Expert-Guided Imitation for Learning Humanoid Loco-Manipulation from Motion Capture

Rohan P. Singh[1†], Pierre-Alexandre Leziart[1†], Masaki Murooka[1], Mitsuharu Morisawa[1], Eiichi Yoshida[2], Fumio Kanehiro[1]

***Abstract*— Despite significant advances in bipedal locomotion, enabling humanoid robots to perform general whole-body tasks through meaningful interaction with their environments remains a challenging open problem. While deep reinforcement learning (RL) has recently demonstrated impressive results in dynamic walking — even on complex and unpredictable terrain — real-world utility demands that humanoids go beyond locomotion to execute task-oriented behaviors.**

In this work, we propose a framework for teaching humanoid robots to imitate humans doing useful tasks by training policies for tracking human motion references. Our approach leverages high-quality in-house motion capture (MoCap) data, from which we perform kinematic retargeting to project human trajectories onto a humanoid platform. Crucially, we adopt a hybrid learning paradigm: the policy is trained to track upper-body and root motions from the MoCap data, and receives additional supervision from a pre-trained omnidirectional walking expert. This expert guidance, implemented via a Behavior Cloning (BC) objective, ensures that leg motion respects dynamics and kinematic constraints of the humanoid. We train policies entirely in simulation and successfully transfer them to a real humanoid robot. We validate our method on a box loco-manipulation task, demonstrating effective sim-to-real transfer and marking a step toward more capable, task-driven humanoid behavior.

Project website: `https://isri-aist.github.io/GuidedHumanoidLocomanipulation/`

I. Introduction

Deep reinforcement learning (RL) algorithms have made significant progress in recent years for controlling legged robots. Quadruped robots can now efficiently accomplish a wide variety of complex locomotion tasks using RL approaches, such as parkour [1], search and rescue [2] or acrobatics [3].

Compared to quadrupeds, learning-based humanoid whole-body locomotion is still less developed, though interest has accelerated recently as real-world environments are fundamentally build around the human form factor [4]. Progress has increased notably in the recent few years [5], [6]. But humanoids remain harder to train due to their greater complexity and inherent instability, which demand additional techniques for achieving reliable balance [7]. These factors also make reward design, training strategies, and sim-to-real transfer more involved and dependent on expert tuning.

In this paper, we seek to mitigate this burden by proposing a framework for humanoid loco-manipulation that leverages human demonstrations to guide RL exploration.

† Equal contribution. `Email: rohan-singh@aist.go.jp`

[1] CNRS-AIST JRL (Joint Robotics Laboratory) IRL, National Institute of Advanced Industrial Science and Technology (AIST), Japan.

[2] Tokyo University of Science, Tokyo, Japan.

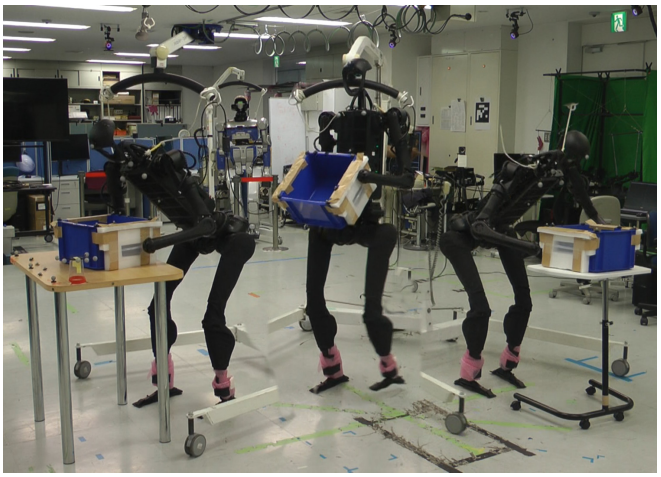


Fig. 1: Loco-manipulation scenario with a H1 humanoid robot: picking a box up and dropping it off on another table.

In the RL scheme, properly shaping the reward function is key to guide the learning process and accomplish the desired task. Some terms directly stem from common sense, such as a penalties when getting close from hardware limits, and others from intuition and the nature of the task itself. However, as task complexity grows, so does the pressure put on the reward function to achieve natural, efficient and safe motions [8]. Stacking dense reward terms to control all aspects of the task increases the burden of hyperparameter tuning and can adversely constrain exploration if badly chosen [9]. Sparse reward terms enable more leeway but the lack of clear guidance can hinder the learning process as the agent receives feedback infrequently [10]. In cases where one already knows how to solve the considered tasks, imitating a fixed reference comes as a way to alleviate this struggle by directly providing the robot with demonstrations on how to behave to solve them [11].

In this paper, we present a framework for humanoid loco-manipulation focused on a practical box-handling task involving approach, transport, and placement. We demonstrate that combining reinforcement learning with imitation of human demonstrations is a viable strategy for producing successful, deployable policies. The core idea is to rely on human demonstrations for the components of the task that involve fine-grained robot–environment interaction, such as grasping, lifting, and placing, while using reference-free RL to learn robust omnidirectional walking through model-free exploration in simulation. This decoupling allows the system to exploit the strengths of model-free RL (using Proximal

Policy Optimization (PPO) [12]) while also alleviate the problem of reward desiging. It also makes it possible to leverage recent advances in sim-to-real transfer to obtain high-performance locomotion on hardware. We validate the full framework on the Unitree H1 humanoid robot, showing that the learned policy executes the complete box loco-manipulation sequence in the real world. Furthermore, we show simulation studies indicating that anchoring the manipulation portion of the policy to reference motion does not constrain its ability to generalize, and the learned controller can successfully adapt to novel scene layouts and object placements.

