From Skills to TAMP: Learning Portable Symbolic Representations for Task and Motion Planning

Anonymous Author(s)
Affiliation
Address
email

Abstract: To solve long-horizon tasks, robots must compose previously learned motor skills using task and motion planning (TAMP). However, TAMP presumes the existence of a symbolic task-level abstraction that is sound, complete, and refinable into motion plans. Designing this representation by hand is brittle, and existing learning methods fail to guarantee the semantics required for TAMP. We present the first approach for *learning portable symbolic representations* from pixels and poses that provably support TAMP. Our method learns (i) object-centric visual predicates and (ii) generative relational spatial predicates from skill executions. These predicates serve dually as binary classifiers over low-level states and as samplers for motion-level refinement. We discuss preliminary experiments on two real-world robot platforms, demonstrating how our approach can learn reusable symbols. In ongoing experiments, we intend to show how these symbols enable zero-shot synthesis of long-horizon plans across novel environments.

1 Introduction

Modern robot learning has produced general-purpose motor skills that manipulate objects via forceful contact [1, 2]. These reusable skills offer procedural abstraction, but combining them to solve novel tasks remains difficult: the sequencing of skills requires long-horizon, discrete decisions, interleaved with continuous reasoning over possible skill parameterizations (e.g., grasp pose selection). *Task and motion planning* (TAMP) [3] addresses this challenge by synthesizing high-level symbolic plans and refining them into collision-free motion trajectories. Yet, this relies on a symbolic task-level representation whose semantics are rarely learned: most prior work either learns symbols not designed for TAMP [4, 5, 6, 7] or bypasses symbolic reasoning altogether [8, 9].

We argue that the core missing element is a symbolic abstraction whose semantics match TAMP's assumptions. Task-level planning must be *optimistic*: it must model only logical dependencies while assuming spatial details can be resolved later. Refinement then checks feasibility given the exact scene. To make this possible, each learned symbol must act both as (1) a classifier for task-relevant conditions and (2) a sampler over feasible relative poses. This dual semantics—absent in prior work—is the linchpin enabling skill composition.

2 Background

We formalize a robot task as a set of objects O and a robot R, with universe $U = O \cup R$. The low-level state space S is object-centric, with an individual state defined as:

$$s = \big(r_b, r_c, \{o_b, o_c, o_p\}_i\big)$$

where r_b and r_c are the robot's base pose and configuration, and each object o_i has a base pose o_b, configuration o_c, and perceptual state o_p (e.g., pixels). Each grounded predicate $\bar{\sigma}(u, v)$ defines a binary classifier $\gamma_{\bar{\sigma}} : S \to \{0, 1\}$, imposing a high-level, abstract state $\bar{s} = \{\bar{\sigma} \mid \gamma_{\bar{\sigma}}(s) = 1\}$.

Skills are modeled as *composable interaction primitives* (CIPs) [2] with policy π_a, initiation set I_a, and termination set β_a. Each skill induces one or more high-level operators $\langle \Theta, \textsc{Pre}, \textsc{Eff} \rangle$. A TAMP planner searches for a task-level plan and attempts to refine it into a collision-free trajectory. Should refinement fail, the task-level plan is revised. TAMP assumes the symbolic model is sound and complete at the task level, though not necessarily at the motion level [10].

3 Method

We aim to learn a symbolic abstraction $\alpha = \langle \Sigma_s, G_s, \Sigma_v, \overline{S}, \overline{A} \rangle$ autonomously from raw skill executions, where Σ_s are spatial predicates with samplers G_s, Σ_v are visual predicates, $\overline{S}$ are symbolic states, and $\overline{A}$ are high-level operators. Given these learned symbolic abstractions, a traditional TAMP planner can be used to solve new problems using the same skills [11].

3.1 Spatial Predicate Invention

Spatial predicates capture where a skill can be executed, ignoring local obstacles. We leverage *relational critical regions* (RCRs) [12]: regions of high density in the relative pose space of object pairs during successful skill executions. See Figure 1a for a visualization of two learned RCRs.

We collect trajectories for each skill in a fast kinematic simulator, extract relative trajectories between objects, cluster relative poses in those trajectories to identify RCRs, and convert each RCR ρ_{ij} into a predicate $\sigma_{ij}(s) = \mathbb{1}[P_{ij}(s) \in \rho_{ij}]$. The support of ρ_{ij} defines a generative sampler g_{ij} over the objects' relative pose space, enabling refinement. Because they assume free space, these predicates are optimistic and portable; local obstacles merely prune samples at planning time.

3.2 Visual Predicate Invention

Some skills depend on non-geometric properties (e.g., whether a jar is open or closed). To represent these properties as visual predicates, we adapt the skills-to-symbols framework [13, 5]. First, we partition observed skill transitions (Φ, ω, Φ'), where Φ and Φ' are perceptual observations before and after a skill ω, such that $P(\Phi' \mid \Phi, \omega) = P(\Phi' \mid \omega)$. We then fit density estimators over each partition to produce visual predicates σ_k^v. These serve as precondition and effect symbols reusable across environments sharing the same objects. Example visual symbols are shown in Figure 2.

