2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

CAIMAN: Causal Action Influence Detection for Sample-Efficient Loco-Manipulation

Yuanchen Yuan[1], Jin Cheng[2], Núria Armengol Urpí[2], Stelian Coros[2]

***Abstract*— Enabling legged robots to perform non-prehensile loco-manipulation is crucial for enhancing their versatility. However, learning behaviors such as whole-body object pushing often necessitates sophisticated planning strategies or extensive task-specific reward shaping. In this work, we present CAIMAN, a practical reinforcement learning framework that encourages the agent to gain *control* over other entities in the environment. CAIMAN leverages causal action influence as an intrinsic motivation objective, allowing legged robots to efficiently acquire object pushing skills even under sparse task rewards. We employ a hierarchical control strategy, combining a low-level locomotion module with a high-level policy that generates task-relevant velocity commands and is trained to maximize the intrinsic reward. To estimate causal action influence, we learn the dynamics of the environment by integrating a kinematic prior with data collected during training. We empirically demonstrate CAIMAN's superior sample efficiency and adaptability to diverse scenarios in simulation, as well as its successful transfer to real-world systems without further fine-tuning. A video demo is available at https://www.youtube.com/watch?v=dNyvT04Cqaw.**

I. INTRODUCTION

Modern day legged robots showcase impressive versatility, from traversing challenging terrains [1, 2] to executing agile maneuvers such as backflips and parkour [3, 4]. Yet as expectations for autonomy grow, enabling these systems to physically interact with their environments remains a central research problem [5, 6, 7]. A common strategy to enhance manipulation capabilities is to equip legged robots with external manipulators for prehensile tasks [8, 9, 10], but such methods are inherently constrained by object size and payload. Harnessing whole-body motion for non-prehensile manipulation offers a promising alternative, but remains a non-trivial challenge.

Traditional approaches for whole-body manipulation explicitly model robot-object interactions, often relying on complex planning and optimization for coordinating motion [11]. These methods require accurate models of both the robot and the environment, which restricts their scalability for high-dimensional systems with complex, stochastic dynamics and limits contact points to predefined regions [12, 13]. Learning-based methods offer a more scalable alternative, improving computational efficiency and reducing dependence on precise object estimation [14, 15, 16]. However, these methods frequently rely on tedious reward shaping or learning curriculums to guide exploration and foster meaningful behavior. Furthermore, the large exploration space of loco-manipulation tasks often necessitates special treatment, such as learning from behavioral priors [17, 18] or task-agnostic explorative rewards [19, 20], to encourage meaningful engagement with the environment.

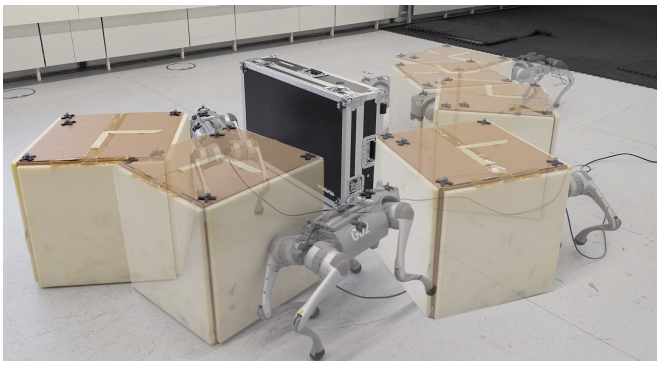


Fig. 1: Robot loco-manipulation in the real world enabled by CAIMAN. The quadruped maneuvers a box around an obstacle to reach a target position.

To alleviate these challenges, we introduce CAIMAN, a framework for training non-prehensile object pushing skills in legged robots. CAIMAN employs a hierarchical control structure that decouples high-level planning from low-level locomotion. We train robust locomotion using an existing pipeline, and we learn the high-level policy with a simple yet effective reward structure consisting of only three terms: a sparse task reward, an action regularizer, and an intrinsically motivated explorative reward. The sparse task reward provides a general signal for various pushing tasks, where the robot is only rewarded for completing the task successfully. The exploratory reward incentivizes the robot to explore and gain control over the environment via Causal Action Influence (CAI) [21], which is a measure quantifying how much influence an agent has on the states of other entities and is computed based on environment dynamics. Instead of learning the dynamics from scratch, CAIMAN combines a simple predefined kinematic prior with learned residual dynamics that compensate for physical interactions beyond the prior, allowing accurate dynamics to be learned efficiently. We show that in a sparse task reward setting, CAIMAN achieves strong learning performance and superior sample efficiency compared to other baselines, including within complex, obstacle-laden scenarios. We also demonstrate that the learned residual accurately captures complex robot-object interactions that are not modeled by the kinematic prior. Finally, we successfully transfer the learned policy to a real quadruped robot, enabling it to perform whole-body object pushing in real-world settings.

[1]Department of Mechanical and Process Engineering, ETH Zurich, Switzerland yuayuan@ethz.ch
[2]Department of Computer Science, ETH Zurich, Switzerland {first.last}@ethz.ch

To summarize, our contribution is three-fold: 1) a general hierarchical framework for learning whole-body object pushing with legged robots in various scenarios, including navigation through obstacles; 2) an intrinsically motivated reward based on causal influence calculated from a combination of a kinematics prior and learned residual dynamics; 3) successful hardware validation on a real-world quadruped robot.

II. RELATED WORK

A. Loco-manipulation on Legged Systems

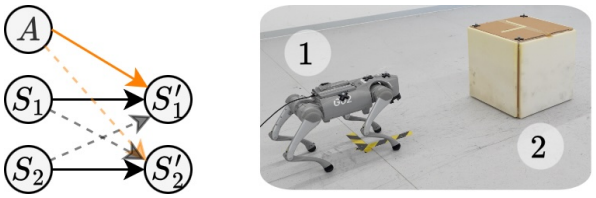
The integration of locomotion and manipulation has gained significant attention as a promising and application-oriented research field for legged robots. Model-based methods that rely on accurate representations of both robot and object to optimize trajectories [9, 11, 13] have shown success in tasks such as box carrying [8, 10], but often require precise state estimation [22, 23] and struggle with scalability due to the complexity of contact modes [24]. Reinforcement learning (RL) is a viable alternative that avoids explicit modeling and has been successfully applied to diverse tasks including soccer dribbling [25, 26], button pushing [15, 27], and door opening [16, 19]. However, achieving effective exploration remains difficult given the large search space of robot-object interactions [14, 28]. To address exploration challenges, researchers have proposed using behavior prior [29, 30] or task-agnostic explorative rewards [19, 20] as mechanisms to guide learning. In addition, hierarchical frameworks [31, 32] decompose tasks into high-level planning and low-level control, enabling effective behaviors across various scenarios [33].

In accordance with previous works, we also implement a hierarchical control framework for whole-body object pushing, decoupling high-level velocity planning from low-level locomotion, similarly to Jeon et al. [34]. We aim to utilize a simple reward structure, as opposed to one that requires sophisticated design effort as seen in [34], to achieve sample-efficient high-level policy training.