In summary, our contributions are the following:

1) We design a two-stages learning pipeline for humanoid box loco-manipulation that relies on a mix of reinforcement learning for frame-by-frame motion tracking and behavior cloning.
2) We study the robustness and generalization of the trained policy under variability between the data capture scene and the policy deployment scene.
3) We highlight how this generalization can be exploited for long distance transport by manipulating the pick and drop locations.
4) We validate our framework by deploying it to a real-world H1 humanoid robot and demonstrating the reproducibility of the whole pick-and-drop cycle.

II. Related Work

The control of biped robots has been studied for decades. Back in the 1970s, the WAP-1 could already playback movements using artificial rubber muscles [13]. The WL-10 RD biped that came later in the 1980s achieved a quasi-dynamic gait transferring support from one foot to another [14]. As the fundamentals of bipedal locomotion started to be mastered, the question of exploiting the manipulation capabilities of a upper body rose as well. These early developments culminated in 1999 with the WABIAN humanoid robot capable of moving while transporting loads with its arms [15]. As both hardware and control frameworks progressed, more complex loco-manipulation scenarios started to be explored, like pushing at table [16] or achieving cooperation between two robots to carry a stretcher [17]. Task-space force controllers can now achieve reliable manipulation of rigid objects and articulated mechanisms, such as doors, drawers, boxes and tables [18], [19]. Provided that the environment is known, humanoid robots can even actively use their surroundings as part of their control scheme by using their arms to support themselves or grabbing a railing to climb stairs [20]. Recent works include exteroception in their control pipeline for environment reconstruction to achieve autonomous multi-contact planning [21], [22]. Model-based control framework are now mature enough to consider industrial loco-manipulation applications [23], [24] although they require extensive software engineering efforts to run in real time and are per nature sensitive to status and model uncertainties, especially for contacts.

Model-free learning approaches appear as a compelling solution to address these issues by circumventing the need to model contacts as part of the control pipeline, as well as having a low computational footprint once trained. Following their rise in the recent years, a wide range of algorithms have achieved impressive results in solving complex tasks [6], [25]. Humanoid locomotion has seen a resurgence of recent developments that benefit from techniques and tools already applied on quadruped robots. Although training policies in simulation before transferring them to the real world has been a popular approach for a while [26], [27], it has now been boosted by several GPU-based simulators capable of simulating thousands of robots in parallel [28], [29], which has streamlined this process [30]. However, even with this step up in term of sample generation, exploration remains an issue for complex tasks, especially when the robots bring their inherent complexity, like humanoid robots.

Imitation learning, also referred to as learning by demonstration, offers the means to quickly transfer skills from a demonstrator to a learning agent [31]. It enables humanoid robots to learn to replicate natural whole-body locomotion patterns and execute seamless gait transitions by mimicking human motions from a motion capture dataset [32], [33]. This also applies to humanoid loco-manipulation tasks using teleoperation examples to imitate [34], [35] or motion tracked demonstrations [36]. [37] provides an alternative with a hierarchical scheme that combines a high-level waypoint planner with several specialized low-level policies that handle motion primitives (pick up, put down, walk). The closest developments from our own appears to be [38] with the sim-to-real transfer of a controller on a Digit robot to move boxes from one table to another using 5 separate RL policies for the various stages of the motion. By comparison, we seamlessly transition between walking, standing and picking-up (or dropping-off) the box with a single policy by mimicking the motion reference.

III. Approach

Our objective is to develop a methodology that enables humanoid robots to learn loco-manipulation behaviors directly from human motion demonstrations. To this end, we propose a framework that combines motion capture, inverse kinematics (IK), and reinforcement learning (RL) with an auxiliary Behavior Cloning (BC) loss term to achieve motion imitation while maintaining dynamic feasibility on humanoid platforms. The pipeline begins with the collection of high-quality human motion data using a motion capture (MoCap) system, specifically targeting the task of box loco-manipulation which involves both locomotion and object interaction.

The captured MoCap trajectories are retargeted to a humanoid robot by training a policy using a decoupled approach. We incorporate a pre-trained omnidirectional walking policy to serve as a foundation for achieving stable bipedal locomotion. On top of this, we design an RL environment for tracking of upper body and root reference motion trajectories and imitating contact states, thereby enabling the

policy to perform both locomotion and manipulation in a sim-to-real plausible manner. Both mechanisms are applied simultaneously during the training stage.

An overview of our system architecture is provided in Fig. 2. We train three separate policies each trained to imitate one reference clip:

- Starting from origin, approach the "box" placed on "Table A" and pick it up.
- Carry the "box" from "Table A" to "Table B" and place it down.
- Return from "Table B" to the starting location.

In the following sections, we describe each component of our proposed framework in detail.

A. Data Capture

We collect our reference motion data in-house by having a human subject perform the loco-manipulation task while wearing a full-body motion capture suit. The capture setup utilizes OptiTrack cameras in conjunction with Motive software [1] to record both the 3D trajectories of body markers and the corresponding skeletal motion. In addition to motion data, we also record contact forces using off-the-shelf loadsol[2] sensors, which are placed on each of the subject's hands as well as on the bottom surface of the *box*. These sensors provide measurements of the normal plantar force, enabling accurate identification of contact events during interaction. This contact data plays a crucial role in policy training, as we incorporate a specific reward term that encourages the reinforcement learning (RL) policy to replicate the observed contact patterns. The human subject performs two motion clips: in the first, they walk forward from a starting position toward a *box* placed on a *table* and grasp it; in the second, they carry the *box* laterally to a second *table* positioned approximately 90 degrees from the initial direction and place the *box* down. The spatial configuration of the scene in both clips is illustrated in Fig. 3.

Timestamps of the measurements from the loadsol sensors and the MoCap are aligned manually by registering an in place jump at the start of the collected sequence.

