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

Fukang Liu, Zhaoyuan Gu, Yilin Cai, Ziyi Zhou, Shijie Zhao, Hyunyoung Jung,
Sehoon Ha, Yue Chen, Danfei Xu†, and Ye Zhao†

***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 precise and systematic control but are limited by high computational complexity and accurate contact sensing. On the other hand, reinforcement learning (RL) provides robustness and handles high-dimensional spaces 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. We generate reference motions for the Digit humanoid robot using differential dynamic programming (DDP) and train RL policies to track these trajectories. Our results demonstrate that Opt2Skill outperforms pure RL methods in both training efficiency and task performance, with optimal trajectories that account for torque limits enhancing trajectory tracking. We successfully transfer our approach to real-world applications.** https://opt2skill.github.io

I. Introduction

Humanoid robots possess inherent advantages in achieving human-like behaviors. Their similar morphology to humans enables them to achieve a wide range of physical locomotion and manipulation tasks, such as handling bulky objects, climbing stairs, and performing agile skills such as jumping. Although their morphology is well suited for real-world human-centered environments, humanoid robots face significant challenges due to their high-dimensional, underactuated nature and the complexities of contact-rich interactions. In particular, leveraging the dynamically-feasible whole-body motion of humanoid robots to achieve diverse loco-manipulation tasks remains an open problem.

Model-based optimal control has been crucial in advancing the capabilities of dynamic legged robots, enabling reliable performance in locomotion tasks [1]. Model Predictive Control (MPC) is widely used for robust online tracking of reference trajectories using various simplified models [2], [3], [4], [5]. To address the limitations of omitting the whole-body model, whole-body MPC has gained attention by incorporating whole-body dynamics [6], [7], [8], [9], [10] or combining whole-body kinematics with centroidal dynamics [11], [12]. Despite its potential, the requirement for high computational power due to high degrees of freedom (DoF) and precise contact state feedback poses challenges for whole-body planning in complex, contact-rich loco-manipulation tasks.

Authors are with the Institute for Robotics and Intelligent Machines, Georgia Institute of Technology, Atlanta, GA, USA.
†Equal advising.

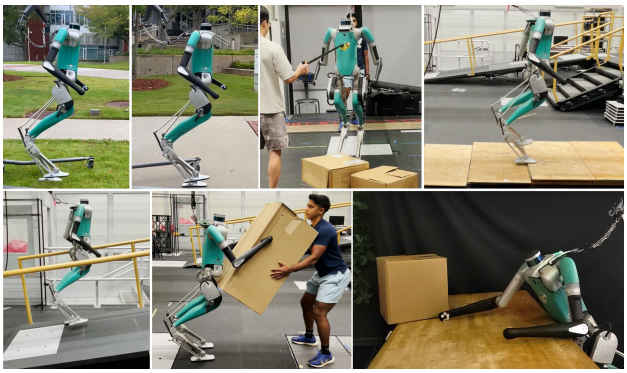

Fig. 1. The proposed Opt2Skill framework enables a Digit humanoid robot to perform various loco-manipulation tasks by mimicking optimal model-based reference trajectories in real-world scenarios.

On the other hand, reinforcement learning (RL) offers a promise for the whole-body control of legged robots, achieving real-time control policies and handling high-dimensional state space [13], [14], [15]. RL methods interact directly with the environment without pre-planned motions, providing a flexible and straightforward training paradigm. However, they often suffer from poor data efficiency. Furthermore, the RL policy learned from scratch can produce unnatural motions [16] and require significant efforts in reward shaping.

To address these challenges in RL without reference, motion imitation has been widely explored to enable humanoid robot loco-manipulation skills. Compared with RL without reference, motion imitation eliminates tedious reward tuning and demonstrates improved data efficiency, yet results in natural motions. Therefore, there is a growing trend in complementing learning-based approaches with motion data for whole-body control of legged robots [17].

Human motion capture has been a primary data source for motion imitation. RL with imitation rewards is one of the most popular approaches to achieving motion imitation while maintaining physical feasibility [18]. Recently, RL-based motion imitation has achieved success in transferring to humanoid hardware [19], [20], [21], enabling versatile teleoperation skills [22], [23]. However, human motion capture data still renders a significant embodiment gap between humans and humanoid robots, which requires motion retargeting to map human data to humanoid data. So far, motion retargeting is mostly limited to upper-body manipulation tasks, and retargeting whole-body loco-manipulation behaviors often produces dynamically infeasible motions that require nontrivial data processing [22]. As such, the current motion retargeting techniques may limit the robot's ability

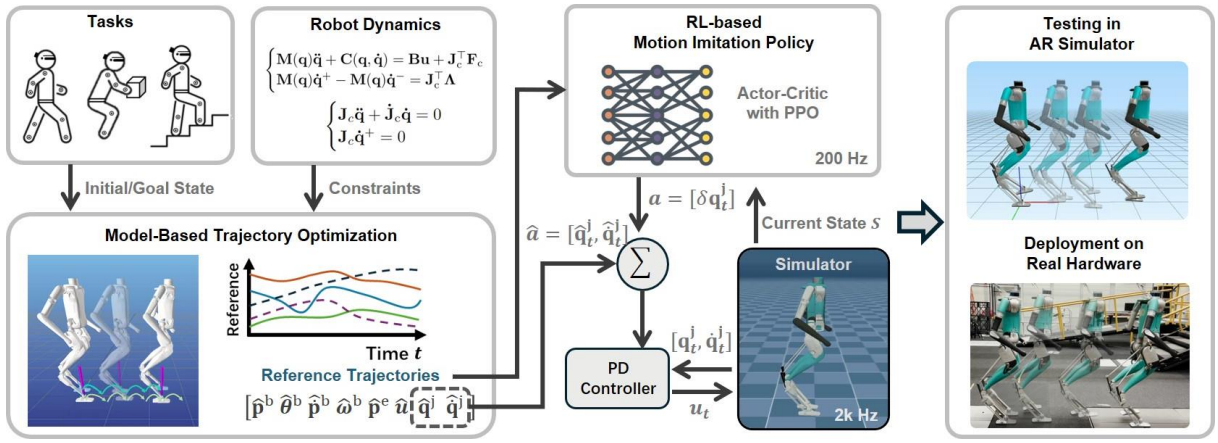

Fig. 2. Overall structure of the Opt2Skill framework. Optimal reference trajectories are generated based on the task goal and robot dynamics model. The RL policy augments the reference trajectories with a residual term, which is applied to a low-level PD controller for torque control. Training includes domain randomization, and the trained policy is able to transfer from the simulation to the real robot.

to imitate the original whole-body motions effectively, and potentially lose behavior versatility.

