Learning to Act Through Contact: A Unified View of Multi-Task Robot Learning

Shafeef Omar, Majid Khadiv
Applied and Theoretical Aspects of Robot Intelligence (ATARI) Lab,
Munich Institute of Robotics and Machine Intelligence,
Technical University of Munich, Germany
firstname.lastname@tum.de

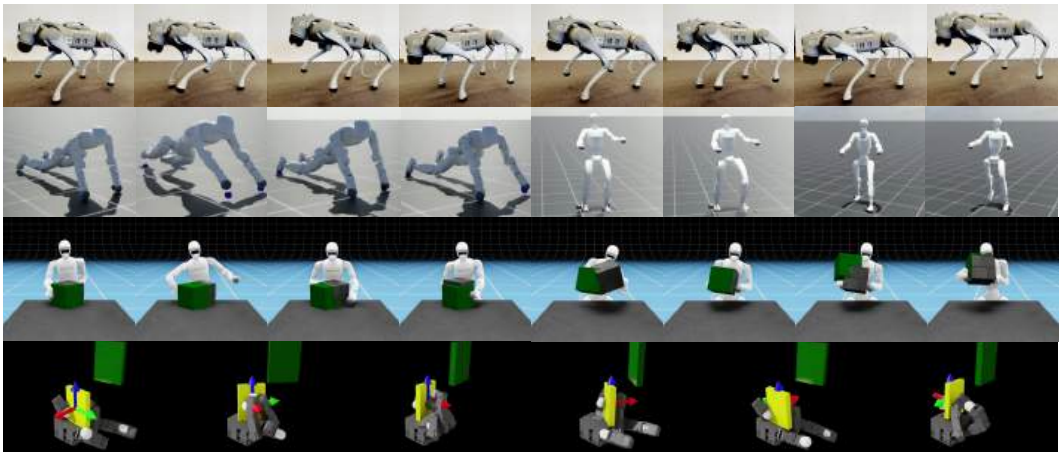

Figure 1: Snapshots of our contact-explicit framework in action. (Row 1) the quadruped demonstrates diverse gaits; (Row 2) the humanoid demonstrates locomotion using bipedal gaits such as walk, jump and hand-assisted gaits such as quadrupedal jump, pace; (Row 3): the humanoid carries out different bimanual manipulation tasks, where the green box represents the target pose; (Row 4): the dexterous hand performing in-hand object reorientation.

Abstract: We present a unified framework for *multi-task* locomotion and manipulation policy learning grounded in contact-explicit representations. Instead of designing different policies for different tasks, our approach unifies the definition of a task through a sequence of contact goals–desired contact positions, timings, and active end-effectors. This enables leveraging the shared structure across diverse contact-rich tasks, leading to a single policy that can perform a wide range of tasks. In particular, we train a goal-conditioned reinforcement learning (RL) policy to realize given contact plans. We validate our framework on multiple robotic embodiments and tasks: a quadruped performing multiple gaits, a humanoid performing multiple biped and quadrupedal gaits, a humanoid executing different bimanual object manipulation tasks, and a dextrous hand performing in-hand manipulation. Each robot is controlled by a single policy trained to execute different tasks grounded in contacts, demonstrating versatile and robust behaviors across morphologically distinct systems. Our results show that explicit contact reasoning significantly improves generalization to unseen scenarios, positioning contact-explicit policy learning as a promising foundation for scalable loco-manipulation. Video available at: https://youtu.be/chKcB8Un22w

Keywords: Goal-Conditioned RL, Task-Agnostic Policy, Contact-Explicit

9th Conference on Robot Learning (CoRL 2025), Seoul, Korea.

1 Introduction

Advances in reinforcement learning (RL) have enabled robots to master complex motor skills, from agile quadruped locomotion [1, 2] to dexterous object manipulation [3, 4, 5]. Yet, prevailing RL policies are often trained with task-specific objectives, making them difficult to transfer to unseen scenarios without retraining from scratch. For example, perceptive locomotion policies are typically tasked to train on velocity and/or position commands that work well on various rough terrain scenarios they have been trained on [6, 7]. Nonetheless, they cannot be directly transferred to environments with sparser footholds and riskier terrains [8], such as stepping stones, even though the required motions are similar to those on which it has been trained on. Similarly, in object manipulation, tasks like lifting an object from a table share similar motor skills with more complex tasks, such as stacking objects. However, the traditional approach of training policies on specific tasks makes it difficult to generalize across different types of physical interactions. Moreover, training a robot on a new task from scratch is not feasible each time it encounters one. This motivates seeking a more fundamental abstraction: one that also unifies locomotion and manipulation. In this light, we propose using *contacts* as a common representation to enable better generalization and adaptability across various physical interaction tasks and embodiments.

Why Contacts? Contacts govern nearly all loco-manipulative behaviors. Locomotion requires coordinated foot placements with the ground, and manipulation relies on purposeful hand-object interactions—both inherently contact-driven. Despite their centrality, contacts are often treated as incidental in reinforcement learning (RL) frameworks, emerging implicitly as a byproduct of motion optimization. This abstraction leads to policies with limited generalization across tasks that share underlying motor principles. In contrast, humans intuitively decompose complex behaviors into contact-explicit subgoals: a climber plans handholds and footholds before ascending, and a parkour athlete sequences hand and foot placements relative to environmental features. These skills transfer seamlessly across structurally similar tasks.

While recent works have explored goal representations such as 3D position targets [9] or motion references [10, 5], they often overlook contact as a fundamental primitive. As a result, such approaches may struggle with tasks where contact timing and placement are crucial. Recent works have shown the benefits of explicitly incorporating contact—either in rewards or task representations [11, 12, 13]—leading to better generalization and performance.

Our work builds on this contact-explicit paradigm, proposing contact goals as a unified task representation for locomotion and contact-rich manipulation, enabling a single policy to produce diverse physical behaviors. We decompose tasks into contact goals—defined by *target locations, timings*, and *active end-effectors*—together with object pose targets for manipulation. Given a high-level planner that generates sequences of these goals, a goal-conditioned RL policy learns to achieve them via joint torques across contact modes. This bypasses task-specific reward design, instead treating contacts as fundamental physical primitives which robots interact with their environment.

We validate our framework with four demonstrations, each with a single policy: (1) on a quadruped robot to execute multiple gaits (trot, pace, bound, jump, and crawl), (2) on a humanoid robot performing multiple biped and quadruped gaits (3) on a humanoid robot to perform bi-manual object manipulation such as object reorientation on a table and object pose tracking while being lifted and (4) on a dexterous robotic hand to perform in-hand object manipulation. Our results demonstrate that using our contact-explicit representation, we can generate multiple skills with a single policy and leverage the shared information between tasks to improve generalization beyond the training distribution.