B. Inverse Kinematics

The problem of kinematic retargeting of the collected MoCap trajectory from a human skeleton to a humanoid robot is formulated as an optimization-based inverse kinematics problem. We formulate IK as a constrained quadratic programming (QP) problem to compute full joint configurations of the humanoid robot that best match a reference human motion. The objective function minimizes the weighted sum of squared errors between the 3D positions and orientations of key end-effectors (hands and feet) as well as the pose of the torso, and the head. This formulation allows for smooth tracking of human motion while preserving the structural semantics of the original trajectory.

To ensure physical plausibility and compatibility with the robot's model, we incorporate hard constraints into the QP that enforce joint limits and prevent self-collisions. Although the QP-based IK solver performs well in producing feasible retargeted motions, it can occasionally introduce unnatural joint configurations or discontinuities, particularly in highly dynamic or constrained scenarios. These artifacts are manually post-processed using a manual cleanup pass to correct for visually implausible poses or abrupt transitions.

The *box* motion, and the positions of the two *tables* (in Fig. 3 do not require any retargetting.

C. Motion Tracking with RL

Thus far, we obtain a time series reference dataset consisting of the whole-body joint positions, joint velocities, global pose of the root body (pelvis), relative poses of the end-effector bodies in the root frame, relative poses of the all scene elements (*box*, *Table A*, *Table B*) in the root frame, and a contact graph. The contact graph represents a binary contact indicator for the following pairs in the scene: hands to *box* (assuming both hands as one node), *box* to *Table A*, *box* to *Table B*.

High-quality and reliable reference data allows us to use model-free deep reinforcement learning to train policies in a simulation environment for imitating the human motion. Synchronization between the robot and the reference is achieved through an observed phase variable. Each policy is trained for tracking only a single clip.

Observations and Actions. Observations include the root roll and pitch angles, root angular velocity in local frame, joint positions, joint velocities, applied joint torques at the previous timestep. Policy also observes the relative pose of the target scene element: the *box* pose while approaching it for pickup, the pose of put down location when *box* is in the hands, and pose of the origin in the robot's root frame. Further, we introduce the observation of two clock signals: (1) phase variable for synchronizing reference state and robot state, (2) periodic bipedal gait. We maintain a clock signal for bipedal gait as it is included in the observation space of the expert walking policy (based on [39]) used in our work. Further, in our experience, we found directly observing a periodic clock signal to be helpful with sim-to-real transfers especially for networks without an observation history.

The action space comprises of joint position targets for all 19 joints of the Unitree H1 humanoid robot.

Intialization and Terminations. We incorporate reference state initialization (RSI) [40] and early termination based on distance from the reference clip. Specifically, we terminate a rollout if the root tracking error or the body tracking error is larger than a certain threshold (0.4 and 0.6 respectively). Another important termination condition includes termination upon "bad contact", which is defined as a collision between the robot and either *tables* or self collisions. Other termination conditions are based on root height (terminate if root height $< 0.6m$) and joint position (terminate if any joint is within 3 degrees of the joint limit).

During RSI, we note that it is important to initialize the *box* pose too, ensuring that both hands are pushing inwards

[1] https://optitrack.com/software/motive/
[2] https://www.novelusa.com/loadsol

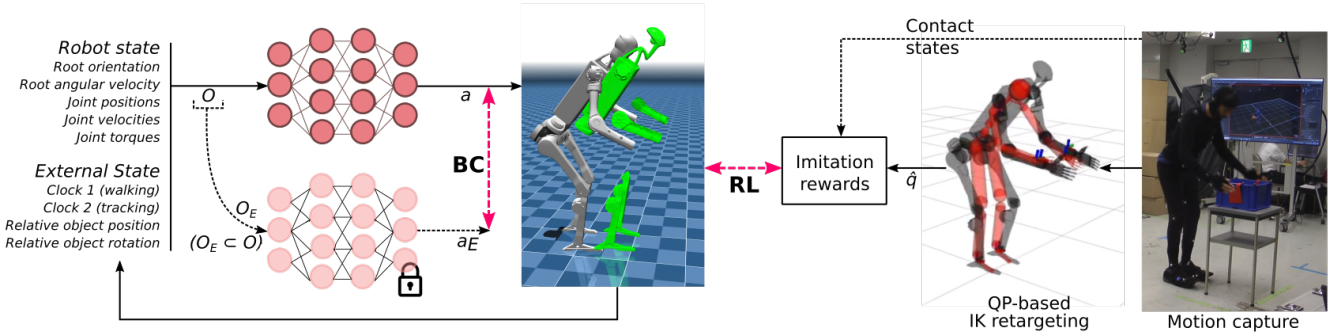


Fig. 2: **Overview of the proposed hybrid training approach.** The whole-body imitation policy is rewarded to track only the upper body and root motion of the human demonstrations while an auxiliary behavior cloning loss term provides supervision from an expert walking policy. The student policy learns to control all joints of the robot and successfully achieves the loco-manipulation task.

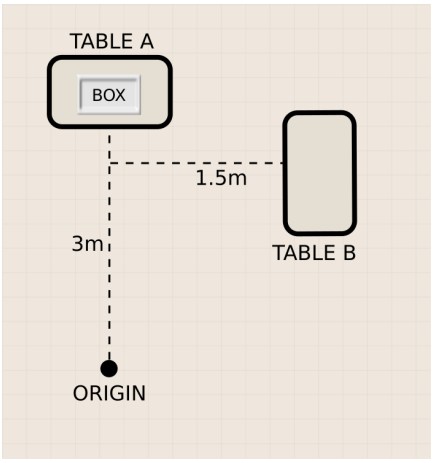


Fig. 3: **Scene layout.** We used the above arrangement of the scene during MoCap data collection and during real robot experiments. For real robot experiments, we split the task into 3 legs and train one policy for each: origin to *box* pickup from *Table A*, *box* carry from *Table A* to *Table B* and put down, walk backwards to the origin. The distances shown here are approximate. The scene objects do not need to be placed precisely at fixed locataions.

on the sides. We found that this requires manually editing the size of the *box* to make it slightly larger.