To address these challenges, recent studies have begun using model-based trajectory optimization (TO) to generate data for motion imitation. Model-based TO produces high-quality motion data by ensuring that the trajectory is dynamically feasible and adheres to physical constraints, such as joint torque limits. TO-based motion imitation has been widely employed on quadruped robots [24], [17], [25]. For humanoid robots, various models have been leveraged for TO-based motion imitation, including single rigid-body model [26], centroidal dynamics model [27], kinodynamic model [28], and full-order-dynamics model [29], [30]. However, most of these studies focus on only locomotion tasks rather than whole-body loco-manipulation ones. For loco-manipulation tasks, the high fidelity of the full-order model is beneficial as it eliminates the requirement for retargeting. Furthermore, full-order motion data includes the joint torque, which is critical for learning high-dimensional contact-rich skills. This work adopts a full-order-dynamics-based TO approach to guide RL, particularly for humanoid loco-manipulation tasks. A closely related study is [25], which also uses torque information from TO for RL motion imitation. However, our approach differs from [25] in several key aspects, including the complexity of the robot dynamics, the method for tracking torque reference, and the extent of hardware validation.

This study presents Opt2Skill, a scalable learning pipeline that transforms TO-based reference motions into RL skills for sim-to-real humanoid loco-manipulation tasks. Our approach leverages differential dynamic programming (DDP) [31], [32] to generate whole-body motions that obey the robot's dynamics and task requirements. We study a rich set of loco-manipulation tasks, including walking, stair traversing, and bulky-object handling, as shown in Fig. 1. We train the Opt2Skill using RL, which enables accurate tracking of these optimal and dynamically feasible motions. Our study indicates that Opt2Skill outperforms pure RL methods learned from scratch in both training efficiency and task performance. Moreover, TO trajectories that account for torque limits enhance trajectory tracking during RL training. We demonstrate the successful transfer of these skills to real-world experiments. The contributions of this study can be summarized as:

- This work represents the first step to adopting full-order-dynamics-based TO to guide RL that achieves humanoid loco-manipulation tasks.
- We demonstrate that the quality of motion data is critical for motion imitation and full-body TO is a high-quality source of motion data. In addition, the joint torque information from the motion data is critical to achieve successful motion tracking. Such information can not be acquired by motion capture data.
- We demonstrate the capability of our framework via successful sim-to-real transfer for a diverse set of humanoid loco-manipulation tasks, including robust locomotion on stairs and in outdoor environments, and multi-contact whole-body manipulation.

II. Methods

Opt2Skill aims to develop loco-manipulation controllers that enable the Digit humanoid robot to track model-based optimal trajectories. Our framework is illustrated in Fig. 2. We start by generating whole-body reference motions that align with the robot's dynamics and meet specific motion targets using DDP through a Crocoddyl solver [33]. These reference trajectories are then used during the training and deployment of our RL policy. The RL policy augments the reference joint trajectories with a residual term, and the augmented target trajectory is fed into a low-level PD controller for the torque-controlled robot. Finally, we deploy the RL policies in both simulation and real-world scenarios.

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 is given by:

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

where $\mathbf{q} = [\mathbf{q}^{\mathrm{b}}; \mathbf{q}^{\mathrm{j}}] \in \mathbb{R}^{n_q}$, $\mathbf{v} = [\mathbf{v}^{\mathrm{b}}; \dot{\mathbf{q}}^{\mathrm{j}}] \in \mathbb{R}^{n_v}$ are the generalized coordinates and velocities partitioned in

base and joint variables. The superscripts b and j are base and joint related quantities, respectively. Base coordinates $\mathbf{q}^{\mathrm{b}} = \left[\mathbf{p}^{\mathrm{b}}; \boldsymbol{\theta}^{\mathrm{b}}\right] \in \mathbb{R}^6$ are partitioned as base position and orientation. Base velocities $\mathbf{v}^{\mathrm{b}} = \left[\dot{\mathbf{p}}^{\mathrm{b}}; \boldsymbol{\omega}^{\mathrm{b}}\right] \in \mathbb{R}^6$ are partitioned as base linear and angular velocities in the world frame. $\mathbf{M}(\mathbf{q})$ is the joint space mass matrix. $\mathbf{C}(\mathbf{q}, \mathbf{v})$ captures the nonlinear effects. $\mathbf{u} \in \mathbb{R}^{n_j}$ denotes joint torques. $\mathbf{J}_{\mathrm{c}}(\mathbf{q})$ is the stacked contact Jacobian and $\mathbf{F}_{\mathrm{c}}$ is the stacked contact reaction force vector. This model will be referred to as the *whole-body dynamics* of the robot.

