 2 (lifting) during the whole-body loco-manipulation task.
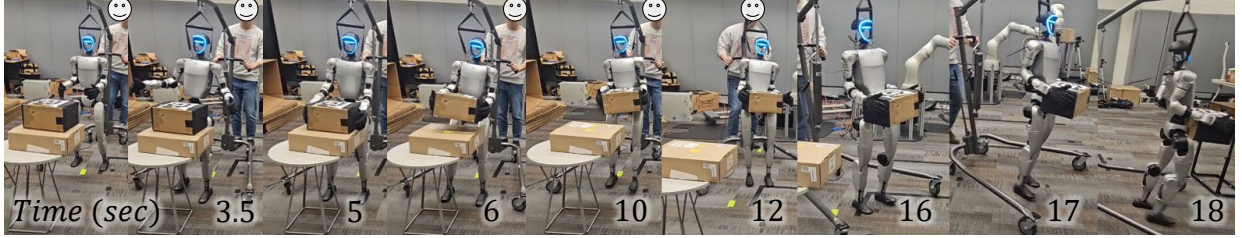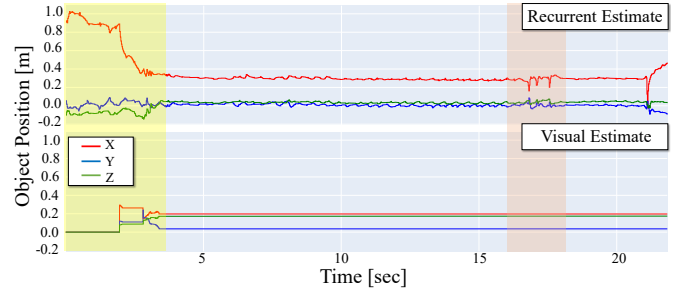


(b) Snapshots illustrating approaching, lifting, and delivery.

Fig. 5: Hardware demonstration of the three-stage whole-body loco-manipulation task.



(a) Left and right hand position trajectories during grasping and lifting.



(b) Object position estimates from the recurrent estimator and visual detection.

Fig. 6: Hardware demonstration of whole-body loco-manipulation. The yellow region indicates the grasp formation phase where the robot carefully coordinates its arms for a secure hold, while the red region highlights the robot's ability to recover from a transient loss of stability through corrective arm motions.

## C. Validation of Object State Estimator

We conduct further experiments to validate the effectiveness of our learned estimator. For this, we collect 50 episodes of data using our policy under the same randomized conditions used in Table II. For each episode, we collected the ground truth pose information of the box and the input observation information for the estimator, along with flags showing whether the box was visible in the robot camera frame or whether the robot was contacting the box.

We supplied this validation data to our estimator and predicted the pose information. The linear and angular pose errors are shown in Figure 4. From this, it is clear that initially, there is a relatively large error which is driven down when the AprilTag goes within the FOV of the camera. After that, the error stayed low for the rest of the episode even as the box was carried around due to the information provided from proprioception, which works very well as long as the robot maintains solid contact with the box.

## D. Hardware Experiments

Finally, we deploy the learned policy on a real Unitree G1 humanoid. As shown in Fig. 5, our end-to-end, fully autonomous policy completes the task using only onboard sensing and object state estimation. In the grasp formation phase (yellow; Fig. 6(a)), the robot carefully coordinates its arms to ensure a secure bimanual grasp before lifting. After grasping, the hand positions remain nearly fixed. This indicates that the robot maintains a stable hold while lifting and transporting the object. Near the later phase of the trial (red), a transient loss of grasp stability occurs and the robot recovers the grasp with a small corrective arm motion.

The object position estimates highlight the advantage of the proposed estimator (see Fig. 6(b)). The visual estimate degrades when the object leaves the camera field of view during floating-base motion (e.g., walking). In contrast, our recurrent estimator continues to track the object. This is enabled by jointly leveraging robot proprioception and policy actions, even when visual observations are intermittent.

Overall, these results confirm that our end-to-end policy transfers effectively to real hardware in a zero-shot manner, enabling robust whole-body loco-manipulation using only onboard sensing and object state estimation.

## VI. CONCLUSION

This paper presents a novel framework for completing whole-body, humanoid loco-manipulation tasks. We introduce AdaptManip, a method which combines multi-modal inputs of LiDAR, vision, and proprioception to maintain a recurrent

belief of the box pose and hierarchical RL in order to effectively learn a policy which utilizes the pose for picking up and carrying a box from an initial position to a target location. While we show good results for this box lifting task, interesting future work could include trying more extended tasks, incorporating additional sensor modalities for better object state estimation, or using articulated hands for better manipulation.