Rewards. The reward function includes terms for tracking the (1) root height and orientation, (2) relative poses of torso, left arm, and right arm in pelvis frame, (3) joint position and velocity (only upper body joints when robot is moving, and all joints when standing) (4) relative pose of target object (*box* or *Table B*) in pelvis frame, (5) contact graph, and (6) relative positions of the hands in the frame of the *box*. We also includes regularization terms for minimizing joint torques, and penalties for joints near the range limits. All terms are implemented in the form of a Gaussian kernel $f(x, \hat{x}) = \exp\left(-\frac{1}{2\sigma^2}\left\|x - \hat{x}\right\|^2\right)$ where $\hat{x}$ represents the reference value for the variable x, and σ is a parameter that controls the kernel's spread. We note several crucial points while developing the reward function. First, we note that the tracking rewards on joints are applied only to the upper body joints (arms, waist) when the reference robot is moving i.e. when the reference root linear velocity in xy-plane is greater than a certain threshold $(= 0.15m/s)$. Second, it is important to ensure that the relative positions of hands in *box* frame is such that the hands are slightly penetrating the sides of the *box*, in order to encourage firm grasping.

The term for tracking contact states is implemented with a very small σ value, making it behave as a sparse reward: $+1$ if the contact graph is matched exactly, and 0 otherwise.

D. Decoupled Supervision for Loco-Manipulation

While our motion tracking reinforcement learning framework enables the robot to learn to imitate human motion, directly mimicking full-body trajectories including walking movements is infeasible due to the substantial dynamic and morphological differences between humans and humanoid robots.

In particular, the discrepancy in limb proportions and joint constraints often leads to violations of dynamic stability when attempting to directly replicate human walking patterns. Our early sim-to-real experiments showed that the robot struggles to make stable foot contacts with the floor while making unrealistically long strides. Furthermore, tuning IK parameters to enforce both kinematic accuracy and plausible contact dynamics such as maintaining foot orientation is labor-intensive and not scalable.

To address this, we decompose the imitation task: the imitation policy being trained (π_θ) acts as a *student* and is rewarded to track only the upper body and root motion of the human demonstrator, governing the manipulation aspect of the task. For the motion of the legs for achieving bipedal locomotion, we instead provide direct supervision from a pre-trained omnidirectional walking policy that serves as an expert *teacher* (π_E). We implement this supervision through a Behavior Cloning (BC) objective, guiding the student policy to match the expert's leg actions during training. This hybrid approach allows the student policy to benefit from high-level human demonstrations while leveraging the robustness and stability of the expert locomotion policy, resulting in coherent whole-body behavior that successfully integrates walking and manipulation.

Training an expert bipedal walking policy. We adapt the approach presented in [39], [41], for the Unitree H1, to train a highly performant expert policy for achieving bipedal locomotion. The trained policy generates actions only for the leg joints given the proprioceptive state of the robot, a

periodic clock signal, and a reference root velocity command. We denote this policy as $\pi_E(a \mid o_E)$, giving the action vector $a \in \mathbb{R}^{10}$ for the observation $o_E \in \mathbb{R}^{36}$. As described in [39], o_E consists of the 2D periodic clock signal, command mode vector, the robot state including base roll and pitch angles, angular velocity of base, and the positions, velocities, and applied torques of only the leg joints.

BC Auxiliary Loss Term. We modify the original PPO loss term by introducing an auxiliary loss term as follows.

Let $\pi_\theta(a \mid o)$ be the student policy parameterized by θ. The training loss L_{total} combines the PPO loss L_{PPO} with a behavior cloning (BC) loss L_{BC}.

$$L_{\text{total}} = L_{\text{PPO}} + \lambda_{\text{BC}}\, L_{\text{BC}}$$

where the BC loss is defined as:

$$L_{\text{BC}} = \left\| \pi_\theta^{\text{legs}}(o) - \pi_E(o_E) \right\|^2$$

Here, $\pi_\theta^{\text{legs}}(s)$ denotes the actions corresponding only to the leg joints produced by the student policy, and λ_{BC} is a weighting coefficient controlling the strength of expert supervision. L_{PPO} is the original clipped surrogate objective combining the policy loss, value loss, and entropy loss terms [12]. We set $\lambda_{\text{BC}} = 0.3$ through all our experiments.

IV. Simulation study

A. Training process

To train our policies, we implement the proposed approach using [39] as a basis. This pipeline relies on the CPU-based simulation engine Mujoco [28] combined with Ray [42] as a parallelization framework to scale training on several cores. First, we leverage the PPO algorithm [12] to train an omnidirectional walking policy on flat ground for a velocity tracking task. We apply random pushes to the robot, dynamics randomization and sensor noise, and we add random bumps on the ground as additional disturbances. The expert walker described in subsection III-D is trained to convergence in around 30000 epochs. The imitation policy is then trained for 20000 epochs with all domain randomization disabled. It is further finetuned after convergence for a few thousands epochs by re-enabling all randomization for better sim-to-real transfer. The whole process amounts to around 36 hours of training using a 32-cores AMD Threadripper PRO 5975WX to gather samples from 32 environments simultaneously

B. Robustness to target offsets

Domain randomization enables some amount of robustness to noise and external disturbances, yet the initial standing position of the robot and those of the box and drop-off table remain the same over the training. Those positions corresponds to the ones that were used for the motion capture recording. Although we are still far from a generalized loco-manipulation controller, for potential future applications it would already be relevant to know how well the proposed approach can handle offsets to the pick and drop locations. This is a way to assess how the policy behaves in slightly out-of-distribution scenarios.