Given this whole-body dynamics, the TO formulation for generating a whole-body trajectory is written as:

$$\min_{\mathbf{x},\mathbf{u}} \quad \sum_{k=0}^{N-1} \left(\|\mathbf{x}[k] - \mathbf{x}^{\mathrm{des}}[k]\|_Q^2 + \|\mathbf{u}[k]\|_R^2 \right) + \|\mathbf{x}[N] - \mathbf{x}^{\mathrm{des}}[N]\|_{Q_{\mathrm{f}}}^2 \tag{2a}$$

subject to

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

$$\text{(Contact/Impact)} \quad \begin{cases} \mathbf{J}_{\mathrm{c}}\ddot{\mathbf{q}} + \dot{\mathbf{J}}_{\mathrm{c}}\dot{\mathbf{q}} = 0 \\ \mathbf{J}_{\mathrm{c}}\dot{\mathbf{q}}^{+} = 0 \end{cases} \tag{2c}$$

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

$$\text{(Friction)} \quad \mathbf{F}_{\mathrm{c}} \in \mathcal{F}(\mu, \mathbf{q}) \tag{2e}$$

where the decision variables include a full-body state vector $\mathbf{x} = \left[\mathbf{q}; \dot{\mathbf{q}}\right] \in \mathbb{R}^{n_q + n_v}$ and the joint torque $\mathbf{u}$. In Eq. (2b), the hybrid dynamics constraint includes either the continuous whole-body dynamics (first row), or the impact dynamics (second row), where $\boldsymbol{\Lambda}$ stands for the contact impulse, and $\dot{\mathbf{q}}^{-}$ and $\dot{\mathbf{q}}^{+}$ are instantaneous velocities before and after the impact. Following (2b), a contact constraint (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 friction coefficient μ are also modeled in (2d)-(2e).

Considering the computational efficiency, we choose the DDP-based method implemented in Crocoddyl [33] to solve the whole-body optimization. The joint limit, torque limit, and friction cone constraints are encoded as quadratic barrier functions inside the objective function.

B. Motion Imitation

The goal of the RL-based motion imitation is to develop a comprehensive closed-loop control policy that replicates the reference motion generated by the trajectory optimizer. This control policy operates within a realistic physics simulation, with the ultimate aim of being deployed on a physical robot. The RL environment is based on the MuJoCo simulator [34], and the RL policy is an Actor-Critic (AC) trained with Proximal Policy Optimization (PPO) algorithm [35].

Robot State and Observation Space. The robot state $s \in \mathbb{R}^{9+n}$ consists of the linear and angular velocities of the robot base, the projected gravity (*i.e.*, a proxy of base orientation), and the position of robot joints. The number of joints $n = 20$.

$$\mathbf{s} = \left[\ \dot{\mathbf{p}}^{\mathrm{b}}; \quad \boldsymbol{\omega}^{\mathrm{b}}; \quad \mathbf{g}_{\mathrm{proj}}; \quad \mathbf{q}^{\mathrm{j}} \ \right] \tag{3}$$

TABLE I

OBSERVATION AND STATE SPACES.

Input	Symbol	Dim	Actor	Critic
Base Lin. Vel.	$\dot{\mathbf{p}}^{\mathrm{b}}$	3	✓	✓
Base Ang. Vel.	$\boldsymbol{\omega}^{\mathrm{b}}$	3	✓	✓
Projected Gravity	$\mathbf{g}_{\mathrm{proj}}$	3	✓	✓
Motor Joint Pos.	$\mathbf{q}^{\mathrm{j}}$	20	✓	✓
Ref Base Lin. Vel.	$\hat{\dot{\mathbf{p}}}^{\mathrm{b}}$	3	✓	✓
Ref Base Ang. Vel.	$\hat{\boldsymbol{\omega}}^{\mathrm{b}}$	3	✓	✓
Ref Motor Joint Pos.	$\hat{\mathbf{q}}^{\mathrm{j}}$	20	✓	✓
Action	$\mathbf{a}$	20	✓	✓
History Motor Joint Pos.	$\mathbf{q}_{\mathrm{h}}^{\mathrm{j}}$	60	✓	✓
Base Translation	$\mathbf{p}^{\mathrm{b}}$	3		✓
Motor Joint Vel.	$\dot{\mathbf{q}}^{\mathrm{j}}$	20		✓
End-effector Pos.	$\mathbf{p}^{\mathrm{e}}$	12		✓
Ref Base Translation	$\hat{\mathbf{p}}^{\mathrm{b}}$	3		✓
Ref Motor Joint Vel.	$\hat{\dot{\mathbf{q}}}^{\mathrm{j}}$	20		✓
Ref End-effector Pos.	$\hat{\mathbf{p}}^{\mathrm{e}}$	12		✓
PD Gains	$\mathbf{K}_p, \mathbf{K}_d$	40		✓
Motor Joint Friction	$\mathbf{f}^{\mathrm{j}}$	20		✓

The description of each notation is detailed in Table I.

The observation space includes the current state of the robot, the action, and the trajectory reference. To enhance training efficiency, we utilize separate observation spaces for the actor and critic agents. The observation space for the actor agent is defined as:

$$\mathbf{o}_{\mathrm{actor}} = \left[\ \mathbf{s}; \quad \hat{\mathbf{p}}^{\mathrm{b}}; \quad \hat{\boldsymbol{\omega}}^{\mathrm{b}}; \quad \hat{\mathbf{q}}^{\mathrm{j}}; \quad \mathbf{a}; \quad \mathbf{q}_{\mathrm{h}}^{\mathrm{j}} \ \right] \tag{4}$$

where $\hat{}$ indicates the trajectory reference. $\mathbf{a}$ is the action space defined in the following section. $\mathbf{q}_{\mathrm{h}}^{\mathrm{j}}$ is a history of the motor joint positions (specifically, three previous training steps).

