

# Opt2Skill: Imitating Dynamically-Feasible Whole-Body Trajectories for Versatile Humanoid Loco-Manipulation

Fukang Liu<sup>1b</sup>, Zhaoyuan Gu<sup>1b</sup>, Yilin Cai<sup>1b</sup>, Ziyi Zhou<sup>1b</sup>, *Graduate Student Member, IEEE*, Hyunyoung Jung<sup>1b</sup>, Jaehwi Jang, Shijie Zhao, Sehoon Ha<sup>1b</sup>, *Member, IEEE*, Yue Chen<sup>1b</sup>, *Member, IEEE*, Danfei Xu<sup>1b</sup>, and Ye Zhao<sup>1b</sup>, *Senior Member, IEEE*

**Abstract**—Humanoid robots are designed to perform diverse loco-manipulation tasks. However, they face challenges due to their high-dimensional and unstable dynamics, as well as the complex contact-rich nature of the tasks. Model-based optimal control methods offer flexibility to define precise motion but are limited by high computational complexity and accurate contact sensing. On the other hand, reinforcement learning (RL) handles high-dimensional spaces with strong robustness but suffers from inefficient learning, unnatural motion, and sim-to-real gaps. To address these challenges, we introduce Opt2Skill, an end-to-end pipeline that combines model-based trajectory optimization with RL to achieve robust whole-body loco-manipulation. Opt2Skill generates dynamic feasible and contact-consistent reference motions for the Digit humanoid robot using differential dynamic programming (DDP) and trains RL policies to track these optimal trajectories. Our results demonstrate that Opt2Skill outperforms baselines that rely on human demonstrations and inverse kinematics-based references, both in motion tracking and task success rates. Furthermore, we show that incorporating trajectories with torque information improves contact force tracking in contact-involved tasks, such as wiping a table.

**Index Terms**—Humanoid and bipedal locomotion, reinforcement learning, whole-body motion planning and control.

## I. INTRODUCTION

**H**UMANOID robots possess inherent advantages in achieving human-like behaviors [1]. Their anthropomorphic morphology makes them well-suited for human-centered environments and offers the potential to learn from human loco-manipulation behaviors, such as transporting bulky objects, making safe contact with the environment, and performing agile skills such as jumping. Controlling humanoid robots, however,

Received 15 June 2025; accepted 24 September 2025. Date of publication 13 October 2025; date of current version 23 October 2025. This article was recommended for publication by Associate Editor E. Mingo Hoffman and Editor A. Kheddar upon evaluation of the reviewers' comments. This work was supported in part by the Office of Naval Research (ONR) under Grant N000142312223, in part by the National Science Foundation (NSF) under Grant CMMI-2144309 and Grant 2339076, and in part by the United States Department of Agriculture (USDA) under Grant 2022-11065. (*Corresponding author: Ye Zhao.*)

The authors are with the Institute for Robotics and Intelligent Machines, Georgia Institute of Technology, Atlanta, GA 30332 USA (e-mail: yezhao@gatech.edu).

We have successfully transferred our approach to real-world applications. <https://opt2skill.github.io>

This article has supplementary downloadable material available at <https://doi.org/10.1109/LRA.2025.3620620>, provided by the authors.

Digital Object Identifier 10.1109/LRA.2025.3620620

still faces challenges due to their high-dimensional, underactuated nature and the complexities of contact-rich interactions.

Model-based methods that rely on an accurate model for control have been foundational in enabling humanoid skills. However, their reliability in multi-contact scenarios often limits them to behaviors with simple, pre-defined contacts. In contrast, model-free RL that trains skills from scratch achieves robustness to complex contact modes even with environment uncertainties. However, it requires heavy reward tuning to regulate the policy behavior and to cross the sim-to-real gap. As a result, providing RL with a reference has become a mainstream approach. For example, learning loco-manipulation by imitating human motions has shown success on humanoid hardware [2], [3], [4]. However, acquiring such data in high quality and for diverse tasks still poses a challenge. Furthermore, these datasets require motion retargeting, and the retargeted motions may not always be kinematically or dynamically feasible, limiting the robot's ability to imitate the original motions effectively, and potentially losing behavior versatility to some extent. Other than using human data, complementing RL with model-based methods for whole-body control of legged robots is also a growing trend [5]. Along this line, learning dynamically-feasible whole-body motions from model-based methods to achieve diverse humanoid loco-manipulation tasks remains an open problem.

This presents **Opt2Skill**, a scalable learning pipeline that transforms whole-body reference motions from trajectory optimization (TO) into humanoid loco-manipulation skills via sim-to-real RL. Our approach leverages differential dynamic programming (DDP) [6], [7] to generate whole-body motions that obey the robot's dynamics and task requirements. Using Opt2Skill, We train RL policies to accurately track these optimal and dynamically feasible motions. Opt2Skill enables a diverse set of loco-manipulation tasks, including bulky-object handling, door traversing, desk-object manipulation, and locomotion over uneven terrains such as stairs, as shown in Fig. 1. Our study indicates that Opt2Skill outperforms policies trained with human data or inverse-kinematics-based references in both tracking accuracy and task success rate. We further show that incorporating joint torque information improves contact force tracking in contact-rich scenarios. Finally, Opt2Skill policies transfer successfully to hardware without online trajectory adaptation. The contributions of this study are listed as:

- This work represents the first step to adopting full-order-dynamics-based TO to guide RL that achieves humanoid loco-manipulation tasks.

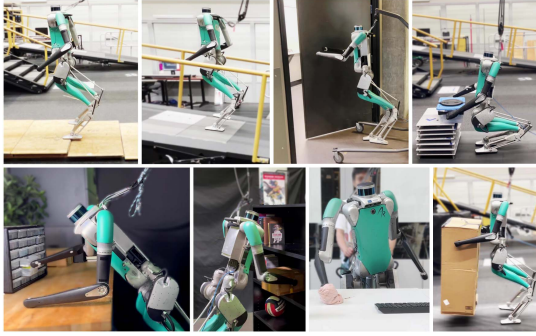


Fig. 1. The proposed Opt2Skill framework enables a Digit humanoid robot to perform various loco-manipulation tasks by mimicking optimal model-based reference trajectories in real-world scenarios.

- We show that full-body TO provides higher-quality motion data than mocap and IK baselines, leading to better motion tracking accuracy and task success. We further demonstrate that joint torque information—only available from TO—enhances tracking performance in contact-rich scenarios.
- We demonstrate the capability of our framework via successful sim-to-real transfer across diverse humanoid loco-manipulation tasks, including multi-contact whole-body manipulation and robust locomotion over stairs and in outdoor environments.

## II. RELATED WORK

### A. Model-Based Trajectory Optimization for Humanoids

Model-based motion planning has been crucial in advancing the capabilities of dynamic humanoid robots, enabling accurate loco-manipulation behaviors [1]. A popular model-based approach formulates motion planning as an optimal control problem (OCP), incorporating adjustable objectives and constraints to generate versatile motions [8].

Model predictive control (MPC) widely uses OCP for motion planning in an online receding horizon fashion. However, due to the computation constraint, MPC for humanoids predominantly uses various simplified models [9], [10], which result in conservative motions [11]. MPC that incorporates full-body models [12], [13] or kino-dynamic models [14], [15] still require high computational power and precise contact state feedback, which poses challenges for whole-body planning in real-time, contact-rich loco-manipulation tasks. Alternatively, TO can synthesize high-quality trajectories that are tracked by whole-body controllers. However, model-based whole-body controllers are sensitive to uncertainty in robot dynamics and environment contacts.

While model-based optimization approaches offer the advantage of precision, flexibility, and systematic motion generation for legged robots, they remain a challenging problem for humanoid whole-body loco-manipulation tasks due to the robot's complex, high-dimensional dynamics and the contact-rich, versatile nature of these tasks.