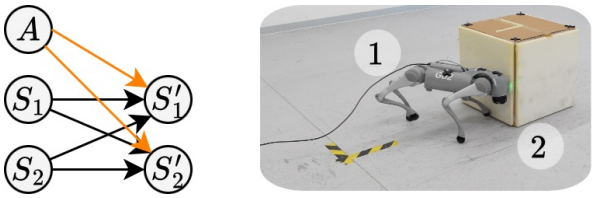
B. Intrinsically Motivated Reinforcement Learning

Intrinsic motivation (IM) [35] plays a vital role in reinforcement learning, especially when extrinsic rewards are sparse or difficult to design. Core IM mechanisms include curiosity [36], learning progress [37], and empowerment [38], each promoting exploration and skill acquisition. Curiosity, often measured via prediction errors in learned world models [39, 40], rewards agents for encountering novel states, and has been previously applied for learning loco-manipulation skills [19, 20]. Learning progress encourages agents to focus on regions in the state space with rapid improvement, supporting curriculum learning and adaptive exploration [41, 42].

Our work builds on empowerment, an information-theoretic quantity defined as the channel capacity between the agent's actions and its sensory observations [38, 43]. Recent work has connected IM to causal reasoning, aiming to improve sample efficiency and interpretability [44, 45].



(a) No influence of A on S'_2



(b) Influence of A on S'_1 and S'_2

Fig. 2: Illustration of the LCM (*left*) for two different environment situations $S = s$ (*right*) in the loco-manipulation task. The LCM captures the transition from S, A to S', factorized into state components. The global SCM is fully connected (dashed and continuous lines), while the LCM $\mathcal{G}_{S=s}$ (continuous lines) is sparser. We are interested in detecting the presence of continuous orange arrows in the LCM, i.e. the influence of the action A on next states S'.

To encourage effective exploration, we employ causal action influence (CAI) [21], a measure of an agent's ability to influence its environment and a conceptual lower bound on empowerment.

III. PRELIMINARIES

We model decision-making in a dynamic environment as a Markov Decision Process (MDP) [46], defined by the tuple $\langle \mathcal{S}, \mathcal{A}, P, R, \gamma \rangle$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ the action space, P the transition kernel, R the reward function, and γ the discount factor. Following the principle of independent causal mechanisms [47], we assume that the world consists of interacting but independent entities. This induces a state space factorization $\mathcal{S} = \mathcal{S}_1 \times \cdots \times \mathcal{S}_N$ for N entities, where each factor $\mathcal{S}_i$ represents the state of entity i. An MDP coupled with a policy $\pi : \mathcal{S} \mapsto \mathcal{A}$ induces a *Structural Causal Model* (SCM) [48] describing the resulting trajectory distribution.

Definition 1 (Structural Causal Model [48]): An SCM is a tuple $(\mathcal{U}, \mathcal{V}, F, P^u)$, where $\mathcal{U}$ is a set of exogenous variables (e.g., latent randomness) sampled from P^u, $\mathcal{V}$ is a set of observed variables (e.g., states, actions, rewards), and F is the set of structural functions capturing the causal relations, such that functions $f_V : \mathrm{Pa}(V) \times \mathcal{U} \to V$, with $\mathrm{Pa}(V) \subset \mathcal{V}$ denoting the set of parents of V, determine the value of endogenous variables V for each $V \in \mathcal{V}$.

SCMs are typically visualized as directed acyclic graphs, where nodes represent variables and edges indicate causal relations, as shown in Fig. 2. The SCM we consider captures one-step transitions with variables $\mathcal{V} = S_1, \ldots, S_N, A, S'_1, \ldots, S'_N$. Due to the Markov property and flow of time, causal dependencies exist only from (S, A) to S'. The global SCM is generally fully connected, encoding all *possible*, however unlikely, interactions between entities (i.e., $S_i/A \to S'_j$ for many i, j) (see Fig. 2). However, most of these dependencies are inactive given a specific state

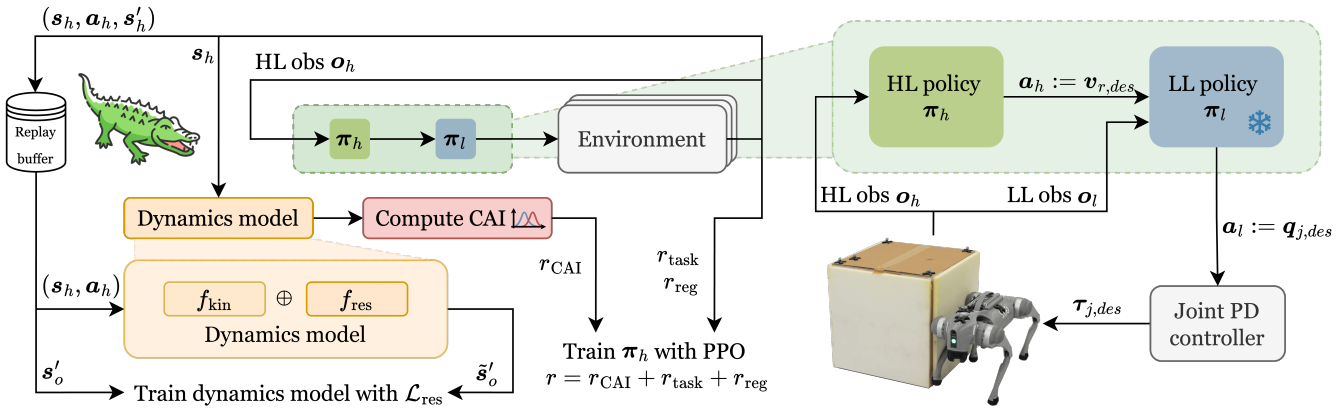


Fig. 3: CAIMAN framework: The high-level (HL) policy generates desired base velocity commands, which are translated into joint commands by a low-level (LL) policy. We utilize a simple kinematic prior and learned residual dynamics to model the robot-object interaction in the environment while providing a CAI-based explorative bonus along with the sparse task reward.

configuration. For example, the robot can only influence an object if it is within a certain proximity to it. To capture such context-specific structure, we use the *Local Causal Model* (LCM) [49].

Definition 2 (Local Causal Model [49]): Given an SCM $(\mathcal{U}, \mathcal{V}, F, P^u)$ and an observation $V = v, V \subset \mathcal{V}$, the local SCM is the SCM with $F_{V=v}$ and graph $\mathcal{G}_{do(V=v)}$ obtained by pruning edges from $\mathcal{G}_{do(V=v)}$ until it is causally minimal.

Intuitively, the LCM is sparser than the global SCM, as it retains only the edges corresponding to entity influences that are active in the current context. This ensures that the graph reflects the agent's *local influences*, rather than all potential ones.

