Learning Task-Informed Exploration Policies for Active System Identification

Marina Y. Aoyama[1], João Moura[1], Juan Del Aguila Ferrandis[1] and Sethu Vijayakumar[1]

***Abstract*— In many dynamic robotic tasks, such as striking pucks into a goal outside the reachable workspace, the robot must first identify the relevant physical properties of the object for successful task execution, as it is unable to recover from failure or retry without human intervention. To address this challenge, we propose a task-informed exploration approach, based on reinforcement learning (RL), that trains an exploration policy using rewards automatically generated from the sensitivity of a privileged task policy to errors in estimated properties. We also introduce an uncertainty-based mechanism to determine when to transition from exploration to task execution, ensuring sufficient property estimation accuracy with minimal exploration time. Our method achieves a 90% success rate on the striking task—significantly outperforming baselines that achieve at most 40% success. Additionally, we demonstrate that our task-informed exploration rewards capture the relative importance of physical properties. Finally, we validate our approach on two manipulation tasks in a physical setup. The project website is available at marina-aoyama.github.io/poke-and-strike/.**

I. INTRODUCTION

Active exploration is essential for identifying physical properties of objects, unlike visual properties. A robot might stir liquids to identify viscosity for pouring [1] or press a sponge to evaluate stiffness for wiping [2].

We address the problem of performing one-shot robotic tasks on objects with unknown physical properties—such as striking pucks with varying friction—where failure is irreversible without human intervention. These tasks require identifying the physical properties of the object prior to task execution, as incorrect assumptions about these properties can lead to failure, making online adaptation unsuitable.

II. METHOD

We propose a task-informed exploration approach that leverages a privileged task policy to generate rewards, guiding the exploration policy to identify task-relevant properties. At deployment, the robot explores, switches to task execution based on uncertainty estimates, and adapts its motion using the estimated properties, as shown in Fig. 1.

A. Privileged Task Policy Learning

We first train a privileged task policy π_{task} using RL. During training in simulation, the task policy receives the ground truth physical property values ϕ^* as input, along with the state observation s_t, and computes actions a_t. We define the task policy reward, r_{task}, based on the task's objectives.

*This work is supported by the JST Moonshot R&D (Grant No. JPMJMS2031).

[1]Authors are with the School of Informatics, The University of Edinburgh, UK. marina.aoyama@ed.ac.uk

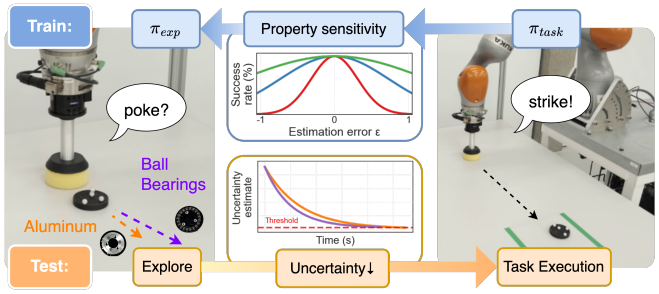

Fig. 1: **Task-informed exploration** approach enables the robot to autonomously learn how to explore and identify task-relevant properties by modeling task sensitivity to each property. For dynamic tasks such as striking, the robot must first identify the object's properties through exploratory motions to achieve task success, avoiding irreversible failure.

B. Learning of Exploration and Property Estimation

We then simultaneously train (1) an exploration policy π_{exp} to perform motions that are informative for estimating the physical properties ϕ prior to task policy execution, and (2) an online property estimator f_ϕ to infer these properties from observations during exploration.

We train an exploration policy π_{exp} via RL, given the robot and object states s_t. We define the exploration reward as

$$r_{\text{exp}} = \begin{cases} r_{\text{estimation}} & \text{if } \forall j \; \varepsilon_{\text{estimation},j} < \varepsilon_{\text{threshold},j} \\ r_{\text{failure}} & \text{otherwise} \end{cases}, \quad (1)$$

where $\varepsilon_{\text{estimation},j}$ represents the estimation error for the j-th physical property, and $\varepsilon_{\text{threshold},j}$ is the threshold for the j-th property. We obtain the estimation errors $\varepsilon_{\text{estimation}}$ by computing the difference between the ground truth physical property values ϕ^* and the values $\hat{\phi}$ estimated by the physical property estimator f_ϕ, as

$$\varepsilon_{\text{estimation},j} = \left| \phi_j^* - \hat{\phi}_j \right|. \quad (2)$$

The robot receives a negative reward r_{failure} for violating the workspace boundaries, ensuring feasible task execution after exploration, without manual reset.