### B. Reinforcement Learning for Humanoids

Reinforcement learning (RL) enables real-time whole-body control of humanoid robots, offering robustness to environmental uncertainty [16], [17]. RL policies are trained through direct

interactions with the environment, providing a straightforward training paradigm that learns from scratch without requiring pre-planned motions. However, such RL training often suffers from poor data efficiency. Furthermore, policies learned from scratch can produce unnatural motions [18] and require significant efforts in reward shaping.

To address these challenges, RL-based motion imitation has been widely explored to enable humanoid loco-manipulation skills [2]. Compared with RL learned from scratch, motion imitation simplifies reward tuning and improves data efficiency, yet results in natural motions. Therefore, there is a growing trend in combining learning-based approaches with motion references for whole-body control of legged robots [5], [19]. Motion references come from various sources. While manually designed references can yield compelling behaviors [17], [20], [21], they are often task-specific and do not generalize to diverse or versatile motions. Human motion capture data is another primary source [1], and has been successfully transferred to humanoid hardware via RL-based imitation [3], [22], enabling versatile teleoperation [4] and autonomous skills [23]. However, a significant embodiment gap remains between humans and humanoids, necessitating nontrivial motion retargeting to adapt human movements to robot morphology. Thus, the embodiment gap may limit robot's ability to faithfully imitate original whole-body motions and reduce behavioral versatility.

To address these challenges, recent studies have begun using model-based TO to generate data for motion imitation. Model-based TO produces high-quality motion data by ensuring that trajectories adhere to robot dynamics and other physical constraints, such as joint torque limits. TO-based motion imitation has been widely employed on quadrupeds [5], [24], [25]. For humanoids, various TO models have been leveraged, including single rigid-body model [26], centroidal dynamics model [27], kino-dynamic model [28], and full-order-dynamics model [29], [30]. However, most studies focus on only locomotion [31]. Instead, this work focuses particularly on humanoid loco-manipulation. To achieve this, our TO specifies not only the lower-body leg motions but also the upper-body contact sequence for object manipulation using full-order dynamics. Therefore, our reference trajectories are no longer restricted to periodic motions, highlighting the capability of learning versatile behaviors. Moreover, our work is the first to demonstrate hardware success in policies learned from full-order-dynamics-based TO on a humanoid robot. For loco-manipulation tasks, the high fidelity of the full-order model is beneficial as it eliminates retargeting and is easier to obtain compared to teleoperation. Furthermore, full-order motion data includes joint torques, distinguishing it from other sources such as human motion capture or reduced-order motion. We demonstrate that joint torque information is critical for learning high-dimensional contact-rich skills.

This work adopts a full-order-dynamics-based TO to guide RL. A closely related study is [25], which also uses TO-based torque supervision for RL. However, our approach differs in several important aspects: we focus on whole-body loco-manipulation using a 30-DOF bipedal humanoid, while [25] focuses on quadrupedal locomotion with an 8-DOF Solo8 robot; we generate trajectories using a full-order dynamics model, whereas they use a simplified rigid-body model; and our work includes extensive simulation and hardware evaluations across seven diverse tasks, along with quantitative analysis of sim-to-real transfer, the impact of torque/force reference information,

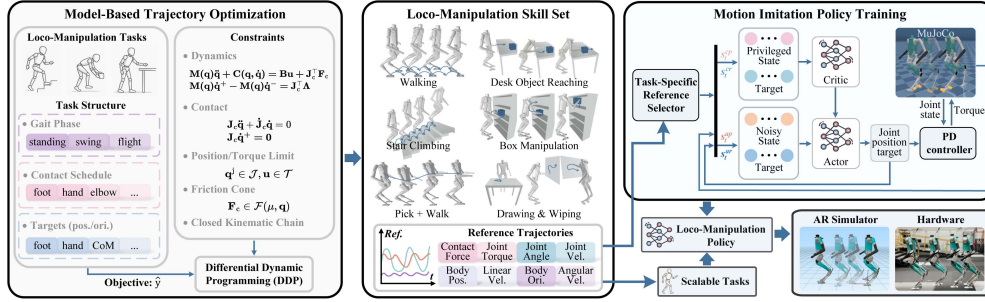


Fig. 2. Overall structure of the Opt2Skill framework. (a) We first generate structured, dynamically feasible reference trajectories using trajectory optimization with contact constraints, torque limits, and task-specific objectives. (b) Each trajectory contains joint angles, joint velocities, body position, orientation, linear and angular velocities, and dynamic-relevant quantities such as joint torques and interaction forces. (c) These trajectories serve as supervision signals to train RL policies that predict joint-level targets, tracked by a low-level PD controller. The resulting policies internalize control strategies grounded in model-based optimization while remaining reactive and robust to disturbances, sensor noise, and dynamics variability, enabling direct deployment to real hardware.

and comparisons to other motion sources such as inverse kinematics and human demonstrations.

### III. METHODS

Opt2Skill aims to develop an RL-based whole-body controller that enables a humanoid robot to track model-based optimal trajectories. These trajectories contain valuable torque reference that enables a high success rate in multi-contact loco-manipulation tasks. The overall framework is illustrated in Fig. 2. We begin by generating dynamically feasible and task-specific motions using DDP in Crocoddyl [32]. These trajectories serve as high-quality reference motions that encode contact-rich dynamics, torque limits, and task-solving strategies. We generate a diverse set of such trajectories offline and use them to guide the training of RL policies that directly predict joint-level target positions. By leveraging physically consistent references, the policy learns robust control strategies that track diverse reference motions sampled offline and transfer effectively to real hardware.

#### A. Whole-Body Trajectory Optimization

Consider the standard floating-base model of a humanoid robot, with an unactuated 6-DoF base and a set of  $n$ -DoF fully-actuated limbs. The equations of motion are given by:

$$\mathbf{M}(\mathbf{q})\dot{\mathbf{v}} + \mathbf{C}(\mathbf{q}, \mathbf{v}) = \underbrace{\begin{bmatrix} \mathbf{0} \\ \mathbf{I} \end{bmatrix}}_{\mathbf{B}} \mathbf{u} + \mathbf{J}_c^T \mathbf{F}_c \quad (1)$$

where  $\mathbf{q} = [\mathbf{q}^b; \mathbf{q}^j] \in \mathbb{R}^{n_q}$  and  $\mathbf{v} = [\mathbf{v}^b; \dot{\mathbf{q}}^j] \in \mathbb{R}^{n_v}$  are the generalized coordinates and velocities, partitioned into base (b) and joint (j) components. The base coordinates  $\mathbf{q}^b = [\mathbf{p}^b; \boldsymbol{\theta}^b] \in SE(3)$  include base position and orientation represented as unit quaternion, while the base velocities  $\mathbf{v}^b = [\mathbf{p}^b; \boldsymbol{\omega}^b] \in se(3)$  are linear and angular velocities in the body frame.  $\mathbf{M}(\mathbf{q})$  is the joint-space mass matrix, and  $\mathbf{C}(\mathbf{q}, \mathbf{v})$  captures the nonlinear effects.  $\mathbf{u} \in \mathbb{R}^{n_j}$  is the joint torque command,  $\mathbf{J}_c(\mathbf{q})$  is the contact Jacobian, and  $\mathbf{F}_c$  is the stacked contact reaction force vector. Given this whole-body dynamics, the TO formulation for generating a whole-body trajectory is written as:

$$\min_{\mathbf{x}, \mathbf{u}} \sum_{k=0}^{N-1} (\|\mathbf{y}[k] - \hat{\mathbf{y}}[k]\|_Q^2 + \|\mathbf{u}[k]\|_R^2) + \quad (2a)$$