## REFERENCES

[1] B. Dynamics, "Atlas," 2024, accessed: 2025-04-07. [Online]. Available: https://bostondynamics.com/atlas/

[2] B. Insider. (2024) People are questioning if a video of tesla's humanoid robot folding a shirt is real. Accessed: 2025-04-07. [Online]. Available: https://www.businessinsider.com/reactions-tesla-robot-folding-shirt-question-video-real-2024-1

[3] S. Yin, Y. Ze, H.-X. Yu, C. K. Liu, and J. Wu, "Visualmimic: Visual humanoid loco-manipulation via motion tracking and generation," *arXiv preprint arXiv:2509.20322*, 2025.

[4] S. Zhao, Y. Ze, Y. Wang, C. K. Liu, P. Abbeel, G. Shi, and R. Duan, "Resmimic: From general motion tracking to humanoid whole-body loco-manipulation via residual learning," *arXiv preprint arXiv:2510.05070*, 2025.

[5] L. Yang, X. Huang, Z. Wu, A. Kanazawa, P. Abbeel, C. Sferrazza, C. K. Liu, R. Duan, and G. Shi, "Omniretarget: Interaction-preserving data generation for humanoid whole-body loco-manipulation and scene interaction," *arXiv preprint arXiv:2509.26633*, 2025.

[6] Z. Chen, M. Ji, X. Cheng, X. Peng, X. B. Peng, and X. Wang, "Gmt: General motion tracking for humanoid whole-body control," *arXiv preprint arXiv:2506.14770*, 2025.

[7] Y. Ze, Z. Chen, J. P. Araújo, Z.-a. Cao, X. B. Peng, J. Wu, and C. K. Liu, "Twist: Teleoperated whole-body imitation system," *arXiv preprint arXiv:2505.02833*, 2025.

[8] 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," *Conference on Robot Learning (CoRL)*, 2024.

[9] Y. Zhang, Y. Yuan, P. Gurunath, T. He, S. Omidshafiei, A.-a. Aghamohammadi, M. Vazquez-Chanlatte, L. Pedersen, and G. Shi, "Falcon: Learning force-adaptive humanoid loco-manipulation," *arXiv preprint arXiv:2505.06776*, 2025.

[10] H. Weng, Y. Li, N. Sobanbabu, Z. Wang, Z. Luo, T. He, D. Ramanan, and G. Shi, "Hdmi: Learning interactive humanoid whole-body control from human videos," *arXiv preprint arXiv:2509.16757*, 2025.

[11] H. Wang, W. Zhang, R. Yu, T. Huang, J. Ren, F. Jia, Z. Wang, X. Niu, X. Chen, J. Chen, *et al.*, "Physhsi: Towards a real-world generalizable and natural humanoid-scene interaction system," *arXiv preprint arXiv:2510.11072*, 2025.

[12] 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. 16 930–16 936.

[13] G. Ji, J. Mun, H. Kim, and J. Hwangbo, "Concurrent training of a control policy and a state estimator for dynamic and robust legged locomotion," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 4630–4637, 2022.

[14] D. Kim, Y. Zhao, G. Thomas, B. R. Fernandez, and L. Sentis, "Stabilizing series-elastic point-foot bipeds using whole-body operational space control," *IEEE Transactions on Robotics*, vol. 32, no. 6, pp. 1362–1379, 2016.

[15] 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.

[16] D. Baek, A. Purushottam, J. J. Choi, and J. Ramos, "Whole-body bilateral teleoperation with multi-stage object parameter estimation for wheeled humanoid locomanipulation," *arXiv preprint arXiv:2508.09846*, 2025.

[17] H. Li and P. M. Wensing, "Cafe-mpc: A cascaded-fidelity model predictive control framework with tuning-free whole-body control," *IEEE Transactions on Robotics*, 2024.

[18] C. Khazoom, S. Hong, M. Chignoli, E. Stanger-Jones, and S. Kim, "Tailoring solution accuracy for fast whole-body model predictive control of legged robots," *IEEE Robotics and Automation Letters (RA-L)*, 2024.

[19] D. Kim, J. Di Carlo, B. Katz, G. Bledt, and S. Kim, "Highly dynamic quadruped locomotion via whole-body impulse control and model predictive control. arxiv 2019," *arXiv preprint arXiv:1909.06586*.

[20] P. Kormushev, S. Calinon, and D. G. Caldwell, "Reinforcement learning in robotics: Applications and real-world challenges," *Robotics*, vol. 2, no. 3, pp. 122–148, 2013.

[21] J. Tan, T. Zhang, E. Coumans, A. Iscen, Y. Bai, D. Hafner, S. Bohez, and V. Vanhoucke, "Sim-to-real: Learning agile locomotion for quadruped robots," *arXiv preprint arXiv:1804.10332*, 2018.

[22] S. Rho, K. Garg, M. Byrd, and S. Ha, "Unsupervised skill discovery as exploration for learning agile locomotion," in *Conference on Robot Learning*. PMLR, 2025, pp. 2678–2694.