In this work, we build on the insight that agents can effectively learn loco-manipulation tasks when encouraged to gain *control* over their environment. We do so by driving the agent toward states where it can influence other entities, achieved through a principled explicit measure of local influence, the Causal Action Influence (CAI) [21, 50] measure. CAI is a state-dependent measure of control that assesses whether an agent's actions can affect the states of other entities, corresponding graphically to the presence of an edge $A \to S'_j$ in the LCM $\mathcal{G}_{do(S=s)}$. Formally, CAI is defined via point-wise conditional mutual information $I(S'_j; A \mid S = s)$, which equals zero when S'_j is independent of A given $S = s$, i.e. $S'_j \perp\!\!\!\perp A \mid S = s$. At state $S = s$, CAI is given by

$$\begin{aligned} C^j(s) &:= I(S'_j; A \mid S = s) \\ &= \mathbb{E}_{a\sim\pi}\left[D_{KL}\left(P_{S'_j|S=s,A=a} \;||\; P_{S'_j|S=s}\right)\right]. \end{aligned} \tag{1}$$

IV. METHOD

The hierarchical control and training framework adopted by CAIMAN is presented in Fig. 3. It consists of a high-level (HL) policy that generates desired base velocity commands and a low-level policy that translates velocity commands into joint-level actions. In this work, we introduce a novel, sample-efficient approach for learning the high-level policy, while building upon an existing pipeline [51] to train robust low-level policies.

Low-level observation o_l	
${}_{\mathcal{B}}\boldsymbol{v}_r \in \mathbb{R}^3$	Robot linear velocity in base frame $\mathcal{B}$
${}_{\mathcal{B}}\boldsymbol{\omega}_r \in \mathbb{R}^3$	Robot angular velocity in base frame $\mathcal{B}$
$\boldsymbol{q}_j \in \mathbb{R}^{12}$	Joint positions
$\dot{\boldsymbol{q}}_j \in \mathbb{R}^{12}$	Joint velocities
${}_{\mathcal{B}}\boldsymbol{g} \in \mathbb{R}^3$	Projected gravity in base frame $\mathcal{B}$
$\boldsymbol{v}_{r,des} \in \mathbb{R}^3$	Desired velocity command
$\boldsymbol{a}_{l,prev} \in \mathbb{R}^{12}$	Previous action
High-level observation o_h (in world frame $\mathcal{W}$)	
$\boldsymbol{v}_r \in \mathbb{R}^3$	Robot linear velocity
$\boldsymbol{\omega}_r \in \mathbb{R}^3$	Robot angular velocity
$\boldsymbol{\xi}_r = (x_r, y_r, \psi_r) \in \mathbb{R}^3$	Robot pose
$\boldsymbol{\xi}_o = (x_o, y_o, \psi_o) \in \mathbb{R}^3$	Object pose
$\boldsymbol{p}_t = (x_t, y_t) \in \mathbb{R}^2$	Target position
$\boldsymbol{a}_{h,prev} \in \mathbb{R}^3$	Previous action
additional:	
$(x_w, y_w, \psi_w) \in \mathbb{R}^3$	Wall pose (for each wall)
High-level state for CAI s_h (in world frame $\mathcal{W}$)	
$\boldsymbol{\xi}_r = (x_r, y_r, \psi_r) \in \mathbb{R}^3$	Robot pose
$\boldsymbol{\xi}_o = (x_o, y_o, \psi_o) \in \mathbb{R}^3$	Object pose
$\boldsymbol{v}_r = (v_r^x, v_r^y, \omega_r^z) \in \mathbb{R}^3$	Robot velocity
$\boldsymbol{v}_o = (v_o^x, v_o^y, v_o^z) \in \mathbb{R}^3$	Object velocity

TABLE I: Detailed observation space for each module.

A. Low-level Locomotion Policy

The low-level locomotion policy $\boldsymbol{\pi}_l$ is trained to generate target joint positions $\boldsymbol{q}_{j,des} \in \mathbb{R}^{12}$ that track a desired base velocity command $\boldsymbol{v}_{r,des} \in \mathbb{R}^3$. This command, $\boldsymbol{v}_{r,des} = (v^x_{r,des}, v^y_{r,des}, \omega^z_{r,des})$, specifies linear velocities in the longitudinal (x) and lateral (y) directions and a yaw rate, all in the robot frame. The desired joint positions $\boldsymbol{q}_{j,des}$ are tracked using joint-level proportional-derivative (PD) controllers to produce the corresponding joint torques $\boldsymbol{\tau}_{j,des} \in \mathbb{R}^{12}$. The policy is trained with velocity commands sampled uniformly from a predefined range. To ensure robust sim-to-real transfer and hardware performance during contact-rich tasks like pushing, we apply domain randomization and external perturbations following [51]. The complete low-level observation space $\boldsymbol{o}_l$ is shown in Table I.

B. High-level Planning Policy

The high-level policy $\boldsymbol{\pi}_h$ is trained to generate desired robot velocity commands $\boldsymbol{a}_h := \boldsymbol{v}_{r,des}$ that achieve successful task completion in object pushing. Its observation

$\boldsymbol{o}_h$ includes the robot's linear and angular velocities, the poses of the robot and object, the target object position $\boldsymbol{p}_t = (x_p, y_p) \in \mathbb{R}^2$, and the previous action—all expressed in the world frame. In scenarios with obstacles (e.g., walls), $\boldsymbol{o}_h$ also includes their poses in the world frame. The complete high-level observation space $\boldsymbol{o}_h$ is shown in Table I.

We use Proximal Policy Optimization (PPO) [52] as the base RL algorithm, with a simple reward function composed of three terms:

$$r = w_1 \mathbb{1}_{\|\boldsymbol{p}_o - \boldsymbol{p}_t\|_2 < \epsilon} + w_2 r_{\text{CAI}} + w_3 \left\|\boldsymbol{a}_h - \boldsymbol{a}_{h,prev}\right\|_2^2, \quad (2)$$

where $\boldsymbol{p}_o$, $\boldsymbol{p}_t$ are the current and target object positions, ϵ is a success threshold, and $\boldsymbol{a}_h$, $\boldsymbol{a}_{h,prev}$ are the current and previous high-level actions, respectively. The exploration bonus r_{CAI} is derived from the CAI measure C^j from (1), where we choose the object to be the entity of interest j. Resorting to an approximation $\tilde{C}_j$, we compute the reward r_{CAI} at a given state $S = \boldsymbol{s}$ as

$$r_{\text{CAI}} = \tilde{C}_{j=\text{object}}(s) = \frac{1}{K}\sum_{i=1}^{K} D_{\text{KL}}\left(P_{S'_j|S=s,A=a^{(i)}} \middle\| \frac{1}{K}\sum_{k=1}^{K} P_{S'_j|S=s,A=a^{(k)}}\right), \quad (3)$$

given K actions $\{a^{(i)}\}_{i=1}^{K}$ sampled from the policy. In this work, we find it sufficient to use $K = 64$. This approximation estimates the marginal $P_{S'_j|s}$ using Monte Carlo sampling. We model the transition distribution $P_{S'_j|S=s,A=a}$ as a fixed-variance Gaussian (details in Section IV-C), which enables closed-form KL divergence calculation using the Gaussian mixture approximation [53].