Our contributions can be summarised as follows:

1. We propose a goal-conditioned RL framework that learns to achieve given contact goals with the world. Using this framework, we train a single policy capable of performing multiple tasks.

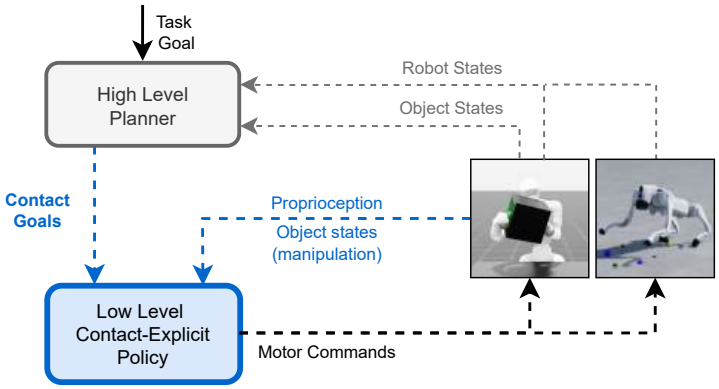

Figure 2: Overview of our contact-explicit framework. A high-level planner generates the contact goals (and object pose targets for manipulation), that is provided as immediate goals for the goal-conditioned RL policy to accomplish.

2. We provide empirical validation on three distinct robot embodiments and various tasks, showing that explicit contact reasoning enables dynamic and robust behaviors across diverse scenarios, generalizing to unseen tasks.

2 Related Work

While locomotion and contact-rich manipulation are similarly realized through intermittent contacts, RL-based methods use different specialized rewards and task representations for each problem.

Locomotion. Early works represented predefined gaits (walking) with step location and robot heading as input [14]. More recent works avoid predefining gaits, and define the desired behavior (goal) through a desired average velocity [15, 6] or reaching a desired position in the world [16, 1, 2]. As this goal representation does not distinguish between different gaits, it is not suitable for multi-gait policy generation. To enable one policy for multiple gaits, recent approaches used either a notion of gait phase as input to the policy [17, 18] or task-specific desired reference motion [19, 20, 21, 22]. The former representation is specific to locomotion and cyclic gaits, while the latter requires another module to generate desired trajectories for every behavior. Unlike these approaches, our findings reveal that we do not need to use any parameterisation to represent the different gaits, but rather contact goals that can directly achieve them.

Manipulation/Loco-Manipulation. In manipulation, the desired behavior is usually specified through the desired object goal [3, 23] or task-specific goals such as for grasping [24, 4]. While successful in learning single tasks, such a representation fails to work in most multi-task and the few-shot learning settings [25]. In loco-manipulation settings, most works simply concatenate separate locomotion and manipulation goals [26, 27] or define and train different tasks separately [28, 29, 30]. However, such an approach does not enable leveraging the shared structure between locomotion and manipulation through contact. [9] trained whole-body controllers using 3D position targets for the robot's hands and then trained a high-level planner using these to perform loco-manipulation. They also released a suite of complex tasks as a benchmark for robot loco-manipulation. However, their low-level reaching policy ignores contacts, which are crucial for loco-manipulation and, consequently, fails in many tasks on their benchmark.

Contacts. Recent studies have shown that including contact information in the reward design and task representation can improve multi-task learning [11, 12, 13]. In particular, [12] showed that a contact-centric representation for multi-gait locomotion learning improves the generalization capability of the gaits when compared to other representations. However, they only showed locomotion results in a behavioral cloning setting. [11] used the contact information in the reward design for various locomotion and loco-manipulation tasks. They proposed sparse contact-based rewards that

are then combined with task-specific rewards to enable complex motions such as humanoid parkour and loco-manipulation. Compared to [11], which learns different policies for different tasks, we show that training one multi-skill policy outperforms the generalization capabilities of the policy to unseen tasks. Furthermore, different from [11], we present a denser reward for contact that facilitates the training procedure and qualitatively produces smoother motions. Closest to our work is a recent study that also uses contact and object pose goals to train an RL policy only to perform bimanual dextrous manipulation [13]. We show that our proposed contact-conditioned policy generalizes better than the (sub)task-conditioned policies in [13] for object manipulation. Furthermore, we show that contact-conditioned policies are general enough to be applied to the locomotion setting as well.

3 Method

3.1 Overview

At the core, we propose a contact-explicit representation that is used to train policies for multi-gait locomotion on a quadruped and multi-task bimanual manipulation on a humanoid robot. In particular, we train a goal-conditioned RL policy to track contact goals, provided by a planner as shown in Fig. 2. The **contact goals** for an end-effector e, $g_e^{\text{con}} = \{p_e^{\text{con}}, S_e^{\text{con}}, \mathcal{I}_e^{\text{con}}\}$, correspond to the 3D location of contact, the command duration, and a binary indicator to be in contact, respectively. In the case of manipulation, g_e additionally comprises the object's goal pose $\{p_{obj}, \theta_{obj}\}$. A new set of contact goals is chosen when the command duration expires. By composing several different contact plans, we can perform various long-horizon tasks. In this work, we have prespecified the contact goals required to achieve the various tasks. However, our method can be integrated with more sophisticated learned contact planners [31, 32], or even contact goals extracted from images/videos [33].

We formalize the problem of finding a multi-task policy $\pi(a_t|s_t, g_t)$ as a goal-conditioned RL problem which is formulated as a Markov Decision Process (MDP) $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \gamma, \mathcal{G} \rangle$. This MDP is defined by states $s_t \in \mathcal{S}$, actions $a_t \in \mathcal{A}$, transtion probability $\mathcal{T}$, a reward $r_t \in \mathcal{R}$, a discount factor γ and goal $g_t \in \mathcal{G}$. The reward r_t is calculated to achieve different contact modes by following the immediate contact goals. The policy aims to maximise the expected return for achieving the contact goal g_t:

$$\max \mathbb{E}_\pi \left[\sum_t \gamma^t r_t(s_t, a_t, g_t) \right]$$

3.2 Learning to Act Through Contact

Contact Phases. We consider three phases for contact that allow us to make or break contact with the environment in a controlled manner using any end-effector, as illustrated in Fig. 3: reach (R), hold (H) and detach (D). During the reach phase of an end-effector, the robot must guide it to a desired contact location provided by the high-level planner. During the hold phase of an end-effector, the robot must maintain its contact where it was guided to during the reach phase. And the last is the detach phase of an end-effector, where the robot is free to move it as long as it does not engage in contact. The detach phase is always followed by a reach phase, which is further followed by a hold phase. Viewing contacts from this perspective, we can develop dense rewards that allow the robot to explore several contact modes by making and breaking contact with the world.