We also observed that different loco-manipulation tasks require distinct PD gains for optimal tracking controller performance. To ensure that the critic agent gains a comprehensive understanding of the control dynamics and provides effective guidance for the actor agent, we also incorporate the specific $\mathbf{K}_p$ and $\mathbf{K}_d$ gains into the observation space of the critic agent.

$$\mathbf{o}_{\mathrm{critic}} = \left[\mathbf{o}_{\mathrm{actor}}; \ \mathbf{p}^{\mathrm{b}}; \ \dot{\mathbf{q}}^{\mathrm{j}}; \ \mathbf{p}^{\mathrm{e}}; \ \hat{\mathbf{p}}^{\mathrm{b}}; \ \hat{\dot{\mathbf{q}}}^{\mathrm{j}}; \ \hat{\mathbf{p}}^{\mathrm{e}}; \ \mathbf{K}_p; \ \mathbf{K}_d; \ \mathbf{f}^{\mathrm{j}} \ \right]$$

Action Space. The action space of the control policy is defined as $\mathbf{a} = \delta\mathbf{q}^{\mathrm{j}} \in \mathbb{R}^{20}$, which are the residual joint positions. These residuals are added to the reference joint positions and velocities to produce an augmented target, which is then fed into a proportional-derivative (PD) joint torque controller.

$$\mathbf{u}_t = \mathbf{K}_p \left(\hat{\mathbf{q}}_t^{\mathrm{j}} + \delta\mathbf{q}_t^{\mathrm{j}} - \mathbf{q}_t^{\mathrm{j}} \right) + \mathbf{K}_d \left(\hat{\dot{\mathbf{q}}}_t^{\mathrm{j}} - \dot{\mathbf{q}}_t^{\mathrm{j}} \right) \tag{5}$$

where $\mathbf{u}_t$ is the torque command sent to the joint motors at time step t, $\mathbf{K}_p$ and $\mathbf{K}_d$ denote the gains for the PD controller, $\hat{\mathbf{q}}^{\mathrm{j}}$ and $\hat{\dot{\mathbf{q}}}^{\mathrm{j}}$ denote the joint positions and velocities output from the trajectory optimization, $\mathbf{q}^{\mathrm{j}}$ and $\dot{\mathbf{q}}^{\mathrm{j}}$ are the actual joint positions and velocities on the robot.

Reward. Inspired by the works in [18] and [36], we design the reward functions r_t at time t to encourage the robot to follow the reference motion. The imitation rewards are detailed in Table II, which outlines our reward design for loco-manipulation tasks. These rewards are fine-tuned according to the specifics of each task. For example, base motion and

TABLE II
REWARD COMPONENTS AND WEIGHTS.

Term	Expression	Weight
Task Reward		
Joint Pos.	$\exp(-5\|\hat{\mathbf{q}}_t^{\mathrm{j}} - \mathbf{q}_t^{\mathrm{j}}\|_2^2)$	0.30
Joint Vel.	$\exp(-0.1\|\hat{\dot{\mathbf{q}}}_t^{\mathrm{j}} - \dot{\mathbf{q}}_t^{\mathrm{j}}\|_2^2)$	0.05
Base Pos.	$\exp(-20\|\hat{\mathbf{p}}_t^{\mathrm{b}} - \mathbf{p}_t^{\mathrm{b}}\|_2^2)$	0.15
Base Ori.	$\exp(-50\|\hat{\boldsymbol{\theta}}_t^{\mathrm{b}} - \boldsymbol{\theta}_t^{\mathrm{b}}\|_2^2)$	0.20
Base Lin. Vel.	$\exp(-2\|\hat{\dot{\mathbf{p}}}_t^{\mathrm{b}} - \dot{\mathbf{p}}_t^{\mathrm{b}}\|_2^2)$	0.15
Base Ang. Vel.	$\exp(-0.5\|\hat{\boldsymbol{\omega}}_t^{\mathrm{b}} - \boldsymbol{\omega}_t^{\mathrm{b}}\|_2^2)$	0.15
End-effector Pos.	$\exp(-20\|\hat{\mathbf{p}}^{\mathrm{e}} - \mathbf{p}^{\mathrm{e}}\|_2^2)$	0.15
Joint Torque	$\exp(-10^{-3}\|\hat{\mathbf{u}}_t - \mathbf{u}_t\|_2^2)$	0.20
Penalty Cost [37]		
Base Motion	$\|\dot{\mathbf{p}}_t^{\mathrm{b,z}}\|_2^2 + 0.5 \times \|\boldsymbol{\omega}_t^{\mathrm{b,xy}}\|_2^2$	-1
Base Ori.	$\|\boldsymbol{\theta}_t^{\mathrm{b,xy}}\|_2^2$	-2
Action Rate	$\|\mathbf{a}_t - \mathbf{a}_{t-1}\|_2^2$	-0.05
Joint Smoothing	$\|\mathbf{a}_t - 2\mathbf{a}_{t-1} + \mathbf{a}_{t-2}\|_2^2$	-0.05
Torques	$\|\mathbf{u}_t / \mathbf{u}_{\text{limit}}\|_2^2$	-0.025
Joint Acc.	$\|\ddot{\mathbf{q}}_t^{\mathrm{j}}\|_2^2$	-5×10^{-7}

orientation penalties are excluded in tasks involving upper body leaning or with non-zero z-velocity, such as hand-pushing objects on a desk or jumping. Additionally, for tasks like box pickup, we include specific rewards for the pickup action, which are detailed in Sec. III-A.

Domain Randomization. To facilitate transfer from simulation to the real robot, we employ dynamics randomization throughout the training process. We randomize the observations, motor parameters, and physical parameters of the simulated environments. Detailed parameters and noise setting are inspired from [37] and listed in Table III. The randomization ranges are selected to ensure diverse yet realistic training scenarios, aiding in the development of a policy that generalizes to the dynamics of the real robot without extensive real-world fine-tuning.

Initialization and Termination. To obtain kinematically and dynamically feasible initial states, we first implement a standing controller that maintains balance from various arbitrary standing poses. We record a set of feasible poses where the standing controller can keep balance and randomly select poses from this set as the initial state of each episode. The episode is terminated when: (1) the episode length exceeds the maximum limit; (2) the robot experiences self-collision; (3) the root velocities exceed specified thresholds; or (4) the body leaning angle surpasses predefined thresholds.