The CAI reward encourages the agent to reach states where it exerts greater influence over the object, thereby promoting task-relevant exploration and learning. The weight w_2 for the CAI reward scales with the raw CAI score:

$$w_2 = w_{2,b} + \max\left(0, (r_{\text{CAI}} - \alpha_1)/\alpha_2\right), \quad (4)$$

where α_1 is the threshold for scaling and α_2 controls the scaling rate.

Finally, we foster more directed exploration by injecting time-correlated noise into the action sampling process during training [54, 55]. Sampling from time-correlated actions reduces the possibility of meaningless back-and-forth behavior that could result from commonly used white-noise samples. We select a correlation strength parameterized as $\beta = 0.5$, corresponding to a colored noise between white and pink.

C. Dynamics learning

To calculate the exploration bonus r_{CAI} in (3), we learn the transition model $P_{S'_j|S=s,A=a^{(k)}}$ for all entities j. In our current setting, we focus on a single entity—the pushable object—but the framework naturally extends to multiple entities. Concretely, this reduces to learning the object transition model $P_{\boldsymbol{s}'_o|\boldsymbol{s}_h,\boldsymbol{a}_h}$. The object state $\boldsymbol{s}_o$ is defined as the object's position, while the high-level state $\boldsymbol{s}_h = (\boldsymbol{\xi}_r, \boldsymbol{\xi}_o, \boldsymbol{v}_r, \boldsymbol{v}_o)$ includes the robot's pose $\boldsymbol{\xi}_r$ and velocity $\boldsymbol{v}_r$ and the object's pose $\boldsymbol{\xi}_o$ and velocity $\boldsymbol{v}_o$. The high-level action $\boldsymbol{a}_h$ corresponds to the desired robot velocity, and the next object state $\boldsymbol{s}'_o = \boldsymbol{p}'_o = (x'_o, y'_o)$ denotes its position at the subsequent timestep. Instead of using a pretrained model, CAIMAN leverages data collected from high-level interactions during training to efficiently learn the dynamics. As described earlier, we model the object's transition probability $P_{\boldsymbol{s}'_o|\boldsymbol{s}_h,\boldsymbol{a}_h}$ as a fixed-variance Gaussian distribution $\mathcal{N}(\boldsymbol{s}'_o; f_\theta(\boldsymbol{s}_h, \boldsymbol{a}_h), \sigma^2)$, where f_θ is a neural network that predicts the mean of the distribution.

To enhance learning efficiency, we incorporate a simple kinematic prior model f_{kin}, which estimates the next object position $\tilde{\boldsymbol{p}}'_o$ using geometric reasoning based on the relative pose between robot and object and the commanded velocity. This estimate is computed by projecting the robot's velocity $\boldsymbol{a}_h$ onto the direction from the robot to the object and updating the object's position accordingly:

$$\tilde{\boldsymbol{p}}'_o = \begin{cases} \boldsymbol{p}_o + \delta t \cdot (\boldsymbol{a}_{h,xy} \cdot \delta\hat{\boldsymbol{p}}) \cdot \delta\hat{\boldsymbol{p}}, \\ \qquad \text{if } \|\boldsymbol{p}_o - \boldsymbol{p}_r\|_2 \le \epsilon_p \text{ and } \boldsymbol{a}_{h,xy} \cdot \delta\hat{\boldsymbol{p}} > 0 \\ \boldsymbol{p}_o, \text{ otherwise} \end{cases} \quad (5)$$

Here, $\delta\hat{\boldsymbol{p}} = (\boldsymbol{p}_o - \boldsymbol{p}_r)/\|\boldsymbol{p}_o - \boldsymbol{p}_r\|_2$ is the unit vector pointing from the robot to the object, δt is the high-level control step size, and ϵ_p is a distance threshold. The conditions ensure that the robot is close enough to and moving toward the object.

We combine the kinematic prior with a learned residual model f_{res}, parameterized by θ, to capture complex physical interactions beyond the kinematics model, including nonlinear effects such as friction, drag, and collisions with obstacles. The final dynamics model is thus:

$$f_\theta(\boldsymbol{s}_h, \boldsymbol{a}_h) = f_{\text{kin}}(\boldsymbol{s}_h, \boldsymbol{a}_h) + f_{\text{res}}(\boldsymbol{s}_h, \boldsymbol{a}_h; \theta), \quad (6)$$

where f_{kin} is deterministic and independent of θ. We train the residual model by minimizing the mean squared error between the predicted and true object positions:

$$\mathcal{L}_{\text{res}}(\theta) = \frac{1}{N}\sum_{i=1}^{N} \left\|\tilde{\boldsymbol{s}}'^{(i)}_o - \boldsymbol{s}'^{(i)}_o\right\|_2^2, \quad (7)$$

where $\tilde{\boldsymbol{s}}'^{(i)}_o = f_{\text{kin}}(\boldsymbol{s}_h^{(i)}, \boldsymbol{a}_h^{(i)}) + f_{\text{res}}(\boldsymbol{s}_h^{(i)}, \boldsymbol{a}_h^{(i)}; \theta)$.

Although the high-level action $\boldsymbol{a}_h$ could, in principle, be sampled from the full support of the desired velocity $\boldsymbol{v}_{r,des}$ [21], not all commands are feasible for the low-level controller to execute under the robot's current state. Therefore, we define $\boldsymbol{a}_h = \tilde{\boldsymbol{v}}_r$ to be the achievable velocity, and use $\tilde{\boldsymbol{v}}_r$ for both dynamics learning and CAI computation. Given training samples $\mathcal{D} = (\boldsymbol{s}_h^{(i)}, \boldsymbol{a}_h^{(i)}, \boldsymbol{s}_h'^{(i)})$, we have access to the achieved robot velocity $\boldsymbol{v}'_r$ at the next timestep. When computing CAI as an exploration reward, we assume that the robot's velocity can only change within a limited range over one high-level step. Thus, for any state $\boldsymbol{s}_h$, we sample the action $\boldsymbol{a}_h$ from a bounded range centered on the current velocity:

$$\boldsymbol{a}_h = \tilde{\boldsymbol{v}}_r \sim \mathcal{U}[\boldsymbol{v}_r \pm \delta\boldsymbol{v}_r] \quad (8)$$

where $\delta\boldsymbol{v}_r$ is a fixed range of velocity deviation. The limits were empirically determined to be $(\delta v_r^x, \delta v_r^y, \delta\omega_r^z) = (0.3, 0.3, 0.4)$.