For an end-effector e at time t, the contact goals from the high-level planner comprise the following information: contact locations with a short horizon of two contact switches, $p_{t,e}^{\text{con}} = \left(\left[p_{t,e}^{\text{con}} \right]_1, \left[p_{t,e}^{\text{con}} \right]_2 \right)$, a binary indicator of contact for two contact switches, $(\mathcal{I}_{t,e}^{\text{con}} = \left(\left[\mathcal{I}_{t,e}^{\text{con}} \right]_1, \left[\mathcal{I}_{t,e}^{\text{con}} \right]_2 \right)$ and the command duration S of the current contact goal to be achieved. The contact phase of an end-effector is determined using the binary indicator, $\left[\mathcal{I}_{t,e}^{\text{con}} \right]_1$, and the time remaining to finish the contact command, s (where s is reset to the value of the newly sampled command duration when the previous one expires). If the remaining command duration is less than

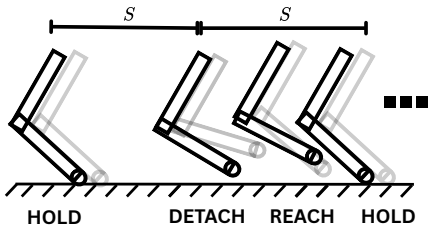

Figure 3: Simple illustration of a robot leg's contact phases during locomotion, for a fixed command duration S. The detach phase is the initial swing phase of the leg, the reach final is the final swing phase of the leg, where it is guided towards the desired contact location, and the hold phase is the stance phase of the leg. Similarly, every contact interaction considered in this work can be viewed in this light.

a threshold δ and the binary contact indicator is 0, we have the reach phase ($\left[\mathcal{I}^{\text{con}}_{t,e}\right]_1 = 0$ and $s < \delta$). If the binary contact indicator is 1, we have the hold phase ($\left[\mathcal{I}^{\text{con}}_{t,e}\right]_1 = 1$). We have the detach phase if the binary contact indicator is 0 and the remaining command duration is greater than the threshold δ ($\left[\mathcal{I}^{\text{con}}_{t,e}\right]_1 = 0$ and $s > \delta$). The binary contact indicators of all the end-effectors, $\mathcal{I}^{\text{con}}_{t,e}$, are stacked to form the contact sequence, referred to in the paper.

Policy Observations and Rewards. Apart from the proprioceptive inputs for the policy's observations, we provide contact goals, $p^{\text{con}}_{t,e}$, $\mathcal{I}^{\text{con}}_{t,e}$, and s, to achieve multiple tasks using the same shared structure. The following rewards are used to achieve the various contact phases described previously:

$$r^{\text{reach}}_{t,e} = \epsilon_{\text{reach}} \cdot \left(1 - \tanh\left(\frac{d\left(\left[p^{\text{con}}_{t,e}\right]_1, p^{\text{act}}_{t,e}\right)}{\sigma}\right)\right) \cdot \mathbb{I}\left[\left[\mathcal{I}^{\text{con}}_{t,e}\right]_1 = 0 \wedge s \leq \delta\right] \tag{1}$$

$$r^{\text{hold}}_{t,e} = \left(1 + \alpha_{\text{hold}} \cdot \exp\left(-\frac{d\left(\left[p^{\text{con}}_{t,e}\right]_1, p^{\text{act}}_{t,e}\right)^2}{\sigma^2}\right)\right) \cdot \mathbb{I}\left[\left[\mathcal{I}^{\text{con}}_{t,e}\right]_1 = \mathcal{I}^{\text{act}}_{t,e} = 1\right] \tag{2}$$

$$r^{\text{detach}}_{t,e} = \mathbb{I}\left[\left[\mathcal{I}^{\text{con}}_{t,e}\right]_1 = \mathcal{I}^{\text{act}}_{t,e} = 0 \wedge s > \delta\right] \tag{3}$$

where $d(\mathbf{a}, \mathbf{b})$ is the $L1$-norm between $\mathbf{a}$ and $\mathbf{b}$, $\epsilon_{\text{reach}} = \alpha_{\text{reach}} \cdot \mathbb{I}\left(\left[\mathcal{I}^{\text{con}}_{t,e}\right]_1 = \mathcal{I}^{\text{act}}_{t,e} = 0\right) + (1 - \alpha_{\text{reach}}) \cdot \mathbb{I}\left(\left[\mathcal{I}^{\text{con}}_{t,e}\right]_2 = \mathcal{I}^{\text{act}}_{t,e} = 1\right)$, α_{reach} is a linearly decreasing value from $1 \rightarrow 0$ when the reach phase begins until the contact is being made and $\mathbb{I}(\cdot)$ is an indicator function. This allows the robot to make graceful contact with the environment to avoid sudden impact between the current and the following contact sequence. The hold reward incentivises the robot to maintain contact, and it gets a higher reward for making contact at the desired location. The detach reward is a scalar reward that incentivises the agent to not make any contact during this phase.

For the case of manipulation, we additionally have a reward for tracking the object pose:

$$r^{pose}_{t,obj} = \frac{c_{pos}}{\epsilon_{pos} + \Delta p_{t,obj}} + \frac{c_{rot}}{\epsilon_{rot} + \Delta \theta_{t,obj}}$$

The total contact reward r^{con}_t is given as:

$$r^{\text{con}}_t = r^{\text{pose}}_{t,obj} + \sum_e \left(r^{\text{reach}}_{t,e} + r^{\text{hold}}_{t,e} + r^{\text{detach}}_{t,e}\right)$$

For further details about the observations and rewards used to train our contact-explicit policy, we refer the reader to the Appendix.

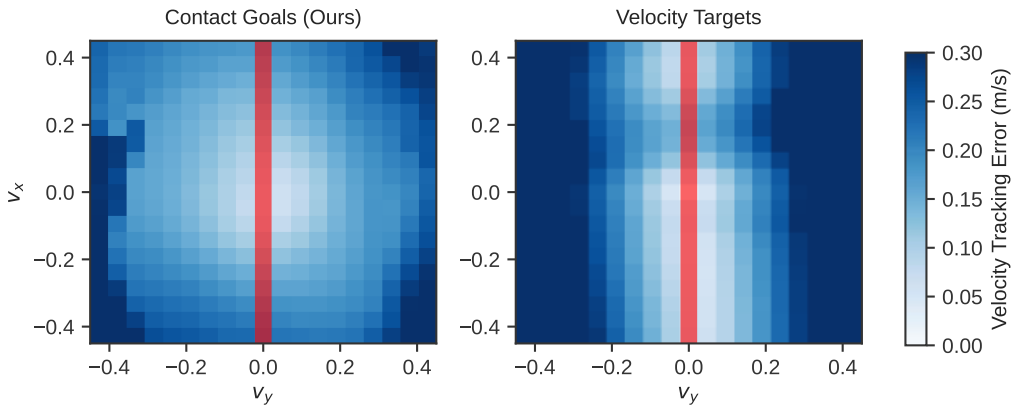

Figure 4: Comparison of velocity tracking error in all x-y directions. Each cell in the grid is a combination of x and y velocity. The red line denotes the velocity combinations seen during training. The results were averaged over 500 episodes (each lasting 15 seconds of simulation time).