For online physical property estimation during exploration, we employ a Long Short-Term Memory (LSTM) [3], as temporal information is essential for capturing object dynamics.

C. Task-Informed Exploration Reward Design

The exploration policy reward function in Eq. (1) requires the estimation error thresholds $\varepsilon_{\text{threshold},j}$. We propose to automatically generate these estimation thresholds by modelling task sensitivity to estimation error in each property.

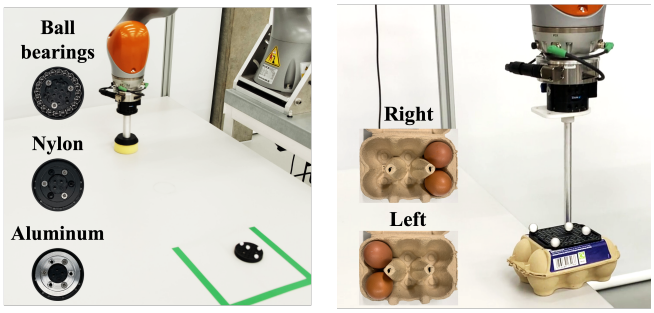

Fig. 2: Manipulation tasks.

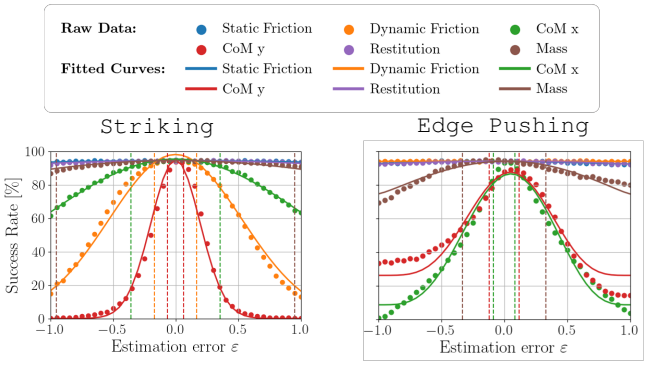

Fig. 3: Sensitivity analysis.

For each physical property, we assume a unimodal relationship between the task success rate y and the estimation error ε, where task performance is highest when the estimation is accurate and decreases as the error increases. The rate at which task success deteriorates with increasing error reflects the sensitivity of the task to each property. To model this relationship, we fit a parametric uni-modal function $g_j(\varepsilon)$ to data $\mathcal{D}_{\varepsilon,j}$ collected by rolling out the privileged task policy, systematically perturbing the ground truth value of the j-th physical property at varying levels of estimation error.

From the fitted uni-modal function g_j, we compute a set of estimation error thresholds $\varepsilon_{\text{threshold},j}$ for each physical property j, such that the task success rate remains above a proportion p of the maximum success rate, defined as:

$$g_j(\varepsilon_{\text{threshold},j}) \geq p \cdot \max_{\varepsilon} g_j(\varepsilon) \tag{3}$$

for each property. These estimation error thresholds define the success criteria in the exploration reward in Eq. (1).

D. Uncertainty-Based Policy Switching

We estimate the predictive uncertainty of the physical property estimator f_ϕ using an ensemble approach that captures both aleatoric uncertainty and epistemic uncertainty [4]. Then, we compute the uncertainty thresholds required for successful task execution by rolling out the exploration and task policies to collect uncertainty data labelled with task outcomes (success or failure). These thresholds enable the robot to switch from exploration to task execution without direct access to estimation error on a physical setup.

III. RESULTS AND DISCUSSION

We evaluate our approach on two tasks: `Striking` a puck to an unreachable goal; and `Edge Pushing` a box to the table edge, in simulation and on a physical robot (Fig. 2).