Thus, we study the success rate of the policy for various pick-and-drop positions and orientations around the baseline training scenario. All control parameters remain the same as for the nominal case except the starting phase clock signal. During training the robot always starts 3 meters from the box and moves towards it as the phase goes from 0 to 1. This phase is a strong signal for the robot to understand where it should be and how it should act. For instance, after being pushed backwards, the robot realizes it is further from the box than expected for the current phase value and will thus speed up to arrive in time in front of the box. To take this effect into account we roughly scale the starting phase with the initial distance from the box, as reported in Table I. If we were to start the episode with the box 1 meter from the robot using the default phase value, it would suddenly try to move backwards and fall.

TABLE I: The initial box distance scales the value of the starting phase.

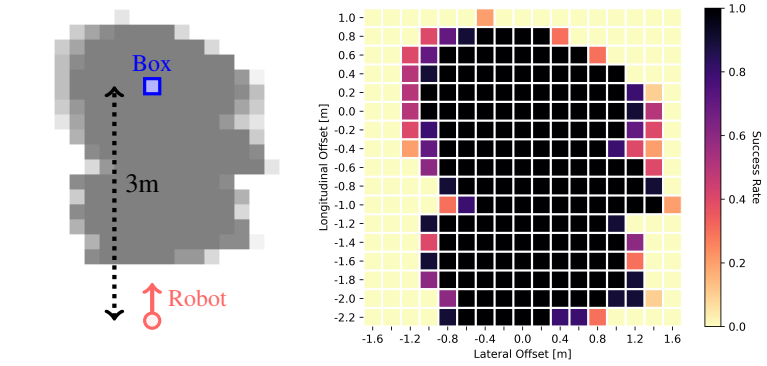
Box distance [m]	< 1	$[1, 1.5]$	$[1.5, 2]$	$[2, 2.5]$	> 2.5
Starting phase	0.5	0.4	0.35	0.2	0.1

We report success rates for pick-up and drop-off in Fig. 4. The robot achieves a consistent pick-up of the box over a wide range of offsets going roughly from $[-2.0, 0.4]m$ and $[-0.8, 0.8]m$ for the longitudinal and lateral axes, $[-5, 15]cm$ in height and $[-0.4, 0.4]rad$ in orientation. This highlights the efficiency of domain randomization, notably of random pushes, for handling setups that deviate from the motion capture recording, thus extending the robot workspace.

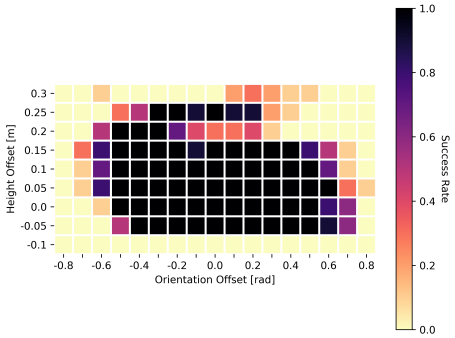
C. Extension to long distance loco-manipulation

This capacity to handle a wide range of deviations from the reference demonstration pushed us into exploring loco-manipulation over long distances. To do so, we leverage a 2D planning algorithm to generate paths along which fake target positions will be set to sequentially lead the robot to the real pick-up and drop-off locations. The training focuses on walking forwards, with small turns or sideways steps when heading to the drop-off table or after being pushed. For this reason we integrate in our pipeline the open-source RRT-Dubins path planner that generates Dubins paths [43] using Rapidly-exploring Random Trees. These paths connect two points with forward travel only and a constraint on the path curvature, which fits the limitations of the policy.

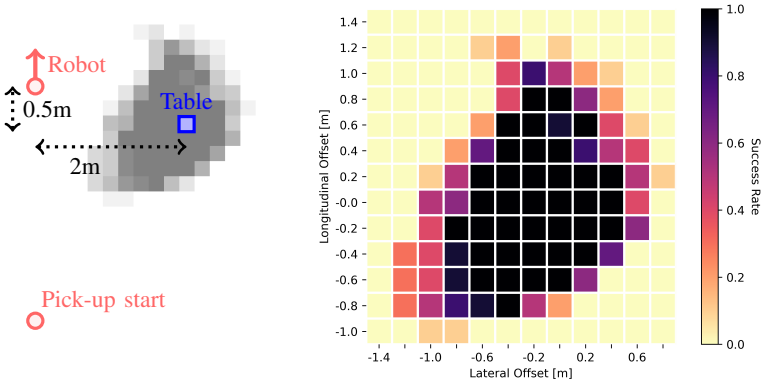
At startup, we compute a path to link the current 2D position and orientation of the robot with the ones of the box or the drop-off table. Then, we generate a fake position and orientation along the path away from the robot. This position is provided to the policy in the observations as if it were the real one, as described in subsection III-C. Once the phase clock signal reaches a threshold we refresh it, otherwise the robot would start to bow down to perform the pick-up or drop-off motion. We generate a new target along the path and set the phase back to a lower value to keep walking. This process is repeated until the true target is reached. Fig. 5 highlights a pick-and-drop sequence alongside the



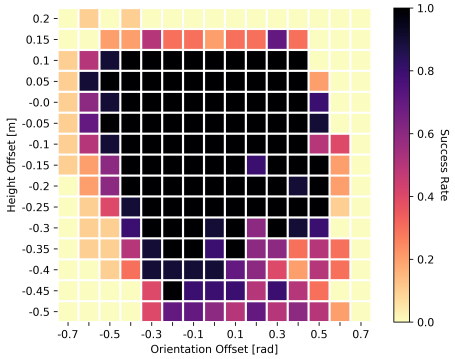
(a) Horizontal position offset on the box with respect to the default pick-up configuration.



(b) Vertical position and orientation offset on box for pick-up.

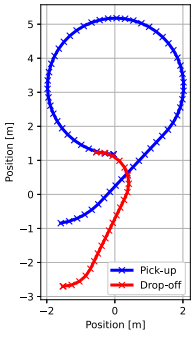


(c) Horizontal position offset on the table with respect to the default drop-off configuration.