4 Experiments and Results

We evaluate our contact-explicit framework for performing multiple tasks on various robotic embodiments: a quadruped, a humanoid and a robotic hand and further conduct extensive experiments on the legged robots for locomotion and bimanual manipulation. Our evaluations are based mainly on the multi-tasking and representation capabilities of our contact-explicit approach. The quadruped is trained to perform multiple gaits, such as *trot, pace, bound, jump and crawl*, as depicted in Fig. 1(a)-(e). The contact locations were sampled to move in all directions, with different stride lengths, stance widths (different for front and rear legs) and yaw rates, and the command durations were sampled from a narrow uniform distribution of $[0.34, 0.36]$ seconds. We use extensive domain randomisation to deploy our policy in the real world without additional fine-tuning. Similar strategy was also used to train the humanoid robot for various biped and quadruped locomotion modes. For bimanual manipulation, the humanoid is trained to perform the following bi-manual manipulation tasks as shown in Fig. 1 (f)-(g): 1) *Repose*: The robot must continuously maintain contact on two surfaces of a box (cuboid) and must track several positions and orientations for the object, sampled from a uniform distribution. These object poses are in the air, hence the humanoid must learn to lift the object and not let it slip from its hand while tracking the various poses. 2) *Reorient*: The robot must make and break contact with the box repeatedly to keep rotating it 45° on the table each time it makes contact. The contact locations were predefined so that the correct surface could be chosen to rotate the object continuously. For dexterous in-hand object manipulation, we used the same strategy as bimanual manipulation.

For both the embodiments, we use the Proximal Policy Optimization (PPO) [34] algorithm with a recurrent architecture (GRU) and entropy decay to train the policy in IsaacLab [35]. We use PPO as our algorithm of choice since it is effective in learning low-level motion primitives, as also observed in [5]. In the future, we'd also like to explore off-policy RL algorithms to make use of goal relabelling [36]. More details about our experimental setup can be found in the Appendix D, E.

Locomotion generalization to unseen velocities/directions. To demonstrate that contact-explicit representations truly generalize better to out-of-distribution scenarios and has better representation capabilities, we compare two policies with different goal representations, contact-explicit (ours) trained to only perform trotting gait against velocity targets (typical in learning quadrupedal locomotion as in [15, 6] and known to converge to a trotting gait), each trained only to move forward/backwards up to a maximum speed of 0.65 m/s (i.e., $v_x \in [-0.65, 0.65]$ m/s, $v_y = 0$). We evaluate the velocity tracking error of these two policies when commanded to move along all directions as shown in Fig. 4.

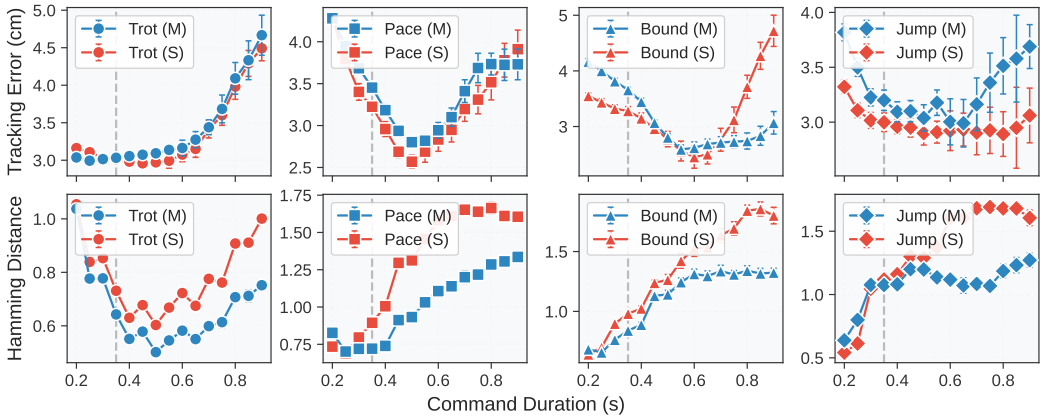

Figure 5: Comparison of contact location tracking ($L2$-norm) and contact plan deviation (Hamming distance) for the multi-gait policy (denoted by M) against single-gait policies (denoted by S) trained exclusively on one gait, evaluated over a broader range of command durations. Results are averaged over 1000 episodes, each lasting 15 seconds of simulation time. The dotted vertical grey line indicates the command duration seen during training.

Although the policy trained with velocity targets has slightly less tracking error while extrapolating velocity targets along the trained direction, it becomes apparent that the contact-explicit policy can cover a much broader range of velocities, compared to the one trained with velocity targets. Especially in the case of lateral/sideways walking ($v_x = 0$), the velocity-conditioned policy hesitates to move. In contrast, the contact-explicit policy moves sideways, even though it hadn't received any reward for the lateral movements during training, but was solely trained to track the contact goals.

Multi-task versus single-task. Our contact-explicit task representation enables learning multiple gaits in a single policy, which helps leverage the shared structure between different gaits to interpolate between them (even though it has not seen those states during training). To test our claims, we compare our multi-gait quadrupedal locomotion policy against separate policies trained on single gaits, as shown in Fig. 5. The policies were trained with command durations sampled from a narrow uniform distribution of range $[0.34s, 0.36s]$ and evaluated over a broader range of $[0.2s, 0.9s]$. We use two metrics: contact location tracking error measured using $L2$-norm between the actual end-effector locations and the planned contact locations while making contact, and the contact plan deviation measured using the Hamming distance between the desired contact plan and the actual contact status of the end-effectors.

From Fig. 5, we observe that the multi-gait (M) policy has the lowest contact plan deviation across all the evaluated command durations and for all gaits. We hypothesize that this is mainly due to our contact-explicit representation that enables learning multiple gaits in a single policy. We also observe that generally, the tracking error of both the single-gait and multi-gait policies are similar, i.e. within 2 cm of difference.

