# Supplementary Material for *Curiosity-Driven Learning of Joint Locomotion and Manipulation Tasks*

**Anonymous Author(s)**
Affiliation
Address
email

In the following, we provide the implementation details of the simulation and real-world experiments presented in the main manuscript.

## 1 Simulation Setup

We train with NVIDIA's Isaac Gym [1] and employ Proximal Policy Optimization (PPO) [2]. A detailed description of the used training pipeline can be found in [3]. A full training run comprises 2000 policy updates to ensure reward convergence for all investigated tasks. It takes one hour to train a policy on a single NVIDIA RTX 2080 Ti graphics card. Subsequently, we give a detailed description of the training environment.

- **Reward Formulation:** The definitions and weights of the reward terms used for the door and the package task are detailed in Table 1. We decided to add two task-related shaping rewards for the task of package manipulation to improve the behavior for real-world tests. Namely, the agent receives penalties for generating high package velocities and exerting large contact forces onto the table. Notice that this choice is not violating the idea of the proposed approach. Firstly, the added penalties are unrelated to the main task, which is still defined by a single sparse reward. Secondly, our approach first generates unbiased behaviors and can then be augmented for more pleasing results. In contrast, other formulations bias the agent as a byproduct of defining the desired task in a dense fashion. Penalizing table contacts and the package velocity, which is part of the chosen curiosity state, clearly increases the difficulty of discovering the desired skill. To compensate for this, we employ a simple reward scaling scheme. The first 1000 training iterations serve as a discovery phase, as most runs discover the sparse reward in that time. Shaping and standing rewards are active but scaled by a factor of 0.1. The second half of training acts as a shaping phase where the scaling factor is gradually increased to 1 over the course of 500 iterations.
- **Observations:** The corresponding observation definitions can be found in Table 2. All observations are subject to noise to account for uncertainties and sensor noise in reality. For more detail in that regard, please refer to [3].
- **Randomization:** To improve generalization to different environments, as well as robustness against mismatches between simulation and reality, masses and friction coefficients are randomized as detailed in Table 3. Additionally, the robot spawns in a randomized pose, i.e., initial position, orientation, and joint configuration vary. All randomized properties are sampled from a uniform distribution in the interval of $[\mu - \frac{\epsilon}{2}, \mu + \frac{\epsilon}{2}]$ for every training environment.

Submitted to the 7th Conference on Robot Learning (CoRL 2023). Do not distribute.

Table 1: Rewards

| Name | Formula | Weight |
|---|---|---|
| Intrinsic Reward | | |
| Random Network Distillation (RND) prediction error | $\left\lVert f(\boldsymbol{s}_c) - \hat{f}(\boldsymbol{s}_c) \right\rVert_2$ | 200 |
| Task Rewards | | |
| Door opened | $\begin{cases} 1, & \text{if } \boldsymbol{q}_{\text{hinge}} > 1.5 \\ 0, & \text{otherwise} \end{cases}$ | 1.0 |
| Package delivered | $\begin{cases} 1, & \text{if } {}_{\mathcal{I}}\boldsymbol{r}_{\text{package}} \in \mathcal{S}_{\text{bin}} \\ 0, & \text{otherwise} \end{cases}$ | 1.0 |
| Standing Rewards | | |
| Height | ${}_{\mathcal{I}}z_{\text{base}}$ | 0.5 |
| Upright base | $\frac{\pi/2 - \arccos({}_{\mathcal{I}}\boldsymbol{e}_x^{\mathcal{B}} \cdot {}_{\mathcal{I}}\boldsymbol{e}_z^{\mathcal{I}})}{\pi/2}$ | 0.5 |
| Straight shoulder joints | $-\lVert \boldsymbol{q}_{\text{shoulders}} \rVert^2$ | 0.5 |
| Straight knee joints | $\exp(-\lVert \boldsymbol{q}_{\text{knees}} \rVert^2)$ | 0.25 |
| Shaping Rewards | | |
| Joint torque | $-\lVert \boldsymbol{\tau} \rVert^2$ | $1.5 \cdot 10^{-5}$ |
| Joint acceleration | $-\lVert \ddot{\boldsymbol{q}} \rVert^2$ | $2.5 \cdot 10^{-7}$ |
| Joint velocity | $-\lVert \dot{\boldsymbol{q}} \rVert^2$ | $2.5 \cdot 10^{-4}$ |
| Table contact force | $-\lVert \boldsymbol{F}_{\text{c, table}} \rVert^2$ | $1.0 \cdot 10^{-5}$ |
| Package velocity | $-\lVert {}_{\mathcal{I}}\dot{\boldsymbol{r}}_{\text{package}} \rVert^2$ | $1.0 \cdot 10^{-2}$ |

Table 2: Observations

| Robot-related Observations | |
|---|---|
| ${}_{\mathcal{B}}\dot{\boldsymbol{r}}_{\text{base}} \in \mathbb{R}^3$ | Linear base velocity |
| ${}_{\mathcal{B}}\boldsymbol{\omega}_{\text{base}} \in \mathbb{R}^3$ | Angular base velocity |
| ${}_{\mathcal{B}}\boldsymbol{g} \in \mathbb{R}^3$ | Projected gravity vector |
| $\boldsymbol{q}_{\text{legs}} \in \mathbb{R}^{12}$ | Joint configuration without wheels |
| $\boldsymbol{o}_{\text{hooks}} \in \mathbb{R}^4$ | Hook directions (for pull doors) |
| $\dot{\boldsymbol{q}} \in \mathbb{R}^{16}$ | Joint velocity |
| $\boldsymbol{a}_{\text{prev}} \in \mathbb{R}^{16}$ | Previous actions |
| Door-related Observations | |
| ${}_{\mathcal{C}}\boldsymbol{r}_{CH} \in \mathbb{R}^3$ | Relative door handle position |
| ${}_{\mathcal{C}}\boldsymbol{r}_{CH_{\text{init}}} \in \mathbb{R}^3$ | Relative initial door handle position |
| Package-related Observations | |
| ${}_{\mathcal{C}}\boldsymbol{r}_{CP} \in \mathbb{R}^3$ | Relative package position |
| ${}_{\mathcal{C}}\boldsymbol{r}_{CT} \in \mathbb{R}^3$ | Relative table position |
| ${}_{\mathcal{C}}\boldsymbol{r}_{CB} \in \mathbb{R}^3$ | Relative bin position |

Table 3: Randomization Parameters

| Uniformly Randomized Property | Mean $\mu$ | Range $\epsilon$ | Unit |
|---|---|---|---|
| Global friction coefficient | $0.75$ | $0.75$ | - |
| Robot position $(x, y)$ | $0$ | $0.6$ | $\mathrm{m}$ |
| Initial robot yaw angle | $0$ | $1$ | $\mathrm{rad}$ |
| Initial joint angle deviation | $0$ | $1$ | $\mathrm{rad}$ |
| Added robot mass | $0$ | $10$ | $\mathrm{kg}$ |
| Package mass | $1.375$ | $1.0$ | $\mathrm{kg}$ |
| Door torque offset $\boldsymbol{\tau}_{\text{const}}$ | $[10 \quad 0]^\top$ | $[10 \quad 0]^\top$ | $\mathrm{N\,m}$ |
| Door spring coefficient $\boldsymbol{k}$ | $[0 \quad 5]^\top$ | $[0 \quad 5]^\top$ | $\frac{\mathrm{N\,m}}{\mathrm{rad}}$ |
| Door damping coefficient $\boldsymbol{d}$ | $[25 \quad 1]^\top$ | $[25 \quad 1]^\top$ | $\frac{\mathrm{N\,m\,s}}{\mathrm{rad}}$ |

- **Termination Conditions:** Episodes terminate after 8 seconds, resetting the environments to their initial state. An episode terminates early if either the robot is in collision, or if the robot's center is too low, i.e., if the robot does not manage to stand and falls. The second condition accelerates training but is not necessary for successful learning. We also terminate an episode if the package is not in contact with either the table or the front wheels to prevent the agent from directly throwing the package. This termination condition is disabled in close proximity to the bin to allow the dropping of the package into the bin.

- **Door Model:** The considered doors feature standard lever door handles that need to be pressed to a certain degree to unlock the door. In simulation, the handle needs to be pressed once to keep the door unlocked for the rest of the episode. Dynamics of the hinge and handle are modeled as spring-damper systems with a constant torque offset $\boldsymbol{\tau}_{\text{door}}$. This is achieved by applying the torque

$$\boldsymbol{\tau}_{\text{door}} = \boldsymbol{\tau}_{\text{const}} + \operatorname{diag}(\boldsymbol{k}) \cdot \boldsymbol{q}_{\text{door}} + \operatorname{diag}(\boldsymbol{d}) \cdot \dot{\boldsymbol{q}}_{\text{door}}, \tag{1}$$

  to the door joints. Constants $\boldsymbol{\tau}_{\text{door}}$, $\boldsymbol{k}$, and $\boldsymbol{d}$ are randomized by sampling from a uniform distribution. Measurements on the lab door provide reference values for realistic door dynamics. Further details are provided in Table 3.

- **Field of View Simulation:** To mimic the perception system of the real robot we simulate the Field of View (FOV) for egocentric vision, as introduced in simulation experiments in [4], resulting in behaviors that actively direct the robot's gaze. A visual marker, further explained in section 2, specifies the position of the door handle. Consequently, the observation ${}_{C}\boldsymbol{r}_{CH}$ is only available if the marker is detected by a camera. Always passing the door handle observation in the simulation would therefore not capture the real system behavior. Instead, the observation is set to $\mathbf{0}$ if the visual marker leaves the camera's FOV. This way, the agent learns to approximately partition the observation space and reason about when it is necessary to see the visual marker. The agent can develop behaviors to mitigate a lost observation and to actively keep the marker in the FOV. An illustration of the approach is provided in Fig. 1. Note that the second door-related observation ${}_{C}\boldsymbol{r}_{CH_{\text{init}}}$ is not set to $\mathbf{0}$ because the initial door handle position is static with respect to the inertial frame. The observation can thus be bootstrapped with the onboard localization of the robot even if the visual marker leaves the FOV.

## 2 Real-World Setup

We utilize AprilTags [5] to obtain task-related observations in the real world. The AprilTag system features a vision-based algorithm that determines the relative position and orientation of detected tags. Two visual markers attached to the door provide the relative door handle position observation ${}_C\boldsymbol{r}_{CH}$. If the robot does not detect the tags, the observation is set to $\mathbf{0}$ to achieve the same behavior as in simulation. The initial door handle position observation ${}_C\boldsymbol{r}_{CH_{\text{init}}}$ is determined by two markers attached to the door frame. We make use of the robot's onboard localization to obtain an observation even if the tags leave the FOV of the camera. AprilTags also provide relative positions of the package, bin, and table. We do not make use of the proposed FOV simulation for the package manipulation task for two reasons. Firstly, it increases the difficulty of learning the desired behavior because the robot tries to keep the package in the FOV by leaning over the bin and falling. Secondly, the package is kept in the FOV naturally until the package is dropped, rendering the additional FOV constraint unnecessary for this task.

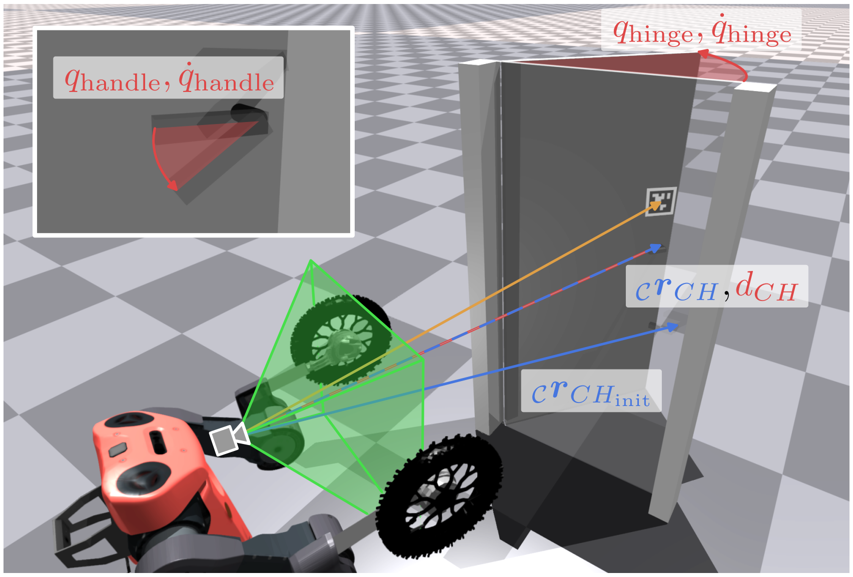
$q_{\text{handle}}, \dot{q}_{\text{handle}}$
$q_{\text{hinge}}, \dot{q}_{\text{hinge}}$
${}_C\boldsymbol{r}_{CH}, d_{CH}$
${}_C\boldsymbol{r}_{CH_{\text{init}}}$

Figure 1: Door setup and FOV simulation. Components of the curiosity state $\boldsymbol{s}_c$ are marked in red, while observations are marked in blue. The green cone represents the camera's FOV. A visual marker, attached to the door, is used to calculate the door handle observation ${}_C\boldsymbol{r}_{CH}$. If the vector from the camera to the visual marker (orange) leaves the FOV cone, the door handle observation is set to $\mathbf{0}$.

## References

[1] V. Makoviychuk, L. Wawrzyniak, Y. Guo, M. Lu, K. Storey, M. Macklin, D. Hoeller, N. Rudin, A. Allshire, A. Handa, et al. Isaac gym: High performance gpu-based physics simulation for robot learning. *arXiv e-prints*, 2021.

[2] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal policy optimization algorithms. *arXiv e-prints*, 2017.

[3] 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*, pages 91–100. PMLR, 2022.

[4] J. Merel, S. Tunyasuvunakool, A. Ahuja, Y. Tassa, L. Hasenclever, V. Pham, T. Erez, G. Wayne, and N. Heess. Catch & carry: reusable neural controllers for vision-guided whole-body tasks. *ACM Transactions on Graphics (TOG)*, 39(4):39, 2020.

[5] E. Olson. Apriltag: A robust and flexible visual fiducial system. In *Proc. IEEE Int. Conf. Robot. Autom.*, pages 3400–3407. IEEE, 2011.