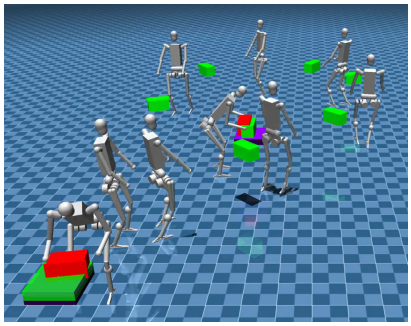


(d) Vertical position and orientation offset on table for drop-off.

Fig. 4: Success rates of the box pick-up (resp. drop-off) motion in simulation depending on position and orientation offsets added to the box (resp. table) with respect to the nominal configuration found in the imitated sequence. Orientation offset are applied along the vertical axis. The nominal configuration corresponds to $(0, 0)$ coordinates on the graphs. For instance, (b) highlights that the robot fails to pick-up the box once it is 10 cm lower than nominal, and that for the nominal height it can consistently pick it up if the box orientation is between $(-0.5, 0.5)$ rad. Success rates are gathered over 10 trials for each combination of offsets. The trials are done on flat ground without random pushes applied to the torso nor dynamics randomization. At the end of the motion, a successful pickup is detected if box height is above 90 cm with both hand in contact with the box. For drop-off, a success is detected if the box lies on the table at the end of the sequence, without any hand touching the box.



(a) Dubins paths.



(b) Snapshots of pick-up motion and final drop-off.

Fig. 5: Dubins path for the pick and drop motion (left) and snapshots along the trajectory (right). Intermediate frames of the drop-off part are omitted for clarity. The fake box and table positions are displayed in green. Due to the box position and the path curvature constraint, the robot does not directly move to the box but instead moves past it before looping back.

paths generated by the RRT-Dubins planner. For this scenario the box is placed 1.5m forwards and 2m to the left instead of just being placed 3 meters forwards as in the motion reference. Without planner this motion would fail since it is out of the usable workspace displayed in Fig. 4(a). This demonstrates a generalized use-case of the policy that goes beyond the single motion capture recording we performed.

V. Experimental Validation

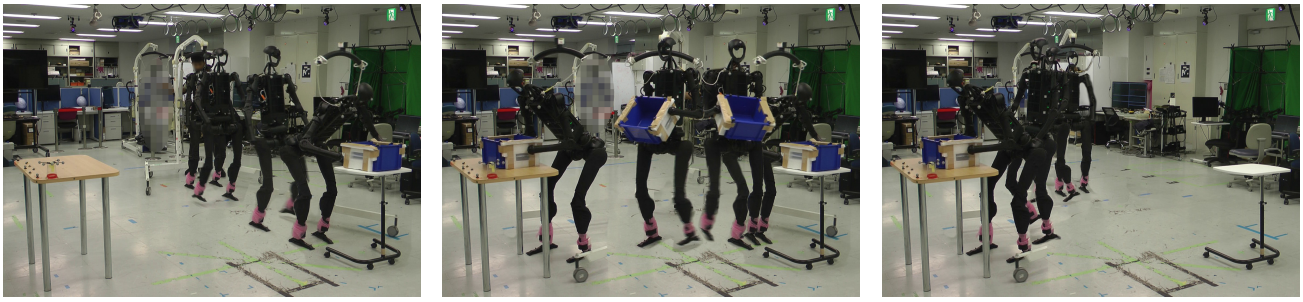
A. Experimental setup

After training in simulation, the controller is directly deployed on a real Unitree H1 robot. The policy runs at 40 Hz on a standard laptop computer by using the Open Neural Network Exchange (ONNX) framework through the ONNX Runtime inference engine [44]. Communications with the robot are ensured by the Unitree SDK2 through an Ethernet connection. As the focus of this work is not to perform fully autonomous demonstrations, we simplify the experimental setup by using motion capture instead of onboard sensors to track the position of the robot, the box and the drop-off table. We arrange the scene manually to roughly match the layout used during data collection; while the resulting deviations in object placement and orientation are non-negligible, they remain within a reasonable range. That is, for real robot eevaluations, the scene layout is broadly consistent with the data-collection setup.

Motion capture is used during deployment too, for obtaining object poses in the robot frame for policy observations.

B. Real-world deployment

Real-world motion control results are shown in Fig. 6. The robot achieves a full loco-manipulation cycle by walking toward the box, picking it up to drop it off on another table,



(a) Picking up the box. (b) Dropping off the box. (c) Going back to starting position.

Fig. 6: Full loco-manipulation cycle by automatically switching policies online. The support crane has been partially edited out for visibility purpose.

then going back to its starting position. This is done in one go by automatically switching between policies when the phase clock signal reaches its final value. The cycle can be repeated on-the-fly by placing the box back on the first table while the robot is returning to the starting position. This successful deployment indicates that the domain randomization we used was effective for crossing the sim-to-real gap.

Limitations. While the locomotion part of the motion is robust and repeatable, most failures occur when picking-up the box. Failures during the drop-off sequences mostly occurred when the box was not properly picked-up, with the box falling from the robot's hands before reaching the table. The differences in dynamics for the contact interactions between our training setup and reality partially explain these failures. Moreover, since H1 only has relative encoders in its arms, slight calibration errors lead to joint position offsets, which can worsen the overall sim-to-real gap. Rewarding the policy to apply more forces on the box during training could be a way to address this issue, to encourage the robot to firmly grasp the box. However, this is not an ideal solution for general loco-manipulation as uncontrolled internal forces may damage the robot and/or the box. Improvements could come from a better contact perception so that the robot can react online and correct improper pick-ups. Finetuning the policy with an extensive domain randomization on box mass, size, or friction could also help reduce this sim-to-real gap.