Implementation Details. The actor and critic networks are defined as Multi-Layer Perceptron (MLP) networks ReLU activations and hidden dimensions of 256×256. We train RL policies in MuJoCo simulator with 40 parallel environments, which takes around 7 hours on a desktop PC with Intel i7-13700K and NVIDIA RTX4080 for each algorithm. The policy runs at 200 Hz and generates actions corresponding to PD setpoint targets for each controlled joint. These actions are then forwarded to a PD controller running at 2 kHz with constant gains. For simulation experiments, we use a high-fidelity physics-based simulator developed by Agility Robotics [38]. The verified policies are further deployed on the Digit hardware in a zero-shot fashion.

III. RESULTS

In this section, we evaluate the performance of Opt2Skill on various loco-manipulation tasks both in simulation and on hardware.

TABLE III
DOMAIN RANDOMIZATION PARAMETERS.

Category	Parameter and Details
Observation	Joint Pos. (Additive): Gaussian (0, 0.0875) Joint Vel. (Additive): Gaussian (0, 0.075) Base Lin. Vel. (Additive): Gaussian (0, 0.15) Base Ang. Vel. (Additive): Gaussian (0, 0.15) Gravity Projection (Additive): Gaussian (0, 0.075)
Delays	Action Delay: Uniform (0, 0.2) × dt
Motor	Motor Strength (Scaling): Uniform (0.85, 1.15) Motor Damping (Scaling): Loguniform (0.3, 4) Mass (Scaling): Uniform (0.5, 1.5) Kp/Kd (Scaling): Uniform (0.9, 1.1)
Environment	Gravity (Additive): Uniform (0, 0.67) Friction (Scaling): Uniform (0.2, 2) Terrain: flat and rough

A. Simulation Experiments

*1) **Implementation and Baseline Setup**:* We train our Opt2Skill algorithms as a more generalized policy by imitating over 100 different robot trajectories for each task, generated by using DDP. We compare Opt2Skill with a pure RL method that learns from scratch in three different tasks. The pure RL baseline excludes reference trajectories in the observation space and the reward design. It uses only joint positions as the action space for a P-controller to track.

- **Forward Walking** The goal of this task is to achieve stable dynamic locomotion, on various terrains such as flat ground, stairs, and ramp.
  - *For Opt2Skill*, the reward structure is detailed in Table II. The reference motion data consists of a library of trajectories with varying step heights (ranging from $0.1 \sim 0.14$ m) and linear velocities (from $0 \sim 0.5$ m/s). The episode length of each trajectory is 1944 timesteps (approximately 10 seconds).
  - *For pure RL*, we use the base velocity tracking reward, foot height reward, and foot contact reward following [39]. The foot height reward encourages the swing foot to lift up periodically, and the foot contact reward encourages the standing foot to maintain contact. We apply the same penalty functions with one additional foot distance penalty to prevent two feet from being too close together or too far away.
- **Box pickup** The goal is to lift up a box on the ground.
  - *For Opt2Skill*, we design heuristic target goals for the robot's hand trajectory, accounting for the transition between contact phases with and without holding the box by adding or removing its mass and inertia from the hands similar to [40]. Opt2Skill also includes a box position tracking reward in addition to the rewards in Table II. Note that we also exclude penalties for base motion and orientation.
  - *For pure RL*, we use the same target goals for the robot hand and box position as Opt2Skill. The two sequential target goals are positions where the robot starts picking from and then lifts up to. We reward the policy as the current position gets closer to the two

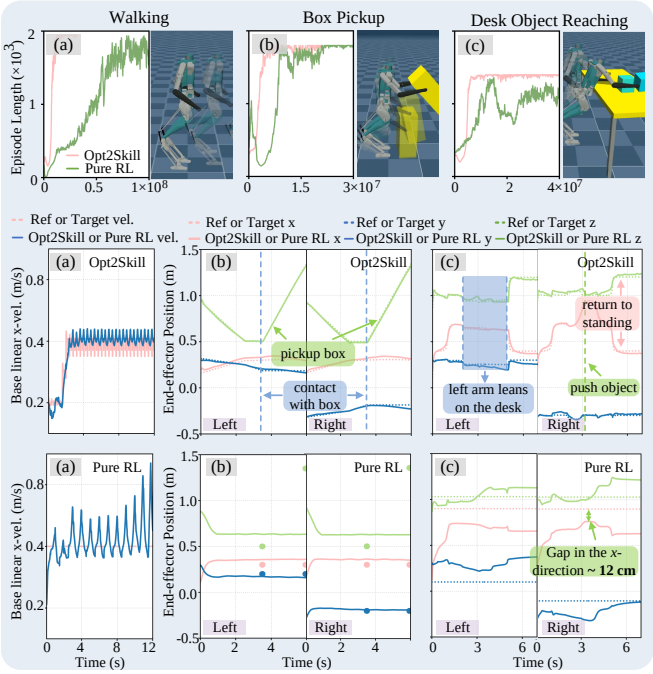

Fig. 3. Performance comparison of Opt2Skill and pure RL across three tasks: (a) forward walking, (b) box pickup, and (c) desk object reaching. The top row shows episode lengths over training timesteps, demonstrating faster convergence of Opt2Skill due to model-based optimal reference trajectories. The middle row illustrates Opt2Skill's precise tracking of reference motions in each task: velocity tracking in the *Walking* task, and end-effector tracking in the *Box Pickup* and *Desk Object Reaching* tasks, which involves dynamic interactions in rich-contact scenarios. The bottom row shows that the pure RL method exhibits larger velocity tracking deviations, fails to complete the box pickup task as the robot cannot lift the box and stand up, and succeeds in pushing the object but fails to reach the target position or return to a standing position. Note that, in (b) and (c), end-effector positions are global.

target goal positions, and we use the same penalty term as Opt2Skill.

- **Desk object reaching** The goal of this multi-contact task is to lean on a desk using the left arm to support the body while using the other arm to reach and push an object on the desk. Once the pushing task is completed, the robot returns to a balanced standing position.
  - *For Opt2Skill*, we define a sequence of discrete contact positions between the robot and the environment and use TO to generate a reference trajectory that finishes the task. We also exclude penalties for base motion and orientation.
  - *For pure RL*, we use the same contact positions as the target goals for the robot's root position. The target goal for the robot's hand is set relative to the robot's root position. Similarly, We reward the policy when the robot's current position is close to the target positions, and the same penalty terms are used as in Opt2Sill.