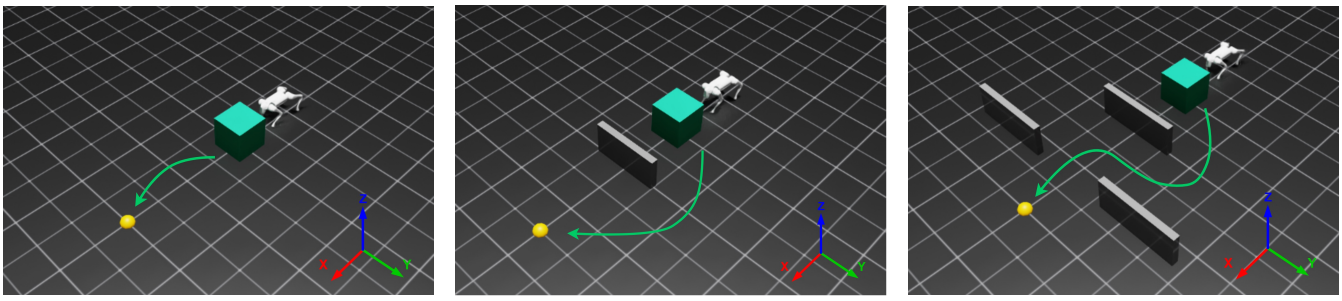


Fig. 4: The `Single Object` (*left*), `Single Wall` (*middle*), and `Multi-Wall` (*right*) tasks. The yellow sphere denotes the object's target position.

Reward functions	
CAI	$r = w_1 r_{\text{task}} + w_2 r_{\text{CAI}} + w_3 r_{\text{reg}}$
RND	$r = w_1 r_{\text{task}} + w_4 r_{\text{RND}} + w_3 r_{\text{reg}}$
heuristics	$r = w_1 r_{\text{task}} + w_5 r_{\text{heu}} + w_3 r_{\text{reg}}$
r_{CAI} weight	Eq. 4

Reward coefficients							
Task	w_1	w_3	w_4	w_5	$w_{2,b}$	α_1	α_2
Single object	15	-5e-3	10	0.01	40	12e-5	1.5e-6
Single wall	40	-5e-3	10	0.01	40	12e-5	1.5e-6
Multi-wall	40	-5e-3	10	0.01	40	12e-5	1.5e-6

TABLE II: Simulation training reward parameters. All CAI methods (`CAI-kinematic`, `CAI-learned`, CAIMAN) use the CAI reward function, all RND methods (`RND-full`, `RND-object`) use the RND reward function, while only the `Heuristics` method uses the heuristics reward function.

V. EXPERIMENTS

We trained both high- and low-level policies using Isaac Sim [56]. The low-level policy operates with a control interval of 0.02 seconds, while the high-level policy has a control interval of 0.2 seconds. At each iteration of the high-level control loop, we advance the environment by 0.2 seconds, compute rewards and terminations, collect transitions for both the high-level and dynamics policies, and augment the rewards buffer with the CAI exploration bonus.

We evaluated CAIMAN on three pushing tasks of increasing difficulty, illustrated in Fig. 4. In the `Single-object` task, the robot pushes a single cuboid object to a fixed target position. The `Single-wall` task introduces a fixed wall that blocks the direct path to the target, requiring the robot to maneuver the object around the obstacle. The `Multi-wall` task further increases complexity with multiple fixed walls through which the robot must navigate the object. The wall obstacles in the single-wall and multi-wall tasks introduce regions in the state space where the object's dynamics become more complex, thereby reducing the robot's ability to exert consistent influence over the object. We trained 1500 iterations for the single object and single wall tasks, and 2000 iterations for the multi-wall task, where each iteration consists of 10 high-level control steps. All tasks have an episode duration of 20 seconds and use a fixed target position.

We compared our approach against several baselines. The `Heuristics` baseline trains the high-level policy using a distance-based heuristic reward, $r_{\text{heu}} = \exp\left(-\|\boldsymbol{p}_r - \boldsymbol{p}_o\|_2\right)$, which encourages the robot to minimize its distance to the object and mirrors the primary exploration reward used in prior work [34]. We further evaluated CAIMAN against intrinsic motivation algorithms, specifically Random Network Distillation (RND) [57], implementing two variants: `RND-full`, which follows the original formulation and computes prediction error over the full state, and `RND-object`, which restricts prediction error to the object state only. For ablation studies, we introduced `CAI-kinematic`, where CAI is computed using only the kinematic prior (Eq. 5), and `CAI-learned`, where CAI relies on a fully learned dynamics model (no prior). The reward functions and corresponding parameters for all tasks and training methods are summarized in Table II. The RND reward term is given as $r_{RND} = \|f(s) - \hat{f}(s)\|_2$, where s is the state to be encoded, and $f(s)$ and $\hat{f}(s)$ denote the target and predictor networks, respectively.

A. Simulation results

We evaluated the learning performance of each method by measuring the success rate for each task. A task is successfully completed if the distance between the object and the target is below a threshold ϵ_s at the end of an episode. The results are shown in Fig. 5.

With only sparse task rewards, task-relevant skill acquisition heavily relies on effective exploration. In the `Single-object` task, all methods—except `RND-full`—achieve some degree of success. CAIMAN, `RND-object`, and `Heuristics` perform the best, while `CAI-kinematic` and `CAI-learned` underperform. The low performance of `CAI-learned` is likely due to the high data and time demands required to learn an accurate dynamics model from scratch. In contrast, `CAI-kinematic` performs comparably with the top methods, suggesting that the naive kinematic prior is sufficient for simpler tasks with straightforward object interactions. In the more complex `Single-wall` and `Multi-wall` tasks, only CAIMAN and `RND-object` demonstrate notable success, with `RND-object` showing lower asymptotic performance than CAIMAN in both scenarios. This highlights CAIMAN's unique ability to guide task-relevant exploration even in the presence of obstacles, validating its effectiveness in cluttered environments and independence from dense extrinsic rewards. The failure of `CAI-kinematic` and `CAI-learned` in these tasks emphasizes the importance of combining a kinematic prior with a learned residual model to efficiently capture the complex object dynamics

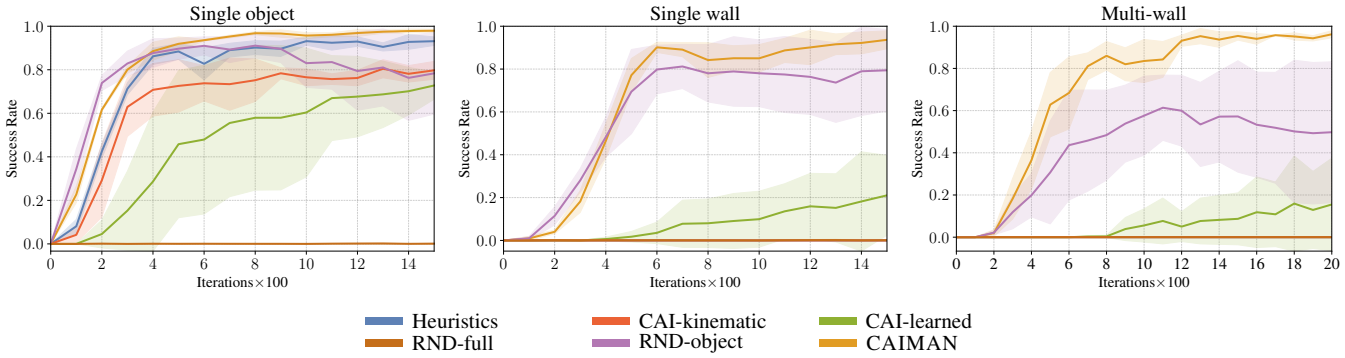


Fig. 5: Policy success rate evaluated at every 100 training iterations for all methods and tasks. Results are evaluated across 800 episodes and averaged over 3 seeds, shaded area represents standard deviation.

