SKILLWRAPPER: Autonomously Learning Interpretable Skill Abstractions with Foundation Models

Ziyi Yang[1], Shreyas Sundara Raman[1], Benned Hedegaard[1], Haotian Fu[1], Linfeng Zhao[2], Stefanie Tellex[1], George Konidaris[1], David Paulius[1†], Naman Shah[1†]

***Abstract*—We envision a future where robots are equipped "out of the box" with a library of general-purpose skills. To effectively compose these skills into long-horizon plans, a robot must understand each skill's preconditions and effects in a form that supports symbolic reasoning. Such representations should be human-interpretable so that robots may understand human commands and humans may understand robot capabilities. Unfortunately, existing approaches to skill abstraction learning often require extensive data collection or human intervention, and typically yield uninterpretable representations. We present SKILLWRAPPER, the first known *active learning* approach that leverages foundation models to learn human-interpretable abstractions of black-box robot skills, producing representations that are both *probabilistically complete* and *suitable* for planning. Given only RGB image observations before and after skill execution, our system actively collects data, invents symbolic predicates, and constructs PDDL-style operators to model the skills. We present preliminary simulation results demonstrating that the abstract representations learned by SKILLWRAPPER can be used to solve previously unseen, long-horizon tasks.**

I. INTRODUCTION

In the near future, robots will be deployed from the factory to the real world, equipped with a set of general-purpose skills to interact with their environment. However, these skills could be black boxes that were learned, engineered, or obtained in unknown ways. As a result, the conditions under which skills can be used in real-world settings may be unavailable or environment-dependent, potentially leading to failure when sequencing skills to solve unseen tasks. Herein lie two important problems: First, without understanding the conditions under which each skill can be successfully executed (i.e., *preconditions*) and the likely outcomes of execution (i.e., *effects*), a robot may fail to identify task plans that effectively use its skills. Second, skill preconditions and effects should be *interpretable* to everyday users, as they would allow users to understand the robot's decision-making process, making it easier to specify goals or task constraints.

In this paper, we present SKILLWRAPPER, the first known approach that uses foundation models to autonomously characterize robot skills, emphasizing *human-interpretable* state abstractions while guaranteeing *probabilistic completeness* and *suitability* for planning. Our approach assumes a skill-type signature as input and learns a PDDL-style [1] symbolic model for each skill. Previous work has extensively explored learning symbolic representations of high-level skills [2, 3, 4, 5].

[1]Brown University, Providence, RI, USA.
[2]Northeastern University, Boston, MA, USA.
[†]Equal advising, listed in alphabetical order.
We gratefully acknowledge support from ONR Grant N00014-22-1-2592.

"There are three items *Vase*, *TissueBox*, and *Bowl*, and three locations *Sofa*, *CoffeeTable*, and *DiningTable*. Their initial positions are shown as follows. The robot is near the *Sofa* initially, and everything is placed stably, and all items can fit in every location. The goal is to have all items on the *Sofa*."

Output Plan

```
GoTo3(Sofa,CoffeeTable),
PickUp5(Vase,CoffeeTable),
GoTo2(CoffeeTable, DiningTable),
DropAt2(Vase,Sofa),
GoTo4(Sofa,DiningTable),
PickUp1(Bowl,DiningTable),
GoTo2(DiningTable,Sofa),
DropAt2(Bowl,Sofa)
```

Fig. 1: An example multi-modal task specification using natural language and egocentric visual observations, followed by the corresponding plan found by planning using the PDDL-style operators learned by SKILLWRAPPER.

However, these works either assume access to privileged information (e.g., object poses [4] or extensive human feedback [5]) or fail to produce human-interpretable representations [2]. Although prior work has explored extracting symbols and language directly from demonstrations [6], existing approaches require significant manual effort to define features and train specialized classifiers for representation learning.

Hence, to facilitate learning skill abstractions from raw robot observations while reducing effort from human experts, we utilize foundation models, such as large language models (LLMs) and vision-language models (VLMs). Several works have exploited language models for robot decision-making and planning [7, 8, 9, 10, 11]. In contrast to these approaches, our method generates planning operators compatible by design with task planners, thus benefiting from efficient domain-independent heuristics [12] and correctness guarantees. This paper briefly introduces SKILLWRAPPER while highlighting key insights from our preliminary experiments in simulation.

II. METHOD

Briefly, SKILLWRAPPER (Figure 2) learns an abstract model for planning with a library of black-box skills by *(1)* actively proposing and *(2)* executing exploratory skill sequences to collect execution traces, *(3)* inventing predicates by contrasting