3.3 Operator Induction

We abstract collected trajectories using the learned predicates into symbolic trajectories $[\bar{s}_1, \ldots, \bar{s}_T]$, then extract abstract state transitions $(\bar{s}_i, \bar{s}_{i+1})$ and lift them over object types to induce parameterized operators $\langle \Theta, \textsc{Pre}, \textsc{Eff} \rangle$. We merge spatial and visual operators using the Cartesian product: if $\bar{a}_s = \langle \Theta_s, \textsc{Pre}_s, \textsc{Eff}_s \rangle$ and $\bar{a}_v = \langle \Theta_v, \textsc{Pre}_v, \textsc{Eff}_v \rangle$, we define:

$$\bar{a} = \langle \Theta_s \cup \Theta_v,\ \textsc{Pre}_s \wedge \textsc{Pre}_v, \textsc{Eff}_s \cup \textsc{Eff}_v \rangle.$$

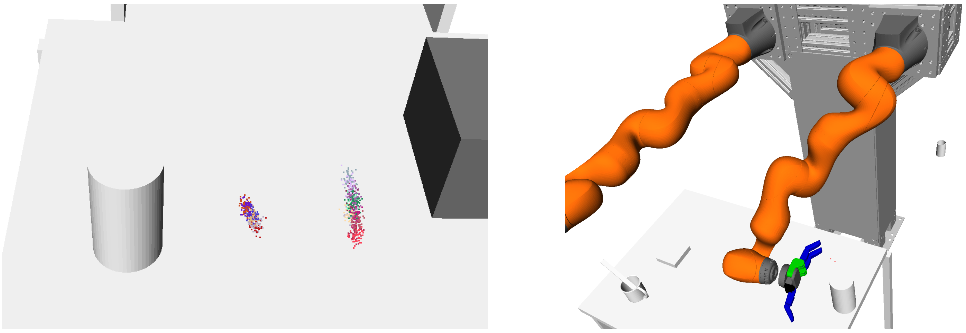
This yields a unified operator set $\overline{A}$ usable by standard TAMP planners [3].

4 Preliminary Experiments

We validate our approach in preliminary real-world experiments on two robot platforms: a mobile manipulator quadruped and a bimanual manipulator. In this section, we present initial results from these experiments and discuss our objectives for ongoing experiments.

4.1 Bimanual Manipulator

In our bimanual manipulation setting, the robot is tasked with opening a jar of peanut butter and spreading peanut butter onto a slice of bread. The robot is provided with the motion-planning-based skill `Grasp(?graspable)` and three skills based on relative trajectory playback:



(a) A visualization of learned generative samplers for object-relative end-effector poses for grasping (left cluster) and preparing to grasp (right cluster).

(b) A visualization of the simulated robot with its end-effector at an example *pre-grasp* pose sampled from a learned distribution.

Figure 1: Preliminary results on a simulated bimanual manipulator with learned pose samplers.

`Spread(?bread, ?knife)`, `Scoop(?jar, ?knife)`, and `Open(?jar)`. In Figure 1, we visualize samples from the learned spatial predicates for the `Grasp` skill. These results demonstrate that our approach for learning spatial predicates can identify multiple critical regions for a single skill, corresponding to boundaries between sub-trajectories during the skill. In ongoing experiments, we are collecting egocentric image data between skill executions on the bimanual manipulator platform. These data will be used to train visual predicates for this domain.

4.2 Mobile Manipulator

To demonstrate the generality of our proposed approach, we are also conducting real-world experiments on a Boston Dynamics Spot mobile manipulator. The Spot is provided a diverse set of object-centric skills implemented using motion planning (e.g., `Pick` or `GoTo`), trajectory playback (e.g., `OpenCabinet`), force control (e.g., `Erase`), and off-the-shelf policies (e.g., `OpenDoor`). In this setting, the Spot must erase a whiteboard in another room, but first has to retrieve the eraser from a cabinet and open a closed door blocking the way. This experimental setup is particularly challenging from an abstraction learning perspective due to the necessity of onboard data collection: the robot must both execute the skills and then capture egocentric observations of affected objects.