$$\|\mathbf{y}[N] - \hat{\mathbf{y}}[N]\|_Q^2$$

subject to

$$\text{(Dynamics)} \quad \begin{cases} \mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{B}\mathbf{u} + \mathbf{J}_c^T \mathbf{F}_c \\ \mathbf{M}(\mathbf{q})\dot{\mathbf{q}}^+ - \mathbf{M}(\mathbf{q})\dot{\mathbf{q}}^- = \mathbf{J}_c^T \boldsymbol{\Lambda} \end{cases} \quad (2b)$$

$$\text{(Contact)} \quad \begin{cases} \mathbf{J}_c \ddot{\mathbf{q}} + \dot{\mathbf{J}}_c \dot{\mathbf{q}} = 0 \\ \mathbf{J}_c \dot{\mathbf{q}}^+ = 0 \end{cases} \quad (2c)$$

$$\text{(Limits)} \quad \mathbf{q}^j \in \mathcal{J}, \mathbf{u} \in \mathcal{T} \quad (2d)$$

$$\text{(Friction)} \quad \mathbf{F}_c \in \mathcal{F}(\mu, \mathbf{q}) \quad (2e)$$

where the decision variables include the full-body state vector  $\mathbf{x} = [\mathbf{q}; \dot{\mathbf{q}}] \in \mathbb{R}^{n_q + n_v}$  and the joint torque  $\mathbf{u}$ . The task space variables  $\mathbf{y} = \Phi(\mathbf{q})$  and  $\hat{\mathbf{y}}$  are defined based on the specific task. In (2b), the hybrid dynamics constraint includes either the continuous whole-body dynamics (first row), or the impact dynamics (second row), where  $\boldsymbol{\Lambda}$  denotes the contact impulse, and  $\dot{\mathbf{q}}^-$  and  $\dot{\mathbf{q}}^+$  are instantaneous velocities before and after the impact. Following (2b), the contact constraints (2c) is added for the stance or impact foot assuming a rigid contact, while a zero force is applied to the non-contact foot. Joint limit  $\mathcal{J}$ , torque limit  $\mathcal{T}$ , and friction cone  $\mathcal{F}$  with coefficient  $\mu$  are imposed as constraints in (2d)–(2e).

Considering computational efficiency, we employ a DDP-based method implemented in Crocoddyl [32] to solve whole-body optimization. To define the desired trajectory  $\hat{\mathbf{y}}$  in (2a), we use task-specific strategies. For tasks requiring precise foot tracking, such as walking or stair climbing, we specify  $\hat{\mathbf{y}}[k]$  at each timestep to enforce desired end-effector trajectories. For more flexible behaviors, such as box pickup or desk-object manipulation, we define sparse subgoals and allow the optimizer to generate smooth transitions toward those targets. In all cases, task-specific costs and contact schedules are predefined to produce feasible motions under full-body dynamics. The optimized trajectories include joint and base states, torques, and end-effector motions, which are extracted as reference data for downstream policy training.

To support robust policy learning across a wide range of loco-manipulation behaviors, we generate reference trajectories for diverse tasks, including *walking*, *stair climbing*, *object pickup*, *drawing*, and *desk-object reaching*. For each task, we systematically vary key motion parameters such as *gait phase*, *contact mode*, *walking speed*, *foot clearance*, *center of mass position*, and *target object location*. Leveraging the flexibility of our TO framework, we can efficiently sample these parameters to produce large datasets of motions. Unlike methods based

solely on human demonstrations or heuristic design, our approach generates dynamically feasible trajectories, which contain critical dynamics-relevant properties such as joint torques and interaction forces that are often missing from kinematic demonstrations. Detailed task-specific definitions of  $\hat{\mathbf{y}}$  and trajectory examples are presented in Section IV.

### B. RL-Based Imitation of Dynamically Feasible Trajectories

We leverage TO to generate reference motions that satisfy full-body dynamics, contact constraints, and torque limits. These trajectories include joint torques and contact forces, which are critical for learning contact-rich loco-manipulation skills. We incorporate this information directly into the policy observations and reward design, helping the policy learn robust and transferable whole-body behaviors.

During training, one trajectory is randomly sampled at the start of each episode to initialize the environment and serve as the reference for computing tracking rewards. We train separate policies for each loco-manipulation task, each using its own TO-generated trajectory dataset. At deployment, the policy is provided with partial references from a new offline-generated TO trajectory tailored to the specific test scenario, without reusing training data or running TO online.

While DDP (used within TO) produces dynamically feasible nominal trajectories, it is rarely employed directly for control due to modeling inaccuracies, contact uncertainties, and high computational cost in real-time settings. To address this, we leverage RL policies that track these trajectories and adapt to real-world disturbances during execution.

1) *Problem Formulation*: We formulate the control problem as an RL problem under a Markov Decision Process (MDP), defined by the tuple  $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma)$ , with state space  $\mathcal{S}$ , action space  $\mathcal{A}$ , transition dynamics  $\mathcal{P}$ , reward function  $\mathcal{R}$ , and discount factor  $\gamma$ . At each timestep  $t$ , the agent observes a state  $s_t \in \mathcal{S}$ , selects an action  $a_t \sim \pi(a|s_t)$ , receives a reward  $r_t$ , and transitions to a new state  $s_{t+1} \sim \mathcal{P}(\cdot|s_t, a_t)$ . In real-world deployments, the agent operates under partial observability due to sensor noise and unobservable environmental factors, resulting in a Partially Observable Markov Decision Process (POMDP). The policy  $\pi(a|o_t)$  must act based on noisy and incomplete observations  $o_t \in \mathcal{O}$ , which are incomplete projections of the true state  $s_t$ . Our goal is to learn an optimal policy  $\pi$  that maximizes the expected discounted return  $\mathbb{E}[\sum_{t=0}^T \gamma^t r_t]$ , where the reward encourages accurate tracking of reference trajectories generated via TO.

We train the RL policy in the MuJoCo Playground simulator [33] using an Actor-Critic (AC) architecture optimized by Proximal Policy Optimization (PPO) [34]. Both the actor and critic networks are composed of a Multi-Layer Perceptron MLP with hidden sizes of [512, 512, 256, 256]. Each policy is trained for fewer than 300 million steps, requiring under 7 hours on a single NVIDIA RTX4090 GPU.

2) *Imitation Policy*: To enable robust deployment on real hardware, the agent must make decisions based on noisy sensor measurements rather than true state information. To address this challenge, we adopt an asymmetric actor-critic framework [35], where the critic has access to privileged state information in simulation, while the actor observes only noisy proprioceptive inputs available during deployment.