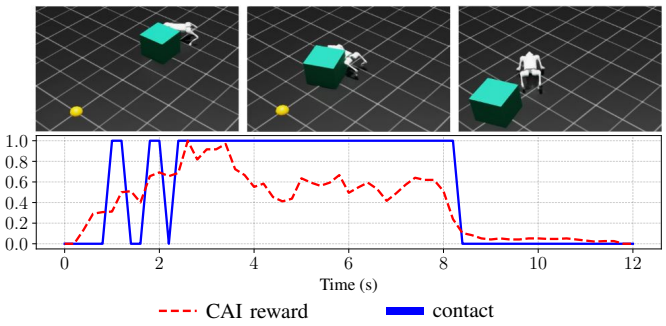


Fig. 6: Scaled CAI reward overlaid with a binary indicator of robot–object contact (1-contact, 0-no contact) over one episode of the `Single-object` task. Representative environment frames are shown above.

needed for effective exploration.

`RND-full` results in agents randomly exploring the state space without yielding meaningful behavior in all tasks. This finding reinforces the need for object-centric exploration bonuses when the skill to be learned involves physical interaction with objects. The performance gap between `RND-object` and CAIMAN can be partially attributed to the phenomenon of *detachment* [58], where the agent drifts away from reward-depleted areas and continuously pursues new regions with higher intrinsic reward, potentially losing avenues of task-relevant exploration.

To validate CAI as an effective exploration signal, Fig. 6 explicitly plots the raw CAI reward alongside robot-object interactions in the `Single-object` task. Elevated CAI values during contact states empirically confirm the role of CAI in guiding exploration.

B. Loco-manipulation of irregular objects

To demonstrate CAIMAN's effectiveness in handling complex objects with asymmetric dynamics, we conducted an additional experiment using a simplified cart with two normal and two caster wheels, as illustrated in Fig. 7. The scene configuration is identical to that of the `Single-object` task, with the cuboid being replaced by the cart. Our results show that CAIMAN is more sample efficient and achieves a higher asymptotic performance (approx. 90% success rate) than most baselines and is on par with the best competitor. In particular, this experiment did not require any changes to the original framework or to the kinematics prior, which was derived assuming an object with symmetric motion. These

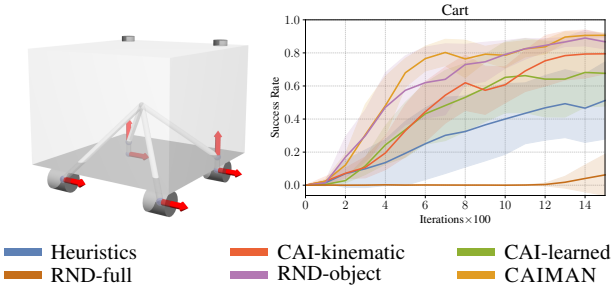


Fig. 7: *Left*: A simplified cart object. *Right*: Single cart pushing results.

results highlight the ability of the learned model to accurately capture complex dynamics and confirm CAIMAN's capability in learning with irregular objects. This environment has no obstacles, but we hypothesize that adding walls would further emphasize CAIMAN's advantages over other baselines.

C. Leveraging pretrained dynamics for new tasks

Since our framework learns the environment dynamics, we propose that the dynamics residual obtained from a previous training can be reused across different tasks—as long as the underlying dynamics remain consistent. We hypothesize that leveraging a pretrained model improves the accuracy of CAI estimates in the early stages of training, thus enhancing the exploration signal and significantly boosting sample efficiency.

To test this hypothesis, we consider a generalization of the `Single-wall` task in which the target position is randomized instead of fixed, sampled uniformly within a predefined area as illustrated in Fig. 8. We compare the original CAIMAN, which learns the dynamics residual from scratch, to one that reuses a pretrained residual model. For the latter, we initialize the residual with one from the original `Single-wall` task and continuously fine-tune it during training with data collected from the new generalized task.

As shown in Fig. 8, the reuse of learned dynamics significantly accelerates learning with sparse rewards. The pretrained model yields more informative CAI estimates, guiding the robot toward meaningful interaction behaviors early in training. This leads to substantial gains in sample efficiency, supporting our hypothesis and demonstrating the benefits of transferring learned dynamics across related tasks.

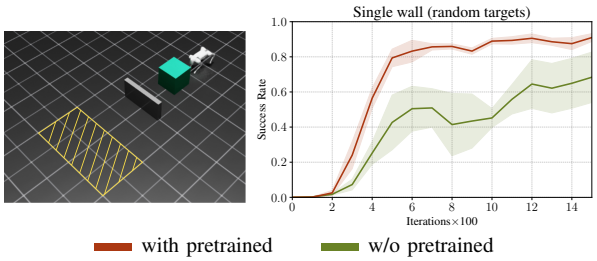


Fig. 8: *Left*: Single wall task with target positions randomly sampled within an area in front of the wall. *Right*: Learning the single wall random target task using models without pretraining (learned from scratch) and models pretrained from the fixed target task.

D. Hardware deployment

We validated our trained policies on real-world quadruped pushing tasks with the Unitree Go2, as shown in Fig. 9. To bridge the sim-to-real gap, we applied domain randomization to the object's mass and friction during high-level policy training. To obtain the observations for the high-level policies, all entities in the environment were tracked using an external motion capture system. Our trained policies were directly deployed to the robot and successfully executed pushing tasks without requiring additional fine-tuning. For further details and visualizations of our trained policies in simulation and on hardware, please refer to the supplementary video.

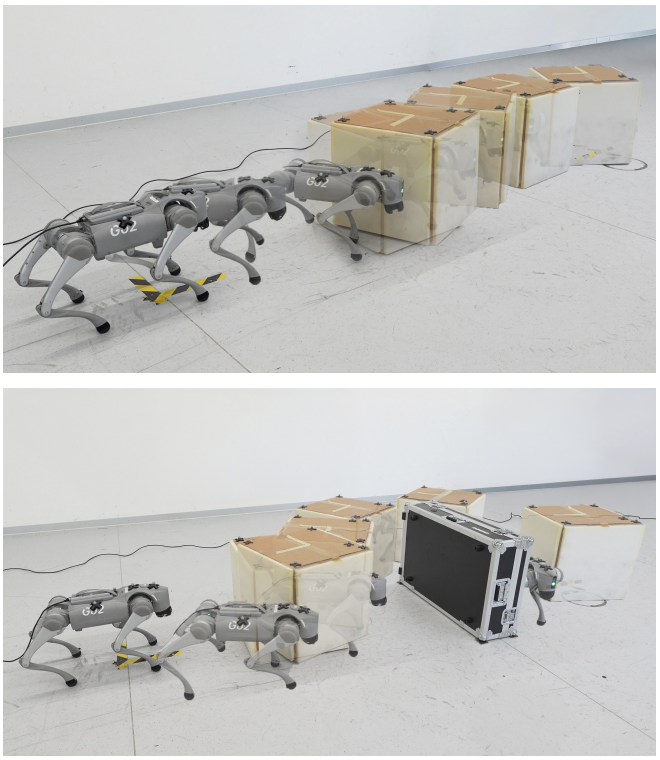


Fig. 9: Snapshots from hardware deployment for the single object and single wall tasks. The mass of the box is 5.5 kilograms with a dimension of $(0.55, 0.55, 0.5)$ meters.