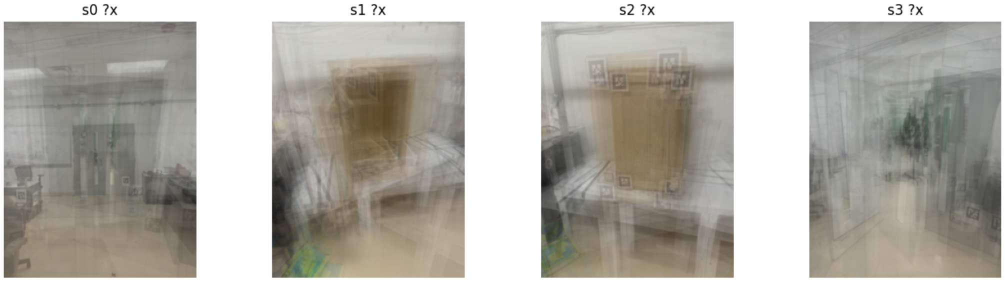


Figure 2: Renderings of four visual symbols corresponding to learned clusters of image observations. Presently, the data used to learn these symbols was manually collected within our intended mobile manipulation environment. From left to right, these symbols correspond to states where a door is closed, a cabinet is open, a cabinet is closed, or a door is open.

Consequently, as a temporary stopgap, we present visual predicates in Figure 2 that our approach learned using manually collected data. Nonetheless, these symbols demonstrate that our method can extract human-interpretable distinctions (e.g., *"The door is open."* vs. *"The door is closed."*) from raw image observations and corresponding synthetic skill execution feasibility traces. We are presently integrating the existing robot skills with interleaved photo-taking and state management, which is intended to enable the real-world system to execute skills while seamlessly collecting the necessary object-centric image observations.

5 Discussion and Conclusion

We have introduced the first method for learning symbolic representations from pixels and poses that are provably compatible with typical TAMP systems. The key is to split symbols into visual and spatial components, giving each the dual semantics of classifiers and samplers. This allows a robot to plan at the symbolic level while retaining the ability to refine plans into concrete motions.

Our work bridges the gap between learned skills and generalizable planning. Limitations include the need for known object poses during training and the assumption of discrete, known object types. In future work, we aim to integrate pose estimation and type discovery to relax these assumptions.

References

[1] O. Kroemer, S. Niekum, and G. Konidaris. A review of robot learning for manipulation: Challenges, representations, and algorithms. *Journal of Machine Learning Research*, 22(30): 1–82, 2021.

[2] B. Abbatematteo, E. Rosen, S. Thompson, T. Akbulut, S. Rammohan, and G. Konidaris. Composable Interaction Primitives: A Structured Policy Class for Efficiently Learning Sustained-Contact Manipulation Skills. In *Proceedings of the 2024 IEEE Conference on Robotics and Automation*, pages 7522–7529, 2024.

[3] C. R. Garrett, R. Chitnis, R. Holladay, B. Kim, T. Silver, L. P. Kaelbling, and T. Lozano-Pérez. Integrated Task and Motion Planning. *Annual Review of Control, Robotics, and Autonomous Systems*, 4:265–293, 2021.

[4] B. Ames, A. Thackston, and G. Konidaris. Learning symbolic representations for planning with parameterized skills. In *Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 526–533, 2018.

[5] S. James, B. Rosman, and G. Konidaris. Autonomous learning of object-centric abstractions for high-level planning. In *Proceedings of the Tenth International Conference on Learning Representations*, 2022.

[6] R. Chitnis, T. Silver, J. B. Tenenbaum, T. Lozano-Perez, and L. P. Kaelbling. Learning neuro-symbolic relational transition models for bilevel planning. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4166–4173. IEEE, 2022.

[7] T. Silver, R. Chitnis, N. Kumar, W. McClinton, T. Lozano-Pérez, L. Kaelbling, and J. B. Tenenbaum. Predicate invention for bilevel planning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 12120–12129, 2023.

[8] J. Achterhold, M. Krimmel, and J. Stueckler. Learning temporally extended skills in continuous domains as symbolic actions for planning. In *Conference on Robot Learning*, pages 225–236. PMLR, 2023.

[9] Y. Liang, N. Kumar, H. Tang, A. Weller, J. B. Tenenbaum, T. Silver, J. F. Henriques, and K. Ellis. Visualpredicator: Learning abstract world models with neuro-symbolic predicates for robot planning. In *The Thirteenth International Conference on Learning Representations*, 2025.

[10] N. Shah, D. K. Vasudevan, K. Kumar, P. Kamojjhala, and S. Srivastava. Anytime Integrated Task and Motion Policies for Stochastic Environments. In *Proceedings of the 2020 IEEE International Conference on Robotics and Automation*, pages 9285–9291, 2020.

[11] B. Hedegaard, Z. Yang, Y. Wei, A. Jaafar, S. Tellex, G. Konidaris, and N. Shah. Beyond task and motion planning: Hierarchical robot planning with general-purpose policies. In *Arxiv*, 2025.

[12] N. Shah, J. Nagpal, P. Verma, and S. Srivastava. From Reals to Logic and Back: Inventing Symbolic Vocabularies, Actions, and Models for Planning from Raw Data. *arXiv preprint arXiv:2402.11871*, 2024.

[13] G. Konidaris, L. P. Kaelbling, and T. Lozano-Perez. From Skills to Symbols: Learning Symbolic Representations for Abstract High-Level Planning. *Journal of Artificial Intelligence Research*, 61:215–289, 2018.