*2) **Performance Analysis**:* Curves showing the returns over episode lengths during training steps are presented in Fig. 3. The results indicate that Opt2Skill converges significantly faster than the pure RL method across all three tasks, thanks to the guidance from reference trajectories.

For the *Forward Walking* task, we test the policy with a new reference trajectory (12 seconds long). Opt2Skill demonstrates tracking of the reference velocities in the x-axis with an average error < 0.05 m/s. For pure RL, the robot is trained to track a target velocity of 0.4 m/s. However, the tracking error could reach up to 0.4 m/s.

For *Box Pickup*, two key factors are crucial: 1) hand position and 2) contact force. A successful pickup requires the robot's hands to precisely reach the box and apply sufficient force to lift it. In Opt2Skill, trajectory optimization (TO) considers both hand positions and contact forces, providing high-quality reference trajectories. Fig. 3 (b) demonstrates that Opt2Skill tracks the hand positions accurately. For pure RL in this task, the robot fails to pick the box up. This is because, without a reference trajectory to guide its motion, the robot's arms collide with the box's top surface, and the pure RL policy fails to find a way to reach the target picking position.

Similarly, for the *Desk Object Reaching* task, Opt2Skill effectively tracks the reference trajectory, benefiting from the dynamics modeling of the robot and environment in the TO. If the trajectory is not dynamically feasible, even manually providing a trajectory or teleoperating the robot to complete the task, the robot may fall due to insufficient torque in the arm needed to support the robot, resulting from an imprecise dynamic configuration pose. For pure RL in this task, the robot successfully reaches the box but leaves a gap of ~ 12 cm from the final target position in the x-direction, failing to push the object to the target goal, as shown in Fig. 3 (c). Additionally, the robot is unable to return to a standing position.

*3) **Ablation Study**:* A notable challenge in motion imitation using human pose data is that the re-targeted motion may not ensure kinematic or dynamic feasibility, which can degrade tracking performance and even cause tracking failures. To demonstrate how dynamics affect motion data quality, we define a *Jumping* task. The robot should track the feet's vertical height (10 cm along the z-axis) and horizontal distance (30 cm along the x-axis) to reach a target position. We generate reference trajectories for *Jumping* task both with and without torque limits from the TO and analyze their difference in Opt2Skill's motion imitation performance. Furthermore, we analyze the effect of the torque reference on tracking performance.

The training curves in Fig. 4 (a) indicate no significant difference in the tracking episode returns for reference trajectories with and without torque limits. However, notable differences are observed in the tracking performance for the base and foot positions in the z direction. For the reference trajectory generated without torque limits, the robot tracks the motion well during the standing phase. However, during the take-off phase, there is a substantial discrepancy between the actual motion and the reference (or desired) motion, with the z discrepancy getting up to 7 cm on the base (see Fig. 4 (b)) and 4 cm on the foot (see Fig. 4 (c)). In contrast, for the reference trajectory generated with proper torque limits, these differences are reduced to 3 cm on the base (see Fig. 4 (d)) and 2.5 cm on the foot (see Fig. 4 (e)).

Moreover, incorporating the torque reference into reward function design further enhances motion tracking performance, as shown in Fig. 4 (d) and (e). When a torque tracking reward, as detailed in Table II, is applied, the robot can track

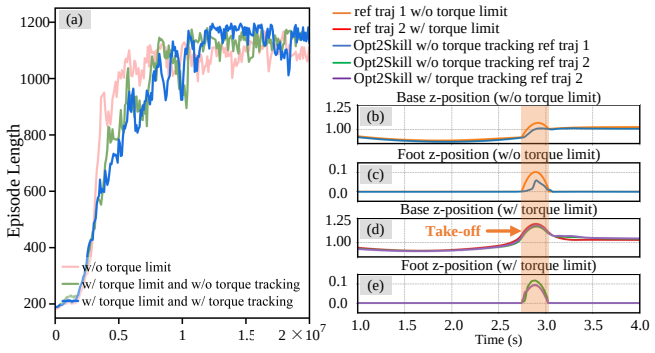

Fig. 4. Impact of torque limit and torque tracking on Opt2Skill's performance in the "jumping" task. (a) Training curves show episode length over timesteps for three setups. (b) and (c) show base and foot z-position tracking without torque limits, highlighting tracking discrepancies during the take-off phase. (d) and (e) display improved tracking of base and foot z-positions with torque limits and torque tracking applied.

the foot z position with minimal error, and the gap in the base z position tracking is reduced to < 1.5 cm during the take-off phase. Therefore, both including the torque limit in the reference trajectory and incorporating the torque reference into reward design significantly contribute to motion tracking performance. The torque limit ensures a high-quality and dynamically feasible reference trajectory, while the torque reference guides the robot to track the motion more precisely.

B. Hardware Experiments

We demonstrate the sim-to-real performance of our Opt2Skill through five hardware experiments. The snapshots of hardware tasks and their corresponding data plots are shown in Fig. 5. For each task, we compare the tracking performance between the robot's measured state and the reference trajectory. The robot states are measured from the robot's onboard sensor and further estimated from a built-in state estimator from Agility Robotics [38].

For locomotion tasks, the plot in Fig. 5 (a) demonstrates accurate tracking of base position in the x, y directions when walking on flat ground, and the plots in Fig. 5 (b) and (c) show the foot height as the robot successfully walks up a stair and a ramp. Notably, our Opt2Skill obtains rough terrain walking capability even though the reference trajectory is only for flat ground scenarios. We also observe the robot keeps its toe compliant with the terrain when stepping on both stair edges and ramp slopes. This is due to the intentional use of small PD gains in the low-level controller, which also reduces the impact noise during walking.