[23] J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter, "Learning quadrupedal locomotion over challenging terrain," *Science robotics*, vol. 5, no. 47, p. eabc5986, 2020.

[24] A. Kumar, Z. Fu, D. Pathak, and J. Malik, "Rma: Rapid motor adaptation for legged robots," *arXiv preprint arXiv:2107.04034*, 2021.

[25] H. Jung, Z. Gu, Y. Zhao, H.-W. Park, and S. Ha, "Ppf: Pre-training and preservative fine-tuning of humanoid locomotion via model-assumption-based regularization," *IEEE Robotics and Automation Letters*, 2025.

[26] Z. Zhang, J. Guo, C. Chen, J. Wang, C. Lin, Y. Lian, H. Xue, Z. Wang, M. Liu, J. Lyu, H. Liu, H. Wang, and L. Yi, "Track any motions under any disturbances," *arXiv:2509.13833*, 2025.

[27] W. Xie, J. Han, J. Zheng, H. Li, X. Liu, J. Shi, W. Zhang, C. Bai, and X. Li, "Kungfubot: Physics-based humanoid whole-body control for learning highly-dynamic skills," *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

[28] N. Mahmood, N. Ghorbani, N. F. Troje, G. Pons-Moll, and M. J. Black, "AMASS: Archive of motion capture as surface shapes," in *International Conference on Computer Vision*, Oct. 2019, pp. 5442–5451.

[29] F. G. Harvey, M. Yurick, D. Nowrouzezahrai, and C. Pal, "Robust motion in-betweening," *ACM Transactions on Graphics (TOG)*, vol. 39, no. 4, pp. 60–1, 2020.

[30] J. Kim, W. Kim, Y. Lu, J. Cheng, F. Zargarbashi, Z. Zeng, Z. Qi, Z. Dou, N. Sontakke, D. Baek, *et al.*, "Switch-justdance: Benchmarking whole body motion tracking policies using a commercial console game," *arXiv preprint arXiv:2511.17925*, 2025.

[31] Y. Li, Y. Lin, J. Cui, T. Liu, W. Liang, Y. Zhu, and S. Huang, "Clone: Closed-loop whole-body humanoid teleoperation for long-horizon tasks," *arXiv preprint arXiv:2506.08931*, 2025.

[32] E. Olson, "AprilTag: A robust and flexible visual fiducial system," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, May 2011, pp. 3400–3407.

[33] B. Wen, W. Yang, J. Kautz, and S. Birchfield, "FoundationPose: Unified 6d pose estimation and tracking of novel objects," in *CVPR*, 2024.

[34] M. Byrd, J. Crandell, M. Das, J. Inman, R. Wright, and S. Ha, "Privileged-dreamer: Explicit imagination of privileged information for rapid adaptation of learned policies," in *2025 IEEE International Conference on Robotics and Automation (ICRA)*, 2025, pp. 3640–3646.

[35] W. Xu and F. Zhang, "Fast-lio: A fast, robust lidar-inertial odometry package by tightly-coupled iterated kalman filter," 2020.

[36] N. Rudin, D. Hoeller, P. Reist, and M. Hutter, "Learning to walk in minutes using massively parallel deep reinforcement learning," in *Proceedings of the 5th Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, A. Faust, D. Hsu, and G. Neumann, Eds., vol. 164. PMLR, 08–11 Nov 2022, pp. 91–100. [Online]. Available: https://proceedings.mlr.press/v164/rudin22a.html

[37] X. B. Peng, M. Andrychowicz, W. Zaremba, and P. Abbeel, "Sim-to-real transfer of robotic control with dynamics randomization," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, May 2018, pp. 1–8.

[38] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," 2017.

[39] W. Yu, G. Turk, and C. K. Liu, "Learning symmetric and low-energy locomotion," *ACM Transactions on Graphics (TOG)*, vol. 37, no. 4, pp. 1–12, 2018.

[40] N. et al., "Isaac lab: A gpu-accelerated simulation framework for multimodal robot learning," 2025.

[41] E. Todorov, T. Erez, and Y. Tassa, "Mujoco: A physics engine for model-based control," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2012, pp. 5026–5033.

[42] "Unitree g1 - unitree robotics." [Online]. Available: https://www.unitree.com/g1

[43] "Realsense d435i." [Online]. Available: https://realsenseai.com/products/depth-camera-d435i/

[44] "Livox mid-360 - livox." [Online]. Available: https://www.livoxtech.com/mid-360

[45] DuckieTown, "Python bindings for the apriltags library," https://github.com/duckietown/lib-dt-apriltags, 2020.

[46] X. B. Peng, P. Abbeel, S. Levine, and M. Van de Panne, "Deepmimic: Example-guided deep reinforcement learning of physics-based character skills," *ACM Transactions On Graphics (TOG)*, vol. 37, no. 4, pp. 1–14, 2018.