VI. Conclusion

In this work, we investigated the problem of retargeting human motion to humanoid robots for loco-manipulation tasks using deep reinforcement learning. Specifically, we captured high-quality human motion data via a motion capture system for a box loco-manipulation scenario, and subsequently employed inverse kinematics to map these trajectories to a humanoid robot model. To ensure the dynamic feasibility of the generated motions, we trained an RL policy to imitate the retargeted trajectories rather than directly replicating the raw human motion. A key contribution of our approach lies in leveraging pre-trained omnidirectional walking policies, which significantly enhance the dynamic stability of the humanoid during locomotion. This is particularly advantageous given the morphological disparities between human legs and humanoid actuators, which make direct leg motion tracking impractical. Additionally, we highlight the critical role of high-fidelity MoCap data in enabling realistic and physically plausible behavior. We further demonstrate that training the bipedal walking policy on randomized terrains introduces beneficial exploration dynamics, resulting in robust locomotion strategies that generalize to a wider range of environmental conditions.

A notable limitation of our current method is its reduced effectiveness in tasks that demand precise leg motion tracking, such as stepping onto elevated platforms to access a box. As part of future work, we aim to expand the diversity of loco-manipulation tasks to include more complex and varied interactions, explore more sophisticated IK techniques, and improve generalization to arbitrary box positions and environmental layouts. Including on-board perception in our pipeline would be a step toward greater autonomy.

Acknowledgements

The authors thank all members of JRL for providing their support in conducting robot experiments that were done during the production of this work. This work was partially supported by JSPS KAKENHI Scientific Research (S) Grants Number JP22H05002 and JP24KF0125, and by the Japan Society for the Promotion of Science (JSPS) Postdoctoral Fellowships for Research in Japan.

References

[1] X. Cheng, K. Shi, A. Agarwal, and D. Pathak, "Extreme parkour with legged robots," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 11 443–11 450.

[2] M. Tranzatto, T. Miki, M. Dharmadhikari, L. Bernreiter, M. Kulkarni, F. Mascarich, O. Andersson, S. Khattak, M. Hutter, R. Siegwart, *et al.*, "Cerberus in the darpa subterranean challenge," *Science Robotics*, vol. 7, no. 66, p. eabp9742, 2022.

[3] N. Rudin, H. Kolvenbach, V. Tsounis, and M. Hutter, "Cat-like jumping and landing of legged robots in low gravity using deep reinforcement learning," *IEEE Transactions on Robotics*, vol. 38, no. 1, pp. 317–328, 2021.

[4] S. Ha, J. Lee, M. van de Panne, Z. Xie, W. Yu, and M. Khadiv, "Learning-based legged locomotion: State of the art and future perspectives," *The International Journal of Robotics Research*, vol. 44, no. 8, pp. 1396–1427, 2025.

[5] Z. Zhuang, S. Yao, and H. Zhao, "Humanoid parkour learning," *arXiv preprint arXiv:2406.10759*, 2024.

[6] L. Bao, J. Humphreys, T. Peng, and C. Zhou, "Deep reinforcement learning for bipedal locomotion: A brief survey," *arXiv preprint arXiv:2404.17070*, 2024.

[7] Y. Xie, B. Lou, A. Xie, and D. Zhang, "A review: Robust locomotion for biped humanoid robots," in *Journal of Physics: Conference Series*, vol. 1487, no. 1. IOP Publishing, 2020, p. 012048.

[8] S. Ibrahim, M. Mostafa, A. Jnadi, H. Salloum, and P. Osinenko, "Comprehensive overview of reward engineering and shaping in advancing reinforcement learning applications," *IEEE Access*, 2024.

[9] M. Riedmiller, R. Hafner, T. Lampe, M. Neunert, J. Degrave, T. Wiele, V. Mnih, N. Heess, and J. T. Springenberg, "Learning by playing solving sparse reward tasks from scratch," in *International conference on machine learning*. PMLR, 2018, pp. 4344–4353.

[10] J. Kober, J. A. Bagnell, and J. Peters, "Reinforcement learning in robotics: A survey," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1238–1274, 2013.

[11] A. Hussein, M. M. Gaber, E. Elyan, and C. Jayne, "Imitation learning: A survey of learning methods," *ACM Computing Surveys (CSUR)*, vol. 50, no. 2, pp. 1–35, 2017.

[12] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[13] H.-o. Lim and A. Takanishi, "Biped walking robots created at waseda university: Wl and wabian family," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 365, no. 1850, pp. 49–64, 2007.

[14] A. Takanishi, M. Ishida, Y. Yamazaki, and I. Kato, "The realization of dynamic walking by the biped walking robot wl-10 rd," *Journal of the Robotics Society of Japan*, vol. 3, no. 4, pp. 325–336, 1985.

[15] J. Yamaguchi, E. Soga, S. Inoue, and A. Takanishi, "Development of a bipedal humanoid robot-control method of whole body cooperative dynamic biped walking," in *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No. 99CH36288C)*, vol. 1. IEEE, 1999, pp. 368–374.

[16] K. Harada, S. Kajita, F. Kanehiro, K. Fujiwara, K. Kaneko, K. Yokoi, and H. Hirukawa, "Real-time planning of humanoid robot's gait for force-controlled manipulation," *IEEE/ASME Transactions on Mechatronics*, vol. 12, no. 1, pp. 53–62, 2007.

[17] S. G. McGill and D. D. Lee, "Cooperative humanoid stretcher manipulation and locomotion," in *2011 11th IEEE-RAS international conference on humanoid robots*. IEEE, 2011, pp. 429–433.

[18] K. Bouyarmane, K. Chappellet, J. Vaillant, and A. Kheddar, "Quadratic programming for multirobot and task-space force control," *IEEE Transactions on Robotics*, vol. 35, no. 1, pp. 64–77, 2018.