The observation space for the critic is defined as  $\mathbf{o}_{\text{critic}} = [\mathbf{s}_t^{\text{cp}}; \mathbf{s}_t^{\text{cr}}]$ , where the privileged proprioception  $\mathbf{s}_t^{\text{cp}}$  is  $[\mathbf{p}_t^b, \boldsymbol{\theta}_t^b, \dot{\mathbf{p}}_t^b, \boldsymbol{\omega}_t^b, \mathbf{g}_t, \mathbf{q}_{\text{hist}}^j, \hat{\mathbf{q}}_t^j, \mathbf{p}_t^e, \mathbf{a}_{\text{hist}}, \mathbf{F}_t^c, \mathbf{u}_t, \mathbf{K}_p, \mathbf{K}_d]$ ,

TABLE I  
REWARD COMPONENTS AND WEIGHTS

Category	Term	Expression	Weight
Task Reward	Joint Pos.	$\exp(-5\ \hat{\mathbf{q}}_t^j - \mathbf{q}_t^j\ _2^2)$	0.30
	Base Pos.	$\exp(-20\ \hat{\mathbf{p}}_t^b - \mathbf{p}_t^b\ _2^2)$	0.30
	Base Ori.	$\exp(-50\ \hat{\boldsymbol{\theta}}_t^b - \boldsymbol{\theta}_t^b\ _2^2)$	0.30
	Base Lin. Vel.	$\exp(-2\ \hat{\dot{\mathbf{p}}}_t^b - \dot{\mathbf{p}}_t^b\ _2^2)$	0.30
	Base Ang. Vel.	$\exp(-0.5\ \hat{\boldsymbol{\omega}}_t^b - \boldsymbol{\omega}_t^b\ _2^2)$	0.30
	End-effector Pos.	$\exp(-20\ \hat{\mathbf{p}}_t^e - \mathbf{p}_t^e\ _2^2)$	0.30
	Joint Torque	$\exp(-0.01\ \hat{\mathbf{u}}_t - \mathbf{u}_t\ _2^2)$	0.10
	Contact Force	$\exp(-0.05\ \hat{\mathbf{F}}_t^c - \mathbf{F}_t^c\ _1)$	0.10
Penalty Cost	Action Rate	$\ \mathbf{a}_t - 2\mathbf{a}_{t-1} + \mathbf{a}_{t-2}\ _2^2$	-0.05
	Torques	$\ \mathbf{u}_t / \mathbf{u}_{\text{limit}}\ _2^2$	-0.03
	Joint Acc.	$\ \hat{\mathbf{q}}_t^j\ _2^2$	-10 <sup>-6</sup>

Note that the "Base Pos." reward is not included in the "walking" task.

TABLE II  
DOMAIN RANDOMIZATION PARAMETERS

Category	Parameter	Type	Range / Std
Obs.	Joint Pos.	Additive (Gauss)	$\sigma = 0.0875$
	Joint Vel.	Additive (Gauss)	$\sigma = 0.075$
	Base Lin. Vel.	Additive (Gauss)	$\sigma = 0.15$
	Base Ang. Vel.	Additive (Gauss)	$\sigma = 0.15$
	Gravity Proj.	Additive (Gauss)	$\sigma = 0.075$
Delays	Action Delay	Uniform	[0.0, 0.02] s
Motor	Motor Strength	Scaling (Uniform)	[0.95, 1.05]
	Kp/Kd Factor	Scaling (Uniform)	[0.9, 1.1]
Body	Mass	Scaling (Uniform)	[0.9, 1.1]
Env.	Gravity	Scaling (Uniform)	[0.9, 1.1]
	Friction	Scaling (Uniform)	[0.3, 1.0]
	Terrain	Discrete	flat, rough

which includes the humanoid body translation  $\mathbf{p}_t^b$ , orientation  $\boldsymbol{\theta}_t^b$ , linear velocity  $\dot{\mathbf{p}}_t^b$ , angular velocity  $\boldsymbol{\omega}_t^b$ , projected gravity  $\mathbf{g}_t$  (a proxy for base orientation), a history of  $N = 10$  past motor joint positions sampled every  $\delta = 4$  timesteps,  $\mathbf{q}_{\text{hist}}^j = [\mathbf{q}_t^j, \mathbf{q}_{t-\delta}^j, \dots, \mathbf{q}_{t-(N-1)\delta}^j]$  (i.e., 50 Hz sampling from a 200 Hz control loop), motor joint velocities  $\dot{\mathbf{q}}_t^j$ , end-effector position (relative to the torso)  $\mathbf{p}_t^e$ , a history of  $N$  past actions sampled with the same schedule,  $\mathbf{a}_{\text{hist}} = [\mathbf{a}_{t-1}, \mathbf{a}_{t-(1+\delta)}, \dots, \mathbf{a}_{t-(1+(N-1)\delta)}]$ , contact force  $\mathbf{F}_t^c$ , joint torque  $\mathbf{u}_t$ , and the PD gains  $\mathbf{K}_p, \mathbf{K}_d$ . The reference state  $\mathbf{s}_t^{\text{cr}}$  is defined as  $[\hat{\mathbf{p}}_t^b, \hat{\boldsymbol{\theta}}_t^b, \hat{\dot{\mathbf{p}}}_t^b, \hat{\boldsymbol{\omega}}_t^b, \hat{\mathbf{q}}_t^j, \hat{\dot{\mathbf{q}}}_t^j, \hat{\mathbf{p}}_t^e, \hat{\mathbf{F}}_t^c, \hat{\mathbf{u}}_t]$ , where the hat ( $\hat{\cdot}$ ) denotes reference trajectory information.

The observation space for the actor is defined as  $\mathbf{o}_{\text{actor}} = [\mathbf{s}_t^{\text{ap}}, \mathbf{s}_t^{\text{ar}}]$ , where the noisy proprioception  $\mathbf{s}_t^{\text{ap}}$  is  $[\tilde{\mathbf{p}}_t^b, \tilde{\boldsymbol{\omega}}_t^b, \tilde{\mathbf{g}}_t, \tilde{\mathbf{q}}_{\text{hist}}^j, \tilde{\dot{\mathbf{q}}}_t^j, \mathbf{a}_{\text{hist}}]$ , where the tilde ( $\tilde{\cdot}$ ) indicates noisy sensor measurements. The actor also receives partial reference information  $\mathbf{s}_t^{\text{ar}} = [\hat{\mathbf{p}}_t^b, \hat{\boldsymbol{\omega}}_t^b, \hat{\mathbf{q}}_t^j, \hat{\mathbf{F}}_t^c, \hat{\mathbf{u}}_t]$ . The actor receives partial references to ensure generalization and avoid reliance on noisy or redundant inputs such as global base states or end-effector poses, while the critic has access to full privileged information for stable value estimation.

The action space of the control policy represents offsets from a default standing pose to specify target positions for the 20 actuated joints. These targets are fed into a PD joint torque controller, which computes torque as  $\mathbf{u}_t = \mathbf{K}_p(\mathbf{a}_t + \mathbf{q}_{\text{dflt}}^j - \mathbf{q}_t^j) - \mathbf{K}_d\dot{\mathbf{q}}_t^j$ . Here,  $\mathbf{q}_{\text{dflt}}^j$  denotes the default standing joint positions, and  $\mathbf{q}_t^j$  and  $\dot{\mathbf{q}}_t^j$  are the measured joint positions and velocities. Our control policy runs at 200 Hz, while the internal PD controller operates at 1 kHz in simulation (during training) and 2 kHz on

hardware (during deployment). No explicit filtering is applied between the policy output and the PD control loop. In hardware, we use the same PD gains as in simulation and send torque commands directly to the robot.

3) *Reward Functions, Domain Randomization, and Curriculum*: We design reward functions to guide the robot to track TO-based reference motions and accomplish diverse tasks across various scenarios. The rewards consist of two parts (inspired by [2], [36]): task and regularization rewards, as detailed in Table I. The task rewards guide the humanoid to track commands such as target velocity, end-effector positions, pickup targets, and contact events for loco-manipulation. Our approach leverages the advantages of the Crocodyl TO solver to explicitly generate parameterized foot trajectories with controllable step height and timing. These structured, dynamically feasible references improve end-effector tracking and eliminate the need for heuristic reward shaping.

In addition, task rewards such as joint torque and contact force tracking improve the policy’s ability to maintain stable contact and adapt to variations in contact conditions during contact-rich loco-manipulation. Regularization rewards promote smooth, physically plausible, and energy-efficient motions by penalizing large torques and abrupt actions, improving robustness and facilitating sim-to-real transfer.

To facilitate sim-to-real transfer, we apply domain randomization during training, including robot dynamics, observation noise, control delays, and external disturbances, inspired by [36] and summarized in Table II. We also employ a noise curriculum that gradually increases observation noise and penalty strength. This helps the policy learn more stably in early stages and improves generalization in later stages.