VI. CONCLUSION

We presented CAIMAN, a general framework for training whole-body pushing skills in legged robots. CAIMAN adopts a hierarchical control strategy that decouples locomotion from high-level planning and introduces an intrinsic CAI-based exploration bonus, encouraging control over relevant entities without hand-crafted reward shaping. The CAI computation is bootstrapped with a simple yet effective kinematic prior that is refined by a learned residual dynamics model. Across diverse quadruped pushing tasks, CAIMAN outperforms competitive baselines in sample efficiency, especially in obstacle-rich scenarios, and moreover generalizes to irregular objects. We also showed that the learned dynamics residual accelerates training in new tasks, highlighting the transferability of our approach. Finally, hardware experiments demonstrate the seamless deployment of policies trained with CAIMAN on a real quadruped. Overall, CAIMAN provides a robust and scalable solution for physically grounded manipulation behaviors in legged robots, paving the way for more autonomous and adaptable robotic systems.

REFERENCES

[1] T. Miki, J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter, "Learning robust perceptive locomotion for quadrupedal robots in the wild," *Science robotics*, vol. 7, no. 62, p. eabk2822, 2022.

[2] S. Choi, G. Ji, J. Park, H. Kim, J. Mun, J. H. Lee, and J. Hwangbo, "Learning quadrupedal locomotion on deformable terrain," *Science Robotics*, vol. 8, no. 74, p. eade2256, 2023.

[3] B. Katz, J. Di Carlo, and S. Kim, "Mini cheetah: A platform for pushing the limits of dynamic quadruped control," in *2019 international conference on robotics and automation (ICRA)*. IEEE, 2019, pp. 6295–6301.

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

[5] S. Ha, J. Lee, M. van de Panne, Z. Xie, W. Yu, and M. Khadiv, "Learning-based legged locomotion; state of the art and future perspectives," *arXiv preprint arXiv:2406.01152*, 2024.

[6] C. Tang, B. Abbatematteo, J. Hu, R. Chandra, R. Martín-Martín, and P. Stone, "Deep reinforcement learning for robotics: A survey of real-world successes," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 8, 2024.

[7] Z. Gu, J. Li, W. Shen, W. Yu, Z. Xie, S. McCrory, X. Cheng, A. Shamsah, R. Griffin, C. K. Liu, *et al.*, "Humanoid locomotion and manipulation: Current progress and challenges in control, planning, and learning," *arXiv preprint arXiv:2501.02116*, 2025.

[8] C. D. Bellicoso, K. Krämer, M. Stäuble, D. Sako, F. Jenelten, M. Bjelonic, and M. Hutter, "Alma-articulated locomotion and manipulation for a torque-controllable robot," in *2019 International conference on robotics and automation (ICRA)*. IEEE, 2019, pp. 8477–8483.

[9] S. Zimmermann, R. Poranne, and S. Coros, "Go fetch!-dynamic grasps using boston dynamics spot with external robotic arm," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 4488–4494.

[10] 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.

[11] M. Murooka, S. Nozawa, Y. Kakiuchi, K. Okada, and M. Inaba, "Whole-body pushing manipulation with contact posture planning of large and heavy object for humanoid robot," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 5682–5689.

[12] E. Farnioli, M. Gabiccini, and A. Bicchi, "Toward whole-body loco-manipulation: Experimental results on multi-contact interaction with the walk-man robot," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016, pp. 1372–1379.

[13] A. Rigo, Y. Chen, S. K. Gupta, and Q. Nguyen, "Contact optimization for non-prehensile loco-manipulation via hierarchical model predictive control," in *2023 ieee international conference on robotics and*