Manipulation generalization to unseen object shapes. We compare our bimanual manipulation policy against a policy that instead uses one-hot task encoding to distinguish the tasks. Here, we evaluate the performance of the two policies, one with contact goals in the state (ours, blue) and another that uses a one-hot task encoding (baseline, red) instead to distinguish the tasks. For the baseline, we additionally provide the object dimensions as an observation to the policy. The policy was only trained on cuboidal shapes and evaluated here with cylindrical and spherical shapes. Our results are summarised in Fig. 6.

As with the contact-explicit locomotion policy being able to generalize to unseen contact locations (velocity directions), we witness with the contact-explicit bimanual manipulation policy that we can better generalize to unseen object shapes. Using the contact-explicit approach, the policy learns to track the contact locations on the different shapes much better than other implicit goal representa-

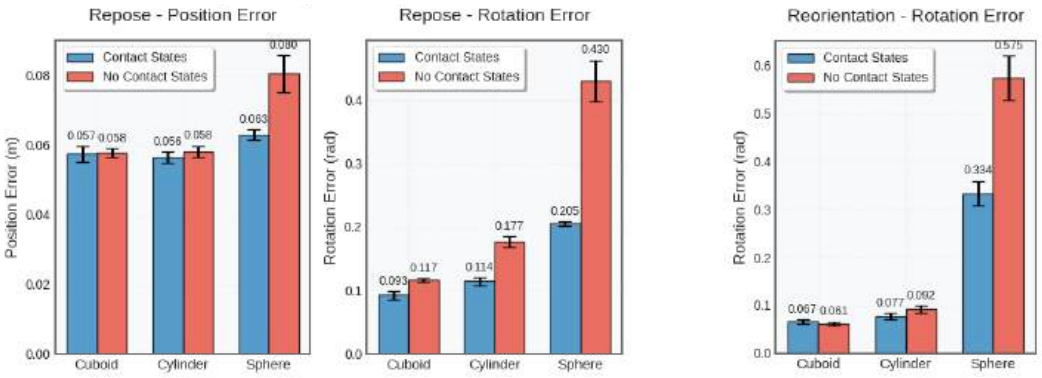

Figure 6: Tracking error comparison on unseen object shapes across various tasks. We compare our contact-explicit policy (uses contact states) with a baseline policy that uses one-hot task encoding (no contact states) to represent the task. Results are averaged over 1000 episodes, each lasting 20 seconds of simulated time.

tions. Especially in the case of spherical shapes for object reorientation on the table, we observe that our policy comes up with emergent retrying behavior to track the object poses as the object starts rolling on the table.

Manipulation generalization to unseen object poses. We compare the two bimanual manipulation policies mentioned in the previous experiment, contact-explicit (ours) and one-hot task encoding (baseline), to track object poses that were outside the training distribution. This was done by extrapolating the range of object poses from those seen in the training distribution. The results, summarized in Table 1, demonstrate that contact-explicit policy consistently outperforms the one-hot task policy, with lower variability indicating more stable task accomplishment. This underscores the value of contact-aware policies for robust bimanual manipulation in uncertain scenarios.

Task / Repose	**Contact-Explicit**	**One-hot task**
Position Error (m)	**0.115** $\pm$ 0.003	0.129 $\pm$ 0.003
Rotation Error (rad)	**0.390** $\pm$ 0.006	0.455 $\pm$ 0.007
Task / Reorientation	**Contact-Explicit**	**One-hot task**
Rotation Error (rad)	**0.109** $\pm$ 0.002	0.191 $\pm$ 0.004

Table 1: Experimental evaluation of our contact-explicit policy against a baseline that uses one-hot task encoding for out-of-distribution object poses. The evaluation details are in the Appendix F.1.

5 Conclusion

We presented a unified contact-explicit task representation for learning a wide range of locomotion and manipulation skills through reinforcement learning. By treating contact as a central physical primitive—rather than a byproduct of motion optimization—our approach enables a single policy to generalize across morphologically diverse platforms and tasks. Empirical results demonstrate that contact-conditioned policies offer stronger generalization to out-of-distribution goal configurations.

Moving forward, we aim to extend this framework to hierarchical reinforcement learning by coupling our low-level contact-conditioned policy with a learned high-level planner. This would allow autonomous long-horizon loco-manipulation in complex environments. We also plan to explore prehensile interactions and improve sim-to-real robustness for real-world deployment.

6 Limitations

A key limitation of our current approach lies in the use of a manually designed high-level contact planner. While sufficient for proof-of-concept validation, this restricts autonomy and task scalability. We plan to address this by training a high-level policy that generates contact plans in tandem with the low-level policy, enabling end-to-end learning of long-horizon behaviours, such that each level of hierarchy can serve as a curriculum for the other. Additionally, since we can train a reliable low-level control policy to track contacts with our current approach, we can freeze it and train a contact planner to learn in the search space of contacts.

Our framework is currently trained and evaluated using a rigid contact model using the GPU-accelerated simulation engine (PhyX). While this allows efficient training at scale, it limits applicability to environments involving deformable objects or soft contacts. Therefore, we are yet to test our framework in such settings, although our contact representation remains unchanged. Future work could explore extending our framework to soft-body interactions by incorporating fuzzy contact representations, leveraging learned deformation models, or applying sim-to-real transfer techniques to adapt to compliant dynamics. Additionally, we currently focus only on non-prehensile manipulation. Although we believe the framework can be extended to prehensile settings with minimal modifications (e.g., introducing a modality flag), this remains untested.

Finally, our reliance on accurate state estimation—especially for object pose during manipulation—may limit real-world applicability. For deployment of the locomotion policy in the real world, we relied on privileged sensing (e.g., Vicon motion capture), but we intend to overcome this via teacher-student distillation or vision-based estimation in future work.

References

[1] D. Hoeller, N. Rudin, D. Sako, and M. Hutter. Anymal parkour: Learning agile navigation for quadrupedal robots, 2023. URL https://arxiv.org/abs/2306.14874.

[2] X. Cheng, K. Shi, A. Agarwal, and D. Pathak. Extreme parkour with legged robots. *arXiv preprint arXiv:2309.14341*, 2023.

[3] OpenAI, I. Akkaya, M. Andrychowicz, M. Chociej, M. Litwin, B. McGrew, A. Petron, A. Paino, M. Plappert, G. Powell, R. Ribas, J. Schneider, N. Tezak, J. Tworek, P. Welinder, L. Weng, Q. Yuan, W. Zaremba, and L. Zhang. Solving rubik's cube with a robot hand, 2019. URL https://arxiv.org/abs/1910.07113.

[4] R. Singh, A. Allshire, A. Handa, N. Ratliff, and K. V. Wyk. Dextrah-rgb: Visuomotor policies to grasp anything with dexterous hands, 2025. URL https://arxiv.org/abs/2412.01791.

[5] Z.-H. Yin, C. Wang, L. Pineda, F. Hogan, K. Bodduluri, A. Sharma, P. Lancaster, I. Prasad, M. Kalakrishnan, J. Malik, M. Lambeta, T. Wu, P. Abbeel, and M. Mukadam. Dexteritygen: Foundation controller for unprecedented dexterity, 2025. URL https://arxiv.org/abs/2502.04307.

[6] N. Rudin, D. Hoeller, P. Reist, and M. Hutter. Learning to walk in minutes using massively parallel deep reinforcement learning, 2022. URL https://arxiv.org/abs/2109.11978.

[7] 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*, 7(62):eabk2822, 2022. doi:10.1126/scirobotics.abk2822. URL https://www.science.org/doi/abs/10.1126/scirobotics.abk2822.

[8] C. Zhang, N. Rudin, D. Hoeller, and M. Hutter. Learning agile locomotion on risky terrains, 2024. URL https://arxiv.org/abs/2311.10484.

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

[10] T. He, W. Xiao, T. Lin, Z. Luo, Z. Xu, Z. Jiang, J. Kautz, C. Liu, G. Shi, X. Wang, et al. Hover: Versatile neural whole-body controller for humanoid robots. *arXiv preprint arXiv:2410.21229*, 2024.

[11] C. Zhang, W. Xiao, T. He, and G. Shi. Wococo: Learning whole-body humanoid control with sequential contacts, 2024.

[12] M. Ciebielski and M. Khadiv. Contact-conditioned learning of locomotion policies. *arXiv preprint arXiv:2408.00776*, 2024.

[13] T. Lin, K. Sachdev, L. Fan, J. Malik, and Y. Zhu. Sim-to-real reinforcement learning for vision-based dexterous manipulation on humanoids. *arXiv:2502.20396*, 2025.

[14] X. B. Peng, G. Berseth, K. Yin, and M. Van De Panne. Deeploco: Dynamic locomotion skills using hierarchical deep reinforcement learning. *ACM Trans. Graph.*, 36(4):41:1–41:13, July 2017. ISSN 0730-0301. doi:10.1145/3072959.3073602. URL http://doi.acm.org/10.1145/3072959.3073602.

[15] J. Hwangbo, J. Lee, A. Dosovitskiy, D. Bellicoso, V. Tsounis, V. Koltun, and M. Hutter. Learning agile and dynamic motor skills for legged robots. *Science Robotics*, 4(26):eaau5872, 2019.

[16] N. Rudin, D. Hoeller, M. Bjelonic, and M. Hutter. Advanced skills by learning locomotion and local navigation end-to-end, 2022. URL https://arxiv.org/abs/2209.12827.

[17] G. B. Margolis and P. Agrawal. Walk these ways: Tuning robot control for generalization with multiplicity of behavior. *Conference on Robot Learning*, 2022.

[18] G. Bellegarda, M. Shafiee, and A. Ijspeert. Allgaits: Learning all quadruped gaits and transitions. *arXiv preprint arXiv:2411.04787*, 2024.

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

[20] Z. Li, X. B. Peng, P. Abbeel, S. Levine, G. Berseth, and K. Sreenath. Reinforcement learning for versatile, dynamic, and robust bipedal locomotion control. *The International Journal of Robotics Research*, page 02783649241285161, 2024.

[21] F. Zargarbashi, J. Cheng, D. Kang, R. Sumner, and S. Coros. Robotkeyframing: Learning locomotion with high-level objectives via mixture of dense and sparse rewards, 2024. URL https://arxiv.org/abs/2407.11562.

[22] 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.

[23] Y. Lin, A. Church, M. Yang, H. Li, J. Lloyd, D. Zhang, and N. F. Lepora. Bi-touch: Bimanual tactile manipulation with sim-to-real deep reinforcement learning. *IEEE Robotics and Automation Letters*, 8(9):5472–5479, 2023.

[24] T. G. W. Lum, M. Matak, V. Makoviychuk, A. Handa, A. Allshire, T. Hermans, N. D. Ratliff, and K. V. Wyk. Dextrah-g: Pixels-to-action dexterous arm-hand grasping with geometric fabrics, 2024. URL https://arxiv.org/abs/2407.02274.

[25] Y. Chen, T. Wu, S. Wang, X. Feng, J. Jiang, Z. Lu, S. McAleer, H. Dong, S.-C. Zhu, and Y. Yang. Towards human-level bimanual dexterous manipulation with reinforcement learning. *Advances in Neural Information Processing Systems*, 35:5150–5163, 2022.

[26] G. Pan, Q. Ben, Z. Yuan, G. Jiang, Y. Ji, S. Li, J. Pang, H. Liu, and H. Xu. Roboduet: Learning a cooperative policy for whole-body legged loco-manipulation. *IEEE Robotics and Automation Letters*, 2025.

[27] Z. Fu, X. Cheng, and D. Pathak. Deep whole-body control: Learning a unified policy for manipulation and locomotion, 2022. URL https://arxiv.org/abs/2210.10044.

[28] J. Dao, H. Duan, and A. Fern. Sim-to-real learning for humanoid box loco-manipulation. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 16930–16936. IEEE, 2024.

[29] M. Liu, Z. Chen, X. Cheng, Y. Ji, R.-Z. Qiu, R. Yang, and X. Wang. Visual whole-body control for legged loco-manipulation. *arXiv preprint arXiv:2403.16967*, 2024.

[30] R.-Z. Qiu, Y. Song, X. Peng, S. A. Suryadevara, G. Yang, M. Liu, M. Ji, C. Jia, R. Yang, X. Zou, et al. Wildlma: Long horizon loco-manipulation in the wild. *arXiv preprint arXiv:2411.15131*, 2024.

[31] S. Omar, L. Amatucci, G. Turrisi, V. Barasuol, and C. Semini. Safesteps: Learning safer footstep planning policies for legged robots via model-based priors. In *IEEE-RAS International Conference on Humanoid Robots*, 2023.

[32] V. Dhedin, A. K. C. Ravi, A. Jordana, H. Zhu, A. Meduri, L. Righetti, B. Schölkopf, M. Khadiv, et al. Diffusion-based learning of contact plans for agile locomotion. In *2024 IEEE-RAS 23rd International Conference on Humanoid Robots (Humanoids)*, pages 637–644. IEEE, 2024.

[33] I. Taouil, H. Zhao, A. Dai, and M. Khadiv. Physically consistent humanoid loco-manipulation using latent diffusion models, 2025. URL https://arxiv.org/abs/2504.16843.

[34] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal policy optimization algorithms, 2017. URL https://arxiv.org/abs/1707.06347.

[35] M. Mittal, C. Yu, Q. Yu, J. Liu, N. Rudin, D. Hoeller, J. L. Yuan, R. Singh, Y. Guo, H. Mazhar, A. Mandlekar, B. Babich, G. State, M. Hutter, and A. Garg. Orbit: A unified simulation framework for interactive robot learning environments. *IEEE Robotics and Automation Letters*, 8(6):3740–3747, 2023. doi:10.1109/LRA.2023.3270034.

[36] M. Andrychowicz, F. Wolski, A. Ray, J. Schneider, R. Fong, P. Welinder, B. McGrew, J. Tobin, P. Abbeel, and W. Zaremba. Hindsight experience replay, 2018. URL https://arxiv.org/abs/1707.01495.

A Sim-to-Real

Quadruped Locomotion. Table 2 provides physical quantities that were randomised during training, and Table 3 provides observation noises used for effective sim2real transfer.

Term	Value
Ground Friction	$\mathcal{U}(0.1, 3.0)$
Ground Restitution	$\mathcal{U}(0.0, 0.5)$
P Gain	$\mathcal{U}(20, 30)$
D Gain	$\mathcal{U}(0.4, 0.6)$
Joint Offsets	$\mathcal{U}(0.5, 1.5) *$ default rad
Link Mass	$\mathcal{U}(0.85, 1.15) *$ default kg
Base Mass	$\mathcal{U}(-5.0, 5.0) *$ default kg
Control Delay	$\mathcal{U}(0, 8)$ms
Impulse Pushes	interval = $5s$, $\Delta_{xy} \sim \mathcal{U}(-0.5, 0.5)$

Table 2: Randomisation of Physical Parameters for sim2real transfer

Observation	Noise
Base Linear Velocity	$\mathcal{U}(-0.1, 0.1)$
Base Angular Velocity	$\mathcal{U}(-0.2, 0.2)$
Projected Gravity	$\mathcal{U}(-0.05, 0.05)$
Relative Feet Location to Goal	$\mathcal{U}(-0.03, 0.03)$
Contact Remaining Time	$\mathcal{U}(-0.05, 0.05)$
Joint Positions	$\mathcal{U}(-0.03, 0.03)$
Joint Velocities	$\mathcal{U}(-0.05, 0.05)$

Table 3: Observation Noises for sim2real transfer

The policy updates the actions at 50Hz, whereas the controller using a PD structure computes the joint motor torques at 200Hz. We also apply a simple exponential moving average filter ($\alpha = 0.3$) to the raw joint position targets given by the policy to smoothen the motions.

B Observations

Quadruped and Humanoid Locomotion. The proprioceptive observations of the contact-explicit locomotion policy include the robot's base linear velocity, base angular velocity, projected gravity, joint positions, joint velocities and previous actions. The additional task observations from the contact planner include the current and the next contact sequence of all feet, the current and next contact locations of all the feet in the base frame, and the current command duration, as described in 3.2, and the relative distance of the robot's feet to its desired contact locations.

Notably, we do not use the contact sensing in the robot's feet, so it is not an observation for the policy. In terms of simulation performance, we didn't find much difference between the two and chose to leave it out due to a lack of reliable contact sensing in the real world. Even though the Unitree Go2 has contact sensors on the feet, we didn't find it to be reliable enough for our use case.

Bimanual and Dextrous Manipulation. The proprioceptive observations of the contact-explicit bimanual manipulation policy include the robot's joint positions, joint velocities, previous actions and the object's position in the base frame, orientation, linear and angular velocities. The additional task observations include the current and next contact sequence of the robot's hands, the current and next contact locations of the robot's hands on the object's surface in the base frame, the current command's duration, and the goal pose relative to the object.

C Rewards

Apart from the rewards mentioned in Sec. 3.2 to achieve the different contact goals for the different phases, we provide additional rewards for obtaining smoother and desirable motions, most of them typically used in these locomotion/manipulation settings, as shown in Table 4 for quadrupedal locomotion and Table 5 for humanoids bumanual manipulation.

Reward	**Expression**	**Weight**
Reach	refer to Sec. 3.2	3.5
Hold	refer to Sec. 3.2	2.0
Detach	refer to Sec. 3.2	2.0
Heading	$\exp\left(-\frac{1}{\sigma^2}(\theta_z - \theta_z^*)^2\right)$	0.25
Joint Torques	$\sum_{joint} \lvert\tau\rvert^2$	$-1e^{-4}$
Joint Deviation	$\sum_{joint} \lvert q - q_{\text{default}}\rvert^2$	-0.2
Joint Acceleration	$\sum_{joint} \lvert\ddot{q}\rvert^2$	$-1.5e^{-7}$
Foot Slide	$\sum_{foot} \left\lvert v_{xy}^{foot}\right\rvert^2 \cdot \mathbb{I}(\text{contact})$	-0.5
Base Z-Velocity	$\left\lvert v_z^{base}\right\rvert^2$	-1.0
Flat Base Orientation	$\left\lvert w_{xy}^{base}\right\rvert^2$	-0.25
Angular Velocity XY	$\left\lvert \dot{w}_{xy}^{base}\right\rvert^2$	-0.025
Action Rate	$\sum_{joint} \lvert\dot{a}\rvert^2$	$-7.5e^{-3}$
Contact Forces	$\sum_{foot} \lvert F_c\rvert - 200$	$-1.5e^{-3}$
Goal Completion Bonus	$\mathbb{I}(\text{goal completed})$	10.0
Termination	$\mathbb{I}(\text{termination})$	-200

Table 4: Full Reward for Contact-Explicit Multi-Gait Locomotion

Reward	**Expression**	**Weight**
Reach	refer to Sec. 3.2	20
Hold	refer to Sec. 3.2	1.5
Detach	refer to Sec. 3.2	1.5
Object Pose Tracking	refer to Sec. 3.2	0.5
Joint Velocity	$\sum_{joint} \lvert\dot{q}\rvert^2$	$-5e^{-5}$
Joint Acceleration	$\sum_{joint} \lvert\ddot{q}\rvert^2$	$-1e^{-8}$
Joint Torques	$\sum_{joint} \lvert\tau\rvert^2$	$-3e^{-7}$
Joint Deviation	$\sum_{joint} \lvert q - q_{\text{default}}\rvert^2$	-0.01
Hand Accelerations	$\sum_{hand} \ddot{x}$	$-5e^{-4}$
Action Rate	$\sum_{joint} \lvert\dot{a}\rvert^2$	$-5e^{-5}$
Contact Forces	$\sum_{hand} \lvert F_c\rvert - 120$	$-5e^{-4}$
Goal Completion Bonus	$\mathbb{I}(\text{goal completed})$	200.0
Termination	$\mathbb{I}(\text{termination})$	-150

Table 5: Full Reward for Contact-Explicit Multi-Task Bimanual Manipulation

Here, θ_z^* is the direction in the line along the XY-plane connecting the average desired feet location of the hind legs and the average desired feet location of the front legs, τ is the joint torques, q is the current joint positions, q_{default} is the default joint positions, $\mathbf{v}$ is the robot's linear velocity in the base frame, $\mathbf{w}$ is the robot's angular velocity, a is the policy action and F_c is the contact force and $\mathbb{I}$ is an indicator function which returns 1 if the condition in the function argument is True.

Figure 7: Snapshot of the training terrain of Unitree Go2 in simulation (IsaacLab).

D Environment-Specific Details

Quadruped Locomotion. During reset, each environment samples stride lengths and stance widths for each leg from a nominal foot position from $\mathcal{U}(0.0m, 0.45m)$ and $\mathcal{U}(0.1m, 0.3m)$ respectively, a gait (trot, pace, bound, jump, crawl), and the command duration $\mathcal{U}(0.34s, 0.36s)$. Since the gaits trot, pace, bound and jump only have two contact sequences that are repeated to execute them, this corresponds to a maximum velocity of $v_{\text{max}} = 0.45/(0.35 * 2) \approx 0.65m/s$, whereas the crawl gait has four sequences which correspond to a maximum velocity of $v_{\text{max}} = 0.45/(0.35 * 4) \approx 0.32m/s$. The stride length is used to compute the desired contact locations along a random heading from $[-\pi, \pi]rad$ or a random yaw rate $[-\pi/2, \pi/2]rad/2$ for the entire episode and fixed in the world frame. We then update the goals to go further one by one, based on the goal passing conditions described in Appendix E. During training, we also sample a new gait in the middle of an episode. The reach phase begins if the time remaining to contact, $s < 0.2s$.

The quadruped was trained on slightly rough terrain as shown in Fig. 7. Qualitatively, this allowed the robot to discover lifting the leg better from the ground without any additional rewards for it.

Bimanual Manipulation. During reset, we choose the robot's hands to be close to the object to aid exploration. The contact locations are chosen at the centre of a desired contact surface. During training, we only trained on cuboidal objects of sizes along each dimension sampled from a uniform distribution $\mathcal{U}[0.1, 0.3]m$ and masses sampled from the uniform distribution $\mathcal{U}[0.1, 1]kg$. The command duration is sampled from the uniform distribution $\mathcal{U}[1.0, 1.5]s$. The reach phase begins if the time remaining to contact, $s < 0.5s$.

The Unitree G1 URDF we use for our bimanual manipulation policy is updated to use simple spherical hands of radius $0.035m$ for performing the repose and reorientation tasks.

We also notice that the goal completion bonus reward for the bimanual manipulation tasks is higher than for locomotion. This is because the commands are sampled more frequently in the case of locomotion.

For the repose task, the robot has to maintain contact with the same surfaces of the object throughout the entire episode while tracking different object poses sampled from the ranges given in Table 6. For the reorientation task, we only move the object by 45° each time it makes contact with it, to keep rotating it. Each time it makes contact, we choose a different surface and use the centre of the surface as the desired contact location.

E Goal Update Conditions

We only update the goals when certain conditions are met as described below for each embodiment. We find that these simple conditions help progress in the task. Providing the goal completion bonus also helps in not getting stuck in local minima, and incentivises it to keep exploring newer goals.

Quadruped Locomotion. For training the locomotion policy, we update the contact goals (location and sequence) only if the XY coordinates of the quadruped's base, p^{xy}_{base} lie within the quadrilateral formed by the desired contact location's XY coordinates, $\left[p^{\text{con}}_{t,e}\right]^{xy}_{1}$.

Bimanual Manipulation. We update the goals for the bimanual manipulation experiments when the rotation error between the object target pose and current pose is less than $0.3\ rad$. This simple condition enables the robot to keep trying until it achieves the goal before moving onto the next one. These trying behaviours could probably have led to the emergent retrying behaviours when the object slips from the robot's hands.

Term	**Value**
X-Position	$\mathcal{U}(0.0, 0.1)$
Y-Position	$\mathcal{U}(-0.15, 0.15)$
Z-Position	$\mathcal{U}(0.05, 0.25)$
Roll	$\mathcal{U}(-0.6, 0.6)$
Pitch	$\mathcal{U}(-0.6, 0.6)$
Yaw	$\mathcal{U}(-0.6, 0.6)$

Table 6: Object Pose Target ranges relative to the robot base frame for Repose Task.

F Evaluation Details

F.1 Manipulation generalization to unseen object poses

The ranges of object poses for repose task used for this evaluation are given in Table 7. The object poses are relative to the robot's base position, whose z-height is close to the table height.

Term	**Value**
X-Position	$\mathcal{U}(0.0, 0.2)$
Y-Position	$\mathcal{U}(-0.3, 0.3)$
Z-Position	$\mathcal{U}(0.05, 0.4)$
Roll	$\mathcal{U}(-1.2, 1.2)$
Pitch	$\mathcal{U}(-1.2, 1.2)$
Yaw	$\mathcal{U}(-1.2, 1.2)$

Table 7: Object Pose Target ranges for Repose Task used for the evaluation of manipulation generalization to unseen object poses.

For the reorientation task, the object positions on the table were sampled along $x \sim \mathcal{U}(0.0, 0.2)$ amd $y \sim \mathcal{U}(-0.25, 0.25)$.

G PPO Hyperparameters

We used the same set of hyperparameters for PPO for both the environments as provided in Table 8.

Hyperparameter	**Value**
max iterations	10000
# environments	8192
# steps per rollout	24
# epochs per rollout	5
# mini batches	4
value loss coefficient	1.0
clip range	0.2
learning rate	$1e^{-3}$
learning rate schedule	Adaptive
discount factor (γ)	0.998
GAE lambda (λ)	0.95
actor and critic dimensions	[512, 256, 128]
activation	ELU
initial entropy bonus	0.01
final entropy bonus	0.001
entropy decay steps	5000
entropy decay type	Exponential
RNN type	GRU
RNN hidden size	256
RNN # layers	3

Table 8: PPO Hyperparameters used for Quadruped and Humanoid Experiments