## IV. RESULTS

### A. Robot System

Digit is a general-purpose bipedal humanoid robot developed by Agility Robotics. It weighs approximately 48 kg and has 30 degrees of freedom (DoF), including 20 actuated joints—four in each arm and six in each leg. Passive joints in the ankle and shin are connected via leaf spring four-bar linkages. The toe joints (pitch and roll) are indirectly actuated through rods driven by two motors. We use a high-fidelity physics-based simulator from Agility Robotics to evaluate the trained policies before hardware deployment. The simulator provides realistic dynamics and physical properties of Digit and shares many features with the hardware.

### B. Comparison With Different Datasets

To evaluate the effectiveness of Opt2Skill, we compare our TO method against two alternative motion sources:

- *Human-retargeted motions (Human)*: Obtained by extracting end-effectors and torso motions from human demonstrations, with joint angles obtained through Quadratic Programming (QP)-based IK.
- *IK-based trajectories (IK)*: Generated from hand-designed keypoint sequences (e.g., footstep and torso motion), converted to joint angles via QP-based IK.

1) *Dataset Generation*: The human-retargeted dataset is selected from the AMASS [37] motion capture dataset, where we choose approximately 70 trajectories containing non-aggressive forward walking motions. Before applying IK, we calibrate the body shape parameters to align with the morphology of Digit. In

addition, we scale the torso speeds to match the new leg length. The torso speed ranges from 0 to 1.5 m/s, with foot clearance between 10 and 20 cm.

To ensure a fair comparison, both IK and TO datasets are generated with matched speed and clearance ranges. The IK dataset is created by specifying target keypoints (feet, hand, torso) and solving IK at each timestep. The TO dataset uses the same target keypoints but solves a whole-body trajectory using DDP. Both the IK and TO datasets contain 2000 trajectories, all generated on flat ground. While the human-retargeted dataset is smaller due to difficulty in generating high-quality retargeted trajectories for the robot, the range of motion parameters is matched across datasets.

2) *Evaluation Metrics*: We evaluate both *per-frame tracking* and *long-term drift*. Per-frame metrics include the mean joint angle error  $E_{\text{joint}}$  (rad), base linear velocity error  $E_{\text{vel-x}}$  (cm/s), and yaw angular velocity error  $E_{\text{vel-yaw}}$  (rad/s). We also report hand and foot end-effector tracking errors  $E_{\text{ee-hand}}$  and  $E_{\text{ee-foot}}$  (cm), which reflect local task-space accuracy. To assess long-term consistency, we report global drift metrics: the normalized final base position drift  $E_{\text{pos-x}}$  (%) along the walking direction relative to the reference path length, and the average per-step drift in lateral base position  $E_{\text{pos-y}}$  (mm/step) and yaw orientation  $E_{\text{yaw}}$  (mrad/step), which reflect accumulated deviations over time.

Additionally, to evaluate the adaptability of the learned policies and datasets to rough terrain, we compare their success rates on two challenging scenarios: stairs with varying heights and slopes with increasing inclinations.

3) *Tracking Performance*: Table III presents the tracking performance of policies trained with different reference datasets. We evaluate 20 trajectories per policy. TO and IK trajectories are approximately 14 seconds long, while human trajectories range from 5–6 seconds. The reported metrics are averaged over the full trajectory duration. Overall, Opt2Skill achieves the best tracking accuracy across most metrics.

Specifically, Opt2Skill achieves an average hand tracking error of 2.00 cm and foot tracking error of 5.23 cm. In contrast, the human-retargeted policy shows higher errors of 4.25 cm (hand) and 9.42 cm (foot), while the IK-based policy yields 5.47 cm (hand) and 5.24 cm (foot). Regarding long-term consistency, Opt2Skill achieves the lowest global drift: 1.14% in base position, 0.06 mm/frame laterally, and 0.02 mrad/frame in yaw, clearly outperforming the human-retargeted and IK-based policies. Although the IK-based policy achieves the lowest joint angle error ( $E_{\text{joint}} = 0.02$  rad), its higher task-space tracking errors indicate a fundamental discrepancy between kinematic feasibility and dynamic consistency. These results show that physical feasibility and task adaptability of TO-based references contribute to both local tracking accuracy and long-term global stability. A Wilcoxon signed-rank test confirmed statistically significant differences in end-effector tracking performance between TO and IK/human datasets ( $p < 0.001$  for both comparisons).

We further analyze Opt2Skill’s performance under different observation history lengths (0, 5, 10, and 15 steps), as shown in Table III. The results indicate that incorporating a short to moderate observation history significantly improves tracking accuracy, while overly long histories can deteriorate policy performance. Overall, Opt2Skill with a 10-step history achieves the best overall accuracy.

4) *Task Success on Rough Terrain*: Fig. 3 shows success rates of the three policies across varying stair heights and slope angles in the Agility simulator. For each policy and each terrain

TABLE III  
TRACKING PERFORMANCE OF OPT2SKILL AND BASELINES (MEAN  $\pm$  SE; LOWER IS BETTER)

Category	Method	$E_{\text{joint}}$ (rad) $\downarrow$	$E_{\text{vel-x}}$ (cm/s) $\downarrow$	$E_{\text{vel-yaw}}$ (rad/s) $\downarrow$	$E_{\text{ee-hand}}$ (cm) $\downarrow$	$E_{\text{ee-foot}}$ (cm) $\downarrow$	$E_{\text{pos-x}}$ (%) $\downarrow$	$E_{\text{pos-y}}$ (mm/step) $\downarrow$	$E_{\text{yaw}}$ (mrad/step) $\downarrow$
Data	Opt2Skill (Ours)	$0.03 \pm 0.01$	<b><math>4.86 \pm 4.18</math></b>	<b><math>0.06 \pm 0.04</math></b>	<b><math>2.00 \pm 0.76</math></b>	<b><math>5.23 \pm 1.75</math></b>	<b><math>1.14 \pm 0.61</math></b>	<b><math>0.06 \pm 0.03</math></b>	<b><math>0.02 \pm 0.01</math></b>
	human-data	$0.04 \pm 0.03$	$6.68 \pm 8.96$	$0.06 \pm 0.08$	$4.25 \pm 3.49$	$9.42 \pm 8.19$	$4.49 \pm 7.35$	$0.11 \pm 0.18$	$0.04 \pm 0.04$
	IK-data	<b><math>0.02 \pm 0.01</math></b>	$7.10 \pm 5.73$	$0.06 \pm 0.05$	$5.47 \pm 2.58$	$5.24 \pm 1.54$	$2.09 \pm 0.27$	$0.10 \pm 0.02$	$0.06 \pm 0.01$
Ablation	Opt2Skill-hist15	<b><math>0.03 \pm 0.01</math></b>	$5.93 \pm 4.81$	<b><math>0.05 \pm 0.04</math></b>	$3.11 \pm 1.58$	$6.48 \pm 2.32$	$2.49 \pm 1.40$	$0.12 \pm 0.06$	$0.05 \pm 0.01$
	Opt2Skill-hist10	<b><math>0.03 \pm 0.01</math></b>	<b><math>4.86 \pm 4.18</math></b>	$0.06 \pm 0.04$	<b><math>2.00 \pm 0.76</math></b>	<b><math>5.23 \pm 1.75</math></b>	$1.14 \pm 0.61$	<b><math>0.06 \pm 0.03</math></b>	<b><math>0.02 \pm 0.01</math></b>
	Opt2Skill-hist5	$0.04 \pm 0.01$	$6.46 \pm 4.85$	$0.06 \pm 0.05$	$5.63 \pm 3.54$	$8.59 \pm 4.24$	<b><math>1.12 \pm 0.96</math></b>	$0.06 \pm 0.05$	$0.07 \pm 0.01$
	Opt2Skill-hist0	$0.04 \pm 0.02$	$7.82 \pm 5.46$	$0.10 \pm 0.09$	$6.68 \pm 4.20$	$8.31 \pm 4.60$	$1.32 \pm 0.66$	$0.07 \pm 0.04$	$0.07 \pm 0.02$