Fig. 5 (d) demonstrates a multi-contact whole-body manipulation task in which the robot reaches for an object on a desk far beyond its supporting polygon. To achieve this, the robot has to lean on the table, using its elbow contact to support a proper upper body pose, while the other arm reaches out to push a box. The demonstration effectively shows that our Opt2Skill framework is capable of high-dimensional loco-manipulation tasks. In Fig. 5 (e), the Digit robot squats down to pick up a bulky box from the ground and then hands it over to a human. The plots in Fig. 5 (d) and (e) display the tracking performance of both end-effector positions during these complex motion sequences. Notably, the box pick-up

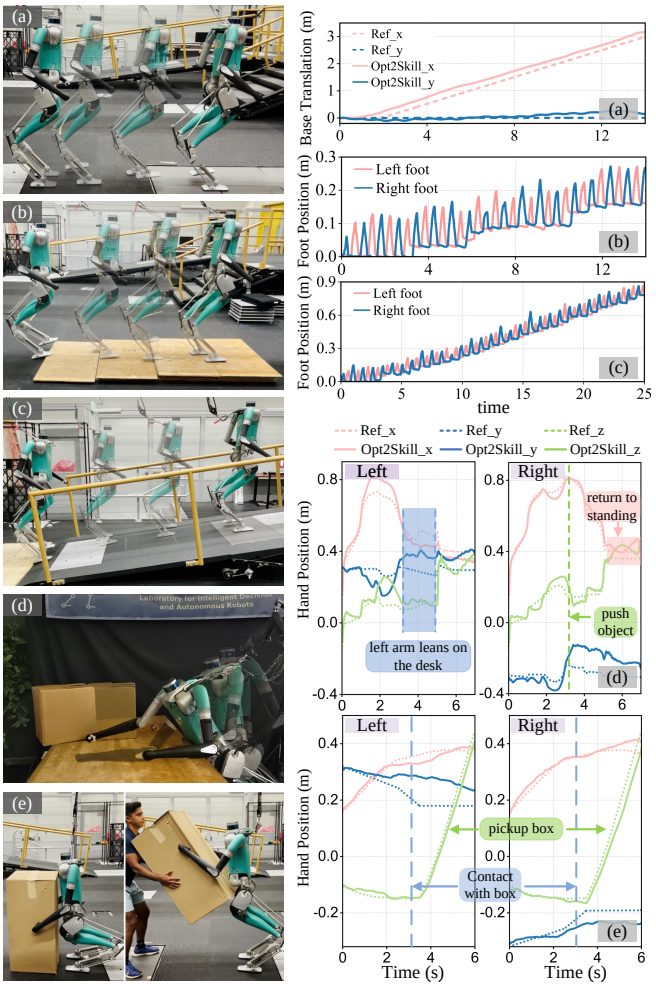

Fig. 5. The snapshots of hardware experiments with data plots. (a) Flat ground locomotion with accurate base position tracking. (b-c) Walking up a stair and a ramp (around 19.5 degrees) with foot z position indicating the height elevation. (d) Desk object reaching with plots of end-effector tracking. (e) The box pick-up and handover process have end-effector tracking deviation when both hands exert contact force to lift the box up. Note that in (d) and (e), end-effector positions are relative to the root.

task exhibits a significant y-axis tracking deviation when the robot makes contact with the box, as indicated by the red arrow in Fig. 5 (e). This deviation is induced by the size and weight mismatch between the simulation and the real world. Despite this mismatch, our framework is able to maintain a compliant contact and adapt to environmental variations.

IV. Conclusion

In this paper, we present a TO-guided RL pipeline for humanoid loco-manipulation. We show 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 walking, stair traversing, and multi-contact box manipulation.

REFERENCES

[1] P. M. Wensing, M. Posa, Y. Hu, A. Escande, N. Mansard, and A. Del Prete, "Optimization-based control for dynamic legged robots," *IEEE Transactions on Robotics*, 2023.

[2] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, and H. Hirukawa, "Biped walking pattern generation by using preview control of zero-moment point," in *2003 IEEE international conference on robotics and automation (Cat. No. 03CH37422)*, vol. 2. IEEE, 2003, pp. 1620–1626.

[3] J. Di Carlo, P. M. Wensing, B. Katz, G. Bledt, and S. Kim, "Dynamic locomotion in the mit cheetah 3 through convex model-predictive control," in *2018 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 2018, pp. 1–9.

[4] Z. Zhou, B. Wingo, N. Boyd, S. Hutchinson, and Y. Zhao, "Momentum-aware trajectory optimization and control for agile quadrupedal locomotion," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 7755–7762, 2022.

[5] Z. Gu, Y. Zhao, Y. Chen, R. Guo, J. K. Leestma, G. S. Sawicki, and Y. Zhao, "Robust-locomotion-by-logic: Perturbation-resilient bipedal locomotion via signal temporal logic guided model predictive control," 2024.

[6] J. Koenemann, A. Del Prete, Y. Tassa, E. Todorov, O. Stasse, M. Bennewitz, and N. Mansard, "Whole-body model-predictive control applied to the hrp-2 humanoid," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2015, pp. 3346–3351.

[7] M. Neunert, M. Stäuble, M. Giftthaler, C. D. Bellicoso, J. Carius, C. Gehring, M. Hutter, and J. Buchli, "Whole-body nonlinear model predictive control through contacts for quadrupeds," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1458–1465, 2018.

[8] S. Katayama and T. Ohtsuka, "Whole-body model predictive control with rigid contacts via online switching time optimization," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 8858–8865.

[9] C. Mastalli, W. Merkt, G. Xin, J. Shim, M. Mistry, I. Havoutis, and S. Vijayakumar, "Agile maneuvers in legged robots: a predictive control approach," *arXiv preprint arXiv:2203.07554*, 2022.

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

[11] J.-P. Sleiman, F. Farshidian, M. V. Minniti, and M. Hutter, "A unified mpc framework for whole-body dynamic locomotion and manipulation," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 4688–4695, 2021.