[19] I. Maroger, O. Stasse, and B. Watier, "From the study of table trajectories during collaborative carriages toward pro-active human-robot table handling tasks," in *2022 IEEE-RAS 21st International Conference on Humanoid Robots (Humanoids)*. IEEE, 2022, pp. 911–918.

[20] J. Carpentier, S. Tonneau, M. Naveau, O. Stasse, and N. Mansard, "A versatile and efficient pattern generator for generalized legged locomotion," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 3555–3561.

[21] P. Seiwald, S.-C. Wu, F. Sygulla, T. F. Berninger, N.-S. Staufenberg, M. F. Sattler, N. Neuburger, D. Rixen, and F. Tombari, "Lola v1. 1–an upgrade in hardware and software design for dynamic multi-contact locomotion," in *2020 IEEE-RAS 20th International Conference on Humanoid Robots (Humanoids)*. IEEE, 2021, pp. 9–16.

[22] P. Seiwald and S.-C. Wu. Humanoid robot lola - vision guided autonomous multi-contact locomotion. TUM Chair of Applied Mechanics. [Online]. Available: https://www.youtube.com/watch?v=ovG2Rz9-1p8

[23] J. Mirabel, F. Lamiraux, T. L. Ha, A. Nicolin, O. Stasse, and S. Boria, "Performing manufacturing tasks with a mobile manipulator: from motion planning to sensor based motion control," in *2021 IEEE 17th International Conference on Automation Science and Engineering (CASE)*. IEEE, 2021, pp. 159–164.

[24] K. Chappellet, M. Murooka, G. Caron, F. Kanehiro, and A. Kheddar, "Humanoid loco-manipulations using combined fast dense 3d tracking and slam with wide-angle depth-images," *IEEE Transactions on Automation Science and Engineering*, 2023.

[25] B. Singh, R. Kumar, and V. P. Singh, "Reinforcement learning in robotic applications: a comprehensive survey," *Artificial Intelligence Review*, vol. 55, no. 2, pp. 945–990, 2022.

[26] 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)*. IEEE, 2018, pp. 3803–3810.

[27] M. Aractingi, P.-A. Léziart, T. Flayols, J. Perez, T. Silander, and P. Souères, "Controlling the solo12 quadruped robot with deep reinforcement learning," *scientific Reports*, vol. 13, no. 1, p. 11945, 2023.

[28] 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.

[29] V. Makoviychuk, L. Wawrzyniak, Y. Guo, M. Lu, K. Storey, M. Macklin, D. Hoeller, N. Rudin, A. Allshire, A. Handa, and G. State, "Isaac gym: High performance gpu-based physics simulation for robot learning," 2021.

[30] N. Rudin, D. Hoeller, P. Reist, and M. Hutter, "Learning to walk in minutes using massively parallel deep reinforcement learning," in *Conference on Robot Learning*, 2022.

[31] A. Gams, T. Petrič, B. Nemec, and A. Ude, "Manipulation learning on humanoid robots," *Current Robotics Reports*, vol. 3, no. 3, pp. 97–109, 2022.

[32] A. Tang, T. Hiraoka, N. Hiraoka, F. Shi, K. Kawaharazuka, K. Kojima, K. Okada, and M. Inaba, "Humanmimic: Learning natural locomotion and transitions for humanoid robot via wasserstein adversarial imitation," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 13 107–13 114.

[33] Q. Zhang, P. Cui, D. Yan, J. Sun, Y. Duan, G. Han, W. Zhao, W. Zhang, Y. Guo, A. Zhang, *et al.*, "Whole-body humanoid robot locomotion with human reference," in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2024, pp. 11 225–11 231.

[34] M. Murooka, T. Hoshi, K. Fukumitsu, S. Masuda, M. Hamze, T. Sasaki, M. Morisawa, and E. Yoshida, "Tact: Humanoid whole-body contact manipulation through deep imitation learning with tactile modality," *IEEE Robotics and Automation Letters*, 2025.

[35] 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.

[36] J. Liu, H. Sim, C. Li, K. C. Tan, and F. Chen, "Birp: Learning robot generalized bimanual coordination using relative parameterization method on human demonstration," in *2023 62nd IEEE Conference on Decision and Control (CDC)*. IEEE, 2023, pp. 8300–8305.

[37] Z. Xie, J. Tseng, S. Starke, M. van de Panne, and C. K. Liu, "Hierarchical planning and control for box loco-manipulation," *Proceedings of the ACM on Computer Graphics and Interactive Techniques*, vol. 6, no. 3, pp. 1–18, 2023.

[38] 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.

[39] R. P. Singh, M. Morisawa, M. Benallegue, Z. Xie, and F. Kanehiro, "Robust humanoid walking on compliant and uneven terrain with deep reinforcement learning," in *2024 IEEE-RAS 23rd International Conference on Humanoid Robots (Humanoids)*. IEEE, 2024, pp. 497–504.

[40] 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.

[41] R. P. singh, Z. Xie, P. Gergondet, and F. Kanehiro, "Learning bipedal walking for humanoids with current feedback," *IEEE Access*, vol. 11, pp. 82 013–82 023, 2023.

[42] P. Moritz, R. Nishihara, S. Wang, A. Tumanov, R. Liaw, E. Liang, M. Elibol, Z. Yang, W. Paul, M. I. Jordan, *et al.*, "Ray: A distributed framework for emerging {AI} applications," in *13th USENIX symposium on operating systems design and implementation (OSDI 18)*, 2018, pp. 561–577.

[43] L. E. Dubins, "On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents," *American Journal of mathematics*, vol. 79, no. 3, pp. 497–516, 1957.

[44] O. developers, "Onnx," https://github.com/onnx/onnx and https://github.com/microsoft/onnxruntime, 2018.