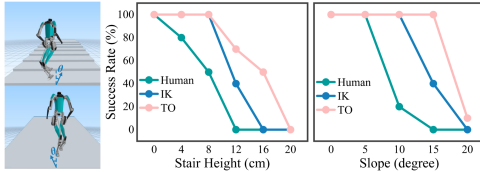


Fig. 3. Success rate comparison across different terrains. Policies trained on human-retargeted, IK-based, and TO-generated trajectories (with matched parameters) are evaluated on stair heights (left) and slope angles (right).

setting, we conduct 10 trials with difference initial yaw angles uniformly sampled from  $-25^\circ$  to  $25^\circ$  to evaluate robustness across diverse orientations ( $\theta$  in Fig. 3). Using similar trajectory parameters (base speed  $\sim 1$  m/s, foot clearance  $\sim 0.2$  m), Opt2Skill consistently outperforms the human-retargeted and IK-based baselines. Specifically, for stair climbing (Fig. 3, left), it maintains high success rates up to 12 cm steps, while the other policies drop sharply beyond 8 cm. Similarly, for slope traversal (Fig. 3, right), Opt2Skill shows higher robustness at steeper inclinations.

These results demonstrate that Opt2Skill-trained policies, even without any task-specific test-time tuning, exhibit strong robustness to varying terrain. This highlights the advantages of training with offline-generated, dynamically feasible trajectories compared to fixed human or purely kinematic references.

### C. Evaluating the Role of Torque Information in Contact-Rich Loco-Manipulation

We conduct additional simulation experiments to investigate the effect of torque information in contact-rich tasks. Specifically, we focus on a wiping task (see Fig. 2) that requires controlled contact force between the end effector and a desk surface. We evaluate four ablation baselines:

- **Pos** that tracks end-effector positions without any contact or torque information in the observation or reward.
- **Pos+F** that adds reference contact force to the observation and includes a contact force tracking reward.
- **Pos+T** that adds reference joint torques to the observation and includes a torque tracking reward.
- **Pos+F+T** that adds both reference contact force and joint torques in the observation, along with corresponding force and torque tracking rewards.

We train all policies using 1400 TO-generated trajectories with varying desk heights (0.85–0.95 m) and contact normal forces (0–20 N).

1) *Performance Analysis*: To evaluate the effectiveness of each policy, we conduct six groups of experiments with different reference contact forces levels, ranging from 0 to 25 N. Each group consists of 10 TO-generated trajectories with randomized desk heights.

TABLE IV  
TRACKING ERRORS UNDER DIFFERENT CONTACT FORCE REFERENCES

Category	Ref. Force	Pos	Pos+F	Pos+T	Pos+F+T
Force Err. (N)	0	$1.2 \pm 2.1$	<b><math>0.0 \pm 0.0</math></b>	<b><math>0.0 \pm 0.0</math></b>	<b><math>0.0 \pm 0.0</math></b>
	5	$4.1 \pm 1.7$	$5.0 \pm 0.4$	$3.6 \pm 1.7$	<b><math>1.1 \pm 1.4</math></b>
	10	$8.7 \pm 2.2$	$3.7 \pm 2.9$	$3.0 \pm 2.6$	<b><math>1.5 \pm 2.2</math></b>
	15	$13.4 \pm 2.5$	$2.3 \pm 2.4$	$2.9 \pm 3.1$	<b><math>1.2 \pm 2.7</math></b>
	20	$18.7 \pm 2.4$	$3.3 \pm 3.7$	$2.9 \pm 3.8$	<b><math>1.6 \pm 3.4</math></b>
	25	$23.9 \pm 2.4$	$2.8 \pm 4.3$	$3.0 \pm 4.5$	<b><math>2.0 \pm 3.7</math></b>
	Avg.	$11.7 \pm 8.3$	$2.8 \pm 3.2$	$2.7 \pm 3.2$	<b><math>1.5 \pm 2.6</math></b>
Pos. Err. (cm)	0	<b><math>2.4 \pm 1.7</math></b>	$3.8 \pm 2.4$	$2.9 \pm 1.5$	$2.6 \pm 0.8$
	5	<b><math>2.4 \pm 1.7</math></b>	$3.0 \pm 1.5$	$2.9 \pm 1.6$	$2.6 \pm 0.9$
	10	<b><math>2.4 \pm 1.7</math></b>	$3.0 \pm 1.3$	$2.8 \pm 1.7$	$3.2 \pm 1.4$
	15	<b><math>2.4 \pm 1.7</math></b>	$3.2 \pm 1.4$	$2.9 \pm 1.7$	$3.2 \pm 1.5$
	20	<b><math>2.4 \pm 1.7</math></b>	$3.8 \pm 1.5$	$3.0 \pm 1.6$	$2.9 \pm 1.3$
	25	<b><math>2.5 \pm 1.7</math></b>	$4.6 \pm 1.7$	$3.0 \pm 1.6$	$2.6 \pm 1.4$
	Avg.	<b><math>2.4 \pm 1.7</math></b>	$3.5 \pm 1.8$	$2.9 \pm 1.6$	$2.9 \pm 1.3$

Top: Contact force tracking error in Newtons. Bottom: Right hand position tracking error in centimeters. Results are averaged over 10 TO-generated trajectories per group with randomized desk heights (range: 0.85–0.95 m) for a wiping task.

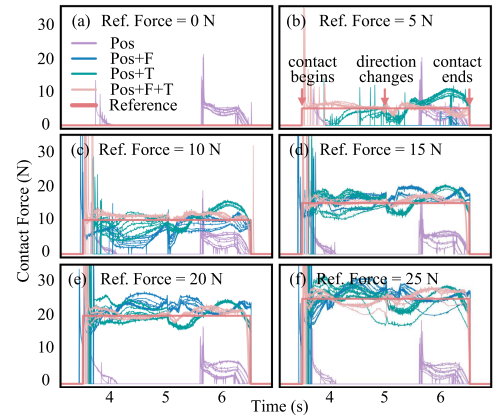


Fig. 4. Contact force profiles across all four policies under varying reference force levels. Each subplot shows 10 trajectories per policy.

Table IV shows the contact force and right end effector position tracking errors across varying contact force levels. The **Pos** policy achieves the best position tracking across all scenarios. This is expected, as it optimizes solely for position accuracy without being constrained by contact or torque objectives. However, it produces similar force patterns regardless of the reference force and either fails to generate sufficient contact force (Fig. 4(b)–(f)) or overshoots when no force is required (Fig. 4(a)), due to lack of contact awareness.