[12] 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 Transactions on Robotics*, vol. 39, no. 2, pp. 905–922, 2023.

[13] H. Z. Ziwen Zhuang, Shenzhe yao, "Humanoid parkour learning," 2024.

[14] J. Dao, H. Duan, and A. Fern, "Sim-to-real learning for humanoid box loco-manipulation," in *International Conference on Robotics and Automation*, 2023.

[15] Z. Xie, J. Tseng, S. Starke, M. van de Panne, and C. K. Liu, "Hierarchical planning and control for box loco-manipulation," *Proc. ACM Comput. Graph. Interact. Tech.*, vol. 6, no. 3, aug 2023. [Online]. Available: https://doi.org/10.1145/3606931

[16] C. Sferrazza, D.-M. Huang, X. Lin, Y. Lee, and P. Abbeel, "Humanoidbench: Simulated humanoid benchmark for whole-body locomotion and manipulation," *arXiv preprint arXiv:2403.10506*, 2024.

[17] F. Jenelten, J. He, F. Farshidian, and M. Hutter, "Dtc: Deep tracking control," *Science Robotics*, vol. 9, no. 86, p. eadh5401, 2024.

[18] 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*, vol. 37, no. 4, jul 2018. [Online]. Available: https://doi.org/10.1145/3197517.3201311

[19] X. Cheng, Y. Ji, J. Chen, R. Yang, G. Yang, and X. Wang, "Expressive whole-body control for humanoid robots," *arXiv preprint arXiv:2402.16796*, 2024.

[20] P. Dugar, A. Shrestha, F. Yu, B. van Marum, and A. Fern, "Learning multi-modal whole-body control for real-world humanoid robots," 2024. [Online]. Available: https://arxiv.org/abs/2408.07295

[21] I. Radosavovic, B. Zhang, B. Shi, J. Rajasegaran, S. Kamat, T. Darrell, K. Sreenath, and J. Malik, "Humanoid locomotion as next token prediction," *arXiv preprint arXiv:2402.19469*, 2024.

[22] T. He, Z. Luo, W. Xiao, C. Zhang, K. Kitani, C. Liu, and G. Shi, "Learning human-to-humanoid real-time whole-body teleoperation," *arXiv preprint arXiv:2403.04436*, 2024.

[23] Z. Fu, Q. Zhao, Q. Wu, G. Wetzstein, and C. Finn, "Humanplus: Humanoid shadowing and imitation from humans," 2024. [Online]. Available: https://arxiv.org/abs/2406.10454

[24] A. Miller, S. Fahmi, M. Chignoli, and S. Kim, "Reinforcement learning for legged robots: Motion imitation from model-based optimal control," 2023.

[25] Y. Fuchioka, Z. Xie, and M. Van de Panne, "Opt-mimic: Imitation of optimized trajectories for dynamic quadruped behaviors," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 5092–5098.

[26] R. Batke, F. Yu, J. Dao, J. Hurst, R. L. Hatton, A. Fern, and K. Green, "Optimizing bipedal maneuvers of single rigid-body models for reinforcement learning," in *IEEE-RAS 21st International Conference on 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 Transactions On Graphics*, vol. 39, no. 4, aug 2020. [Online]. Available: https://doi.org/10.1145/3386569.3392432

[28] D. Marew, N. Perera, S. Yu, S. Roelker, and D. Kim, "A biomechanics-inspired approach to soccer kicking for humanoid robots," 2024. [Online]. Available: https://arxiv.org/abs/2407.14612

[29] S. Levine and V. Koltun, "Guided policy search," in *Proceedings of the 30th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, S. Dasgupta and D. McAllester, Eds., vol. 28, no. 3. Atlanta, Georgia, USA: PMLR, 17–19 Jun 2013, pp. 1–9. [Online]. Available: https://proceedings.mlr.press/v28/levine13.html

[30] E. Chaikovskaya, I. Minashina, V. Litvinenko, E. Davydenko, D. Makarov, Y. Danik, and R. Gorbachev, "Benchmarking the full-order model optimization based imitation in the humanoid robot reinforcement learning walk," in *2023 21st International Conference on Advanced Robotics (ICAR)*, 2023, pp. 206–211.

[31] D. Mayne, "A second-order gradient method for determining optimal trajectories of non-linear discrete-time systems," *International Journal of Control*, vol. 3, no. 1, pp. 85–95, 1966.

[32] Y. Tassa, N. Mansard, and E. Todorov, "Control-limited differential dynamic programming," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, 2014, pp. 1168–1175.

[33] C. Mastalli, R. Budhiraja, W. Merkt, G. Saurel, B. Hammoud, M. Naveau, J. Carpentier, L. Righetti, S. Vijayakumar, and N. Mansard, "Crocoddyl: An efficient and versatile framework for multi-contact optimal control," in *IEEE International Conference on Robotics and Automation*, 2020, pp. 2536–2542.

[34] 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.

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

[36] X. B. Peng, E. Coumans, T. Zhang, T.-W. Lee, J. Tan, and S. Levine, "Learning agile robotic locomotion skills by imitating animals," *arXiv preprint arXiv:2004.00784*, 2020.

[37] I. Radosavovic, T. Xiao, B. Zhang, T. Darrell, J. Malik, and K. Sreenath, "Real-world humanoid locomotion with reinforcement learning," *Science Robotics*, vol. 9, no. 89, p. eadi9579, 2024.

[38] Agility Robotics, "Humanoid robots digit is changing everything the world knows about labor," 2024. [Online]. Available: https://agilityrobotics.com/

[39] X. Gu, Y.-J. Wang, X. Zhu, C. Shi, Y. Guo, Y. Liu, and J. Chen, "Advancing humanoid locomotion: Mastering challenging terrains with denoising world model learning," *arXiv preprint arXiv:2408.14472*, 2024.

[40] S. Martinez, R. Griffin, and C. Mastalli, "Multi-contact inertial estimation and localization in legged robots," *arXiv preprint arXiv:2403.17161*, 2024.