automation (icra). IEEE, 2023, pp. 9945–9951.
[14] F. Shi, T. Homberger, J. Lee, T. Miki, M. Zhao, F. Farshidian, K. Okada, M. Inaba, and M. Hutter, "Circus anymal: A quadruped learning dexterous manipulation with its limbs," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 2316–2323.
[15] X. Cheng, A. Kumar, and D. Pathak, "Legs as manipulator: Pushing quadrupedal agility beyond locomotion," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 5106–5112.
[16] P. Arm, M. Mittal, H. Kolvenbach, and M. Hutter, "Pedipulate: Enabling manipulation skills using a quadruped robot's leg," in *41st IEEE Conference on Robotics and Automation (ICRA 2024)*, 2024.
[17] H. Ha, Y. Gao, Z. Fu, J. Tan, and S. Song, "Umi on legs: Making manipulation policies mobile with manipulation-centric whole-body controllers," *arXiv preprint arXiv:2407.10353*, 2024.
[18] N. A. Urpí, M. Bagatella, O. Hilliges, G. Martius, and S. Coros, "Efficient learning of high level plans from play," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 10 189–10 196.
[19] C. Schwarke, V. Klemm, M. Van der Boon, M. Bjelonic, and M. Hutter, "Curiosity-driven learning of joint locomotion and manipulation tasks," in *Proceedings of The 7th Conference on Robot Learning*, vol. 229. PMLR, 2023, pp. 2594–2610.
[20] C. Zhang, W. Xiao, T. He, and G. Shi, "Wococo: Learning whole-body humanoid control with sequential contacts," *arXiv preprint arXiv:2406.06005*, 2024.
[21] M. Seitzer, B. Schölkopf, and G. Martius, "Causal influence detection for improving efficiency in reinforcement learning," *Advances in Neural Information Processing Systems*, vol. 34, pp. 22 905–22 918, 2021.
[22] C. Lin, X. Liu, Y. Yang, Y. Niu, W. Yu, T. Zhang, J. Tan, B. Boots, and D. Zhao, "Locoman: Advancing versatile quadrupedal dexterity with lightweight loco-manipulators," *arXiv preprint arXiv:2403.18197*, 2024.
[23] A. Schakkal, G. Bellegarda, and A. Ijspeert, "Dynamic object catching with quadruped robot front legs," in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2024, pp. 6848–6855.
[24] X. Cheng, S. Patil, Z. Temel, O. Kroemer, and M. T. Mason, "Enhancing dexterity in robotic manipulation via hierarchical contact exploration," *IEEE Robotics and Automation Letters*, vol. 9, no. 1, pp. 390–397, 2023.
[25] Y. Ji, G. B. Margolis, and P. Agrawal, "Dribblebot: Dynamic legged manipulation in the wild," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 5155–5162.
[26] Y. Hu, K. Wen, and F. Yu, "Dexdribbler: Learning dexterous soccer manipulation via dynamic supervision," *arXiv preprint arXiv:2403.14300*, 2024.
[27] Z. He, K. Lei, Y. Ze, K. Sreenath, Z. Li, and H. Xu, "Learning visual quadrupedal loco-manipulation from demonstrations," *arXiv preprint arXiv:2403.20328*, 2024.
[28] Z. Fu, X. Cheng, and D. Pathak, "Deep whole-body control: learning a unified policy for manipulation and locomotion," in *Conference on Robot Learning*. PMLR, 2023, pp. 138–149.
[29] J.-P. Sleiman, M. Mittal, and M. Hutter, "Guided reinforcement learning for robust multi-contact loco-manipulation," in *8th Annual Conference on Robot Learning (CoRL 2024)*, 2024.
[30] F. Liu, Z. Gu, Y. Cai, Z. Zhou, S. Zhao, H. Jung, S. Ha, Y. Chen, D. Xu, and Y. Zhao, "Opt2skill: Imitating dynamically-feasible whole-body trajectories for versatile humanoid loco-manipulation," *arXiv preprint arXiv:2409.20514*, 2024.
[31] A. Rigo, M. Hu, S. K. Gupta, and Q. Nguyen, "Hierarchical optimization-based control for whole-body loco-manipulation of heavy objects," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 15 322–15 328.
[32] J. Wang, R. Dai, W. Wang, L. Rossini, F. Ruscelli, and N. Tsagarakis, "Hypermotion: Learning hybrid behavior planning for autonomous loco-manipulation," in *8th Annual Conference on Robot Learning*, 2024.
[33] K. N. Kumar, I. Essa, and S. Ha, "Cascaded compositional residual learning for complex interactive behaviors," *IEEE Robotics and Automation Letters*, vol. 8, no. 8, pp. 4601–4608, 2023.
[34] S. Jeon, M. Jung, S. Choi, B. Kim, and J. Hwangbo, "Learning whole-body manipulation for quadrupedal robot," *IEEE Robotics and Automation Letters*, vol. 9, no. 1, pp. 699–706, 2023.
[35] R. Rm, "Intrinsic and extrinsic motivations: Classic definitions and new directions," *Contemporary educational psychology*, vol. 25, pp. 54–67, 2000.
[36] J. Schmidhuber, "A possibility for implementing curiosity and boredom in model-building neural controllers," in *Proceedings of the first international conference on simulation of adaptive behavior on From animals to animats*, 1991, pp. 222–227.
[37] ——, "Formal theory of creativity, fun, and intrinsic motivation (1990–2010)," *IEEE transactions on autonomous mental development*, vol. 2, no. 3, pp. 230–247, 2010.
[38] A. S. Klyubin, D. Polani, and C. L. Nehaniv, "Empowerment: A universal agent-centric measure of control," in *2005 ieee congress on evolutionary computation*, vol. 1. IEEE, 2005, pp. 128–135.
[39] D. Pathak, P. Agrawal, A. A. Efros, and T. Darrell, "Curiosity-driven exploration by self-supervised prediction," in *International conference on machine learning*. PMLR, 2017, pp. 2778–2787.
[40] Y. Burda, H. Edwards, A. Storkey, and O. Klimov, "Exploration by random network distillation," in *Seventh International Conference on Learning Representations*, 2019, pp. 1–17.
[41] S. Blaes, M. Vlastelica Pogančić, J. Zhu, and G. Martius, "Control what you can: Intrinsically motivated task-planning agent," *Advances in Neural Information Processing Systems*, vol. 32, 2019.
[42] C. Colas, P. Fournier, M. Chetouani, O. Sigaud, and P.-Y. Oudeyer, "Curious: intrinsically motivated modular multi-goal reinforcement learning," in *International conference on machine learning*. PMLR, 2019, pp. 1331–1340.
[43] S. Mohamed and D. Jimenez Rezende, "Variational information maximisation for intrinsically motivated reinforcement learning," *Advances in neural information processing systems*, vol. 28, 2015.
[44] L. Buesing, T. Weber, Y. Zwols, S. Racaniere, A. Guez, J.-B. Lespiau, and N. Heess, "Woulda, coulda, shoulda: Counterfactually-guided policy search," *arXiv preprint arXiv:1811.06272*, 2018.
[45] S. A. Sontakke, A. Mehrjou, L. Itti, and B. Schölkopf, "Causal curiosity: Rl agents discovering self-supervised experiments for causal representation learning," in *International conference on machine learning*. PMLR, 2021, pp. 9848–9858.
[46] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.
[47] J. Peters, D. Janzing, and B. Schölkopf, *Elements of causal inference: foundations and learning algorithms*. The MIT Press, 2017.
[48] J. Pearl, *Causality*. Cambridge university press, 2009.
[49] S. Pitis, E. Creager, and A. Garg, "Counterfactual data augmentation using locally factored dynamics," *Advances in Neural Information Processing Systems*, vol. 33, pp. 3976–3990, 2020.
[50] N. A. Urpí, M. Bagatella, M. Vlastelica, and G. Martius, "Causal action influence aware counterfactual data augmentation," in *Forty-first International Conference on Machine Learning*, 2024.
[51] N. Rudin, D. Hoeller, P. Reist, and M. Hutter, "Learning to walk in minutes using massively parallel deep reinforcement learning," in *Conference on Robot Learning*. PMLR, 2022, pp. 91–100.
[52] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.
[53] J.-L. Durrieu, J.-P. Thiran, and F. Kelly, "Lower and upper bounds for approximation of the kullback-leibler divergence between gaussian mixture models," in *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Ieee, 2012, pp. 4833–4836.
[54] O. Eberhard, J. Hollenstein, C. Pinneri, and G. Martius, "Pink noise is all you need: Colored noise exploration in deep reinforcement learning," in *The Eleventh International Conference on Learning Representations*, 2023.
[55] J. Hollenstein, G. Martius, and J. Piater, "Colored noise in ppo: Improved exploration and performance through correlated action sampling," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38(11), 2024, pp. 12 466–12 472.
[56] M. Mittal, C. Yu, Q. Yu, J. Liu, N. Rudin, D. Hoeller, J. L. Yuan, R. Singh, Y. Guo, H. Mazhar, *et al.*, "Orbit: A unified simulation framework for interactive robot learning environments," *IEEE Robotics and Automation Letters*, vol. 8, no. 6, pp. 3740–3747, 2023.
[57] Y. Burda, H. Edwards, A. Storkey, and O. Klimov, "Exploration by random network distillation," *arXiv preprint arXiv:1810.12894*, 2018.
[58] A. Ecoffet, J. Huizinga, J. Lehman, K. O. Stanley, and J. Clune, "Go-explore: a new approach for hard-exploration problems," *arXiv preprint arXiv:1901.10995*, 2019.