Adding reference contact force to the observation and reward in **Pos+F** improves average force tracking accuracy, but instability remains in certain conditions. For instance, in Fig. 4(b), **Pos+F** fails to produce the desired 5 N contact force. In contrast, **Pos+T**, which includes reference torques in the observation and

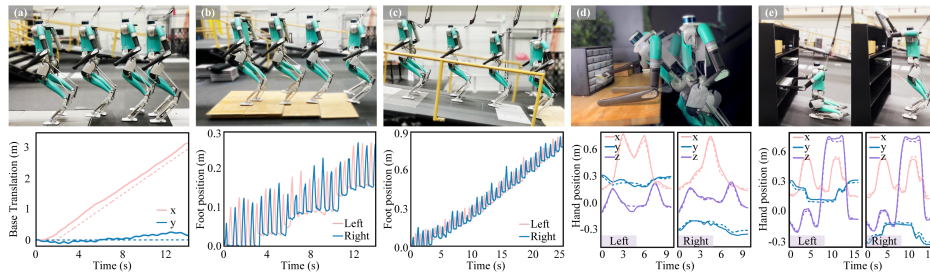


Fig. 5. Snapshots of hardware experiments with data plots. (a) Flat-ground locomotion with accurate base position tracking. (b)–(c) Walking up a stair and a ramp ( $\sim 19.5^\circ$ ) with foot  $z$ -position plots indicating elevation. (d) Desk object reaching, with plots of end-effector tracking. (e) Box pick-up from one shelf layer to another. Note that in (d) and (e), end-effector positions are expressed relative to the root. Dashed lines indicate the reference trajectories.

reward, shows improved contact force tracking (average error: 2.7 N) and better position tracking (average error: 2.9 cm) compared to **Pos+F** (Table IV). While the reference force in **Pos+F** helps the policy learn how much force to apply, the reference torque in **Pos+T** provides dynamically grounded information about *when* and *how* to apply it, resulting in more physically consistent behavior.

Finally, **Pos+F+T**, which combines both reference contact force and torque signals, achieves the best overall force tracking (Table IV). Its position tracking is also competitive with **Pos**, and it exhibits the most stable force application across all scenarios (Fig. 4). This highlights the complementary nature of contact force and torque signals, and the critical role of reference torque in enabling generalizable, physically grounded contact behaviors—information uniquely available in TO-generated datasets but absent in IK or human datasets. Moreover, policies trained with TO-generated references adapt when the environment deviates from nominal settings (e.g., table height). For instance, with a trajectory defined for 0.95 m height and 15 N contact force, the policy maintained wiping at 0.9 m and 1.0 m table heights, achieving average forces of  $9.6 \pm 1.5$  N and  $21.3 \pm 2.0$  N, and tracking errors of  $4.1 \pm 1.6$  cm and  $4.5 \pm 2.0$  cm, respectively. A Wilcoxon signed-rank test confirmed statistically significant differences in both contact force and hand position tracking errors between **Pos+F+T** and other baselines across all force levels ( $p < 0.001$ ).

The integration of force and torque signals significantly enhances tracking performance, a contribution we believe is essential for high-fidelity imitation of complex dynamic skills. However, we acknowledge this necessitates a more complex pipeline that requires an offline TO stage prior to policy training. We anticipate that future advances in sensing technology and force estimation will simplify the data collection process, further facilitating the adoption of such high-fidelity force signals.

#### D. Hardware Experiments

We demonstrate the sim-to-real performance of our Opt2Skill through five hardware experiments. Snapshots of the tasks and their corresponding data plots are shown in Fig. 5. We compare the robot’s measured states with the corresponding references to evaluate tracking performance. The robot states are obtained from onboard sensors and estimated using the built-in state estimator provided by Agility Robotics.

For locomotion tasks, Fig. 5(a) shows accurate base position tracking in the  $x$  and  $y$  directions during flat ground walking. We generate the reference trajectories by specifying a fixed gait frequency, step length, and foot clearance, using cubic-spline interpolation to compute desired foot trajectories  $\hat{\mathbf{y}}$  described in

Section III-A. Fig. 5(b) and (c) show the foot height as the robot successfully walks up a stair and a ramp. Notably, Opt2Skill achieves rough terrain walking capability despite the reference being designed only for flat ground scenarios. The robot also maintains toe compliance when stepping on stair edges and ramp slopes—enabled by low PD gains in the low-level controller.

Fig. 5(d) demonstrates a multi-contact whole-body manipulation task, where the robot reaches for an object on a desk located well beyond its support polygon. We generate these reference trajectories via motion phases: (1) one arm moves to the target position; (2) the elbow maintains contact with the desk; (3) the opposite arm reaches the target position and returns; (4) release elbow contact; (5) return torso and hands to the initial pose. These target positions serve as the desired trajectories  $\hat{\mathbf{y}}$  in Section III-A. To achieve this, the legs coordinate with upper-body motion to stabilize and enable the forward-reaching behavior, reflecting the necessity of whole-body motion coordination. This example highlights the ability of Opt2Skill to handle high-dimensional loco-manipulation tasks. In Fig. 5(e), the robot squats down to pick up a box from a lower shelf and place it on a higher one. Trajectories are generated by: (1) squat down; (2) both arms reach and squeeze to grasp the box; (3) retract arms slightly and stand up; (4) reach to the target and release; (5) return torso and hands to the initial pose. During training, hand targets range from  $[0.40, \pm 0.10, 0.55]$  m to  $[0.50, \pm 0.20, 1.70]$  m. In deployment (Fig. 5(e)), the hands reach 0.49 m (lower) and 1.76 m (upper), indicating that the policy can handle moderate variations beyond the training range.

The plots in Fig. 5(d) and (e) show the tracking performance of both hand positions during these complex motion sequences. In the *desk object pushing* task, Opt2Skill achieves an average tracking error of 3.7 cm for the left hand and 3.8 cm for the right. In the *shelf box pick-up* task, the errors are 3.6 cm for both hands. Notably, a deviation along the  $y$ -axis occurs when the robot makes contact with the box, caused by a size mismatch between the reference and the real-world box (approximately 2cm in the  $y$  direction for each hand).

Additionally, we conduct four more real-world tasks: *heavy box pickup*, *pickup+walk*, *door opening*, and *drawing+wiping*, as shown in Fig. 6. In the first task, the robot squats down and lifts a box weighing up to 4.9 kg, the trajectory is generated by predefining grasping goals followed by lifting goals for both hands. In the second task, the robot picks up a box from a standing pose and then walks forward with it, using a trajectory obtained by combining the pickup and locomotion tasks. In the third task, the robot pushes open two heavy fire doors using both arms while walking, with end-effector target positions added to the locomotion trajectory. Finally, the robot uses its right hand to draw various lines on a whiteboard and then wipes them off

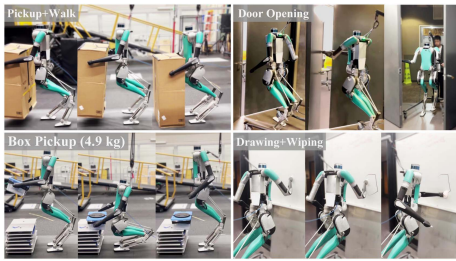


Fig. 6. Snapshots of hardware experiments demonstrating the *pickup+walk*, *door opening*, *box pickup* and *drawing+wiping* tasks.

using its left hand, following a sequence of end-effector subgoals constrained to a planar surface.

These results illustrate that our framework maintains compliant contact, adapts effectively to environmental variations, and enables the robot to perform everyday loco-manipulation behaviors. All policies are trained using over 1000 reference trajectories per task with randomized parameters, supporting the versatility and generalization capability of the Opt2Skill framework across diverse real-world scenarios.

## V. CONCLUSION

In this paper, we present a TO-guided RL pipeline for humanoid loco-manipulation. We show that the RL tracking performance is affected by the quality of the motion reference. The full-body-dynamics-based TO provides high-quality and dynamically-feasible trajectories. Based on such trajectories, motion imitation yields better tracking performance, especially through the use of torque information. We demonstrate our sim-to-real results on the humanoid robot Digit with versatile loco-manipulation skills, including dynamic stair traversing, multi-contact box manipulation, and door traversing.

## REFERENCES

- [1] Z. Gu et al., "Humanoid locomotion and manipulation: Current progress and challenges in control, planning, and learning," *IEEE/ASME Trans. Mechatron.*, 2025, doi: [10.1109/TMECH.2025.3579247](https://doi.org/10.1109/TMECH.2025.3579247).
- [2] X. B. Peng, P. Abbeel, S. Levine, and M. Van de Panne, "DeepMimic: Example-guided deep reinforcement learning of physics-based character skills," *ACM Trans. Graph.*, vol. 37, no. 4, pp. 1–14, 2018.
- [3] X. Cheng, Y. Ji, J. Chen, R. Yang, G. Yang, and X. Wang, "Expressive whole-body control for humanoid robots," in *Robot. Sci. Syst.*, 2024.
- [4] T. He et al., "Learning human-to-humanoid real-time whole-body teleoperation," in *Proc. 2024 IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, 2024, pp. 8944–8951.
- [5] F. Jenelten, J. He, F. Farshidian, and M. Hutter, "DTC: Deep tracking control," *Sci. Robot.*, vol. 9, no. 86, 2024, Art. no. eadh5401.
- [6] D. Mayne, "A second-order gradient method for determining optimal trajectories of non-linear discrete-time systems," *Int. J. Control.*, vol. 3, no. 1, pp. 85–95, 1966.
- [7] Y. Tassa, N. Mansard, and E. Todorov, "Control-limited differential dynamic programming," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2014, pp. 1168–1175.
- [8] P. M. Wensing, M. Posa, Y. Hu, A. Escande, N. Mansard, and A. Del Prete, "Optimization-based control for dynamic legged robots," *IEEE Trans. Robot.*, vol. 40, pp. 43–63, 2023.
- [9] S. Kajita et al., "Biped walking pattern generation by using preview control of zero-moment point," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2003, pp. 1620–1626.
- [10] Z. Gu et al., "Robust-locomotion-by-logic: Perturbation-resilient bipedal locomotion via signal temporal logic guided model predictive control," *IEEE Trans. Robot.*, vol. 41, pp. 4300–4321, 2025.
- [11] D. Kang, J. Cheng, M. Zamora, F. Zargarbashi, and S. Coros, "RL + model-based control: Using on-demand optimal control to learn versatile legged locomotion," *IEEE Robot. Automat. Lett.*, vol. 8, no. 10, pp. 6619–6626, Oct. 2023.
- [12] J. Koenemann et al., "Whole-body model-predictive control applied to the HRP-2 humanoid," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2015, pp. 3346–3351.
- [13] 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 Robot. Automat. Lett.*, vol. 9, no. 12, pp. 11074–11081, Dec. 2024.
- [14] J.-P. Sleiman, F. Farshidian, M. V. Minniti, and M. Hutter, "A unified MPC framework for whole-body dynamic locomotion and manipulation," *IEEE Robot. Automat. Lett.*, vol. 6, no. 3, pp. 4688–4695, Jul. 2021.
- [15] A. Meduri, P. Shah, J. Viereck, M. Khadiv, I. Havoutis, and L. Righetti, "BiConMP: A nonlinear model predictive control framework for whole body motion planning," *IEEE Trans. Robot.*, vol. 39, no. 2, pp. 905–922, Apr. 2023.
- [16] Z. Zhuang, S. Yao, and H. Zhao, "Humanoid parkour learning," in *Proc. Conf. Robot. Learn.*, 2024.
- [17] J. Dao, H. Duan, and A. Fern, "Sim-to-real learning for humanoid box loco-manipulation," in *Proc. Int. Conf. Robot. Automat.*, 2023, pp. 16930–16936.
- [18] C. Sferazza, D.-M. Huang, X. Lin, Y. Lee, and P. Abbeel, "Humanoid-Bench: Simulated humanoid benchmark for whole-body locomotion and manipulation," 2024, [arXiv:2403.10506](https://arxiv.org/abs/2403.10506).
- [19] H. J. Lee, S. Hong, and S. Kim, "Integrating model-based footstep planning with model-free reinforcement learning for dynamic legged locomotion," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2024, pp. 11248–11255.
- [20] C. Li, M. Vlastelica, S. Blaes, J. Frey, F. Grimmering, and G. Martius, "Learning agile skills via adversarial imitation of rough partial demonstrations," in *Proc. Conf. Robot Learn.*, 2023, pp. 342–352.
- [21] L. Krishna, N. Sobanbabu, and Q. Nguyen, "OGMP: Oracle guided multi-mode policies for agile and versatile robot control," 2024, [arXiv:2403.04205](https://arxiv.org/abs/2403.04205).
- [22] P. Dugar, A. Shrestha, F. Yu, B. van Marum, and A. Fern, "Learning multi-modal whole-body control for real-world humanoid robots," 2024, [arXiv:2408.07295](https://arxiv.org/abs/2408.07295).
- [23] Z. Fu, Q. Zhao, Q. Wu, G. Wetzstein, and C. Finn, "HumanPlus: Humanoid shadowing and imitation from humans," in *Proc. Conf. Robot. Learn.*, 2024.
- [24] A. Miller, S. Fahmi, M. Chignoli, and S. Kim, "Reinforcement learning for legged robots: Motion imitation from model-based optimal control," 2023 [arXiv:2305.10989](https://arxiv.org/abs/2305.10989).
- [25] Y. Fuchioka, Z. Xie, and M. Van de Panne, "OPT-mimic: Imitation of optimized trajectories for dynamic quadruped behaviors," in *Proc. Int. Conf. Robot. Automat.*, 2023, pp. 5092–5098.
- [26] R. Batke et al., "Optimizing bipedal maneuvers of single rigid-body models for reinforcement learning," in *Proc. IEEE-RAS 21st Int. Conf. Humanoid Robots*, 2022, pp. 714–721.
- [27] T. Kwon, Y. Lee, and M. Van De Panne, "Fast and flexible multilegged locomotion using learned centroidal dynamics," *ACM Trans. Graph.*, vol. 39, no. 4, 2020, Art. no. 46.
- [28] D. Marew, N. Perera, S. Yu, S. Roelker, and D. Kim, "A biomechanics-inspired approach to soccer kicking for humanoid robots," in *Proc. IEEE-RAS 23rd Int. Conf. Humanoid Robots*, 2024, pp. 722–729.
- [29] S. Levine and V. Koltun, "Guided policy search," in *Proc. Int. Conf. Mach. Learn.*, 2013, pp. 1–9.
- [30] E. Chaikovskaya et al., "Benchmarking the full-order model optimization based imitation in the humanoid robot reinforcement learning walk," in *Proc. Int. Conf. Adv. Robot.*, 2023, pp. 206–211.
- [31] L. Krishna, G. A. Castillo, U. A. Mishra, A. Hereid, and S. Kolathaya, "Linear policies are sufficient to realize robust bipedal walking on challenging terrains," *IEEE Robot. Automat. Lett.*, vol. 7, no. 2, pp. 2047–2054, Apr. 2022.
- [32] C. Mastalli et al., "Crocodyl: An efficient and versatile framework for multi-contact optimal control," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2020, pp. 2536–2542.
- [33] K. Zakka et al., "Mujoco playground," 2025, [arXiv:2502.08844](https://arxiv.org/abs/2502.08844).
- [34] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," 2017, [arXiv:1707.06347](https://arxiv.org/abs/1707.06347).
- [35] L. Pinto, M. Andrychowicz, P. Welinder, W. Zaremba, and P. Abbeel, "Asymmetric actor critic for image-based robot learning," *Robot. Sci. Syst.*, 2018.
- [36] I. Radosavovic, T. Xiao, B. Zhang, T. Darrell, J. Malik, and K. Sreenath, "Real-world humanoid locomotion with reinforcement learning," *Sci. Robot.*, vol. 9, no. 89, 2024, Art. no. eadi9579.
- [37] N. Mahmood, N. Ghorbani, N. F. Troje, G. Pons-Moll, and M. J. Black, "AMASS: Archive of motion capture as surface shapes," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 5442–5451.