A. Does the task-informed exploration improve task success?

In the one-shot `Striking` task, domain randomisation (DR) [5] learns an average motion across properties and achieves only 25.4% success. In DR methods incorporating temporal information, using stacked observations or an LSTM backbone, exploratory pushing motions merged but they reach only 35.4% and 23.3% success, respectively. These methods rely on delayed task rewards, provided only after task execution, making it difficult to associate exploratory actions with rewards. Adding property estimation rewards achieves only 34.8%, due to the difficulty of balancing exploration and task objectives within a single policy.

The baseline methods leveraging a privileged policy (Student [6], UP-OSI [7], and RMA [8]) achieved success rates of 23.0%, 33.7%, and 31.0%, respectively. Since the privileged task policy lacks exploratory behaviour, imitating it (Student) or rolling it out—whether using online estimation (UP-OSI) or latent encodings (RMA)—leads to task failure.

In contrast, our method achieves a 90.1% success rate, significantly outperforming all baselines. Our policies achieve 92.3% success in exploration and 98.7% in task, demonstrating that the exploration policy estimates properties with sufficient accuracy for successful task performance. We confirmed consistent results on the `Edge Pushing` task.

B. Do task-informed exploration rewards capture task-relevant physical properties?

Fig. 3 presents task sensitivity to errors in each physical property parameter. In the `Striking` task, the CoM in the y-direction and dynamic friction show a sharper decline in success rate as error increases, compared to less sensitive parameters such as static friction. This sensitivity model indicates that the former parameters are more relevant for performing the task, resulting in tighter estimation thresholds for exploration rewards. Similarly, in the `Edge Pushing` task, each property exhibits distinct relationships, reflecting varying levels of relevance to task performance.

C. How does our approach perform on a physical robot?

Our approach addresses sim-to-real challenges by learning to explore and estimate task-relevant physical properties of objects, enabling the task policy to adapt its motion based on these estimates. In the `Striking` task with pucks of different friction properties (ball bearings, nylon, aluminium), the estimator distinguishes their properties during exploration, with friction estimates converging to 0.09, 0.12, and 0.15, respectively. Using these estimates, our method achieves 8/9 successful runs in a physical setup, demonstrating accurate property estimation and uncertainty-based policy switching.

IV. CONCLUSIONS

We propose and demonstrate a task-informed exploration approach to identify task-relevant properties, with uncertainty-based switching to task execution enabling one-shot manipulation of unknown objects on a physical robot.

REFERENCES

[1] T. Lopez-Guevara, R. Pucci, N. K. Taylor, M. U. Gutmann, S. Ramamoorthy, and K. Subr, "Stir to pour: Efficient calibration of liquid properties for pouring actions," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020. DOI: 10.1109/IROS45743.2020.9340852.

[2] M. Y. Aoyama, J. Moura, N. Saito, and S. Vijayakumar, "Few-shot learning of force-based motions from demonstration through pre-training of haptic representation," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 2024. DOI: 10.1109/icra57147.2024.10610502.

[3] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997. DOI: 10.1162/neco.1997.9.8.1735.

[4] J. Gawlikowski et al., "A survey of uncertainty in deep neural networks," *Artificial Intelligence Review*, vol. 56, no. Suppl 1, pp. 1513–1589, 2023. DOI: 10.1007/s10462-023-10562-9.

[5] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, "Domain randomization for transferring deep neural networks from simulation to the real world," in *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, 2017. DOI: 10.1109/iros.2017.8202133.

[6] D. Chen, B. Zhou, V. Koltun, and P. Krähenbühl, "Learning by cheating," in *Conference on Robot Learning*, 2020. [Online]. Available: https://proceedings.mlr.press/v100/chen20a.html.

[7] W. Yu, J. Tan, C. Karen Liu, and G. Turk, "Preparing for the unknown: Learning a universal policy with online system identification," in *Robotics: Science and Systems XIII*, 2017. DOI: 10.15607/rss.2017.xiii.048.

[8] A. Kumar, Z. Fu, D. Pathak, and J. Malik, "Rma: Rapid motor adaptation for legged robots," in *Robotics: Science and Systems XVII*, 2021. DOI: 10.15607/rss.2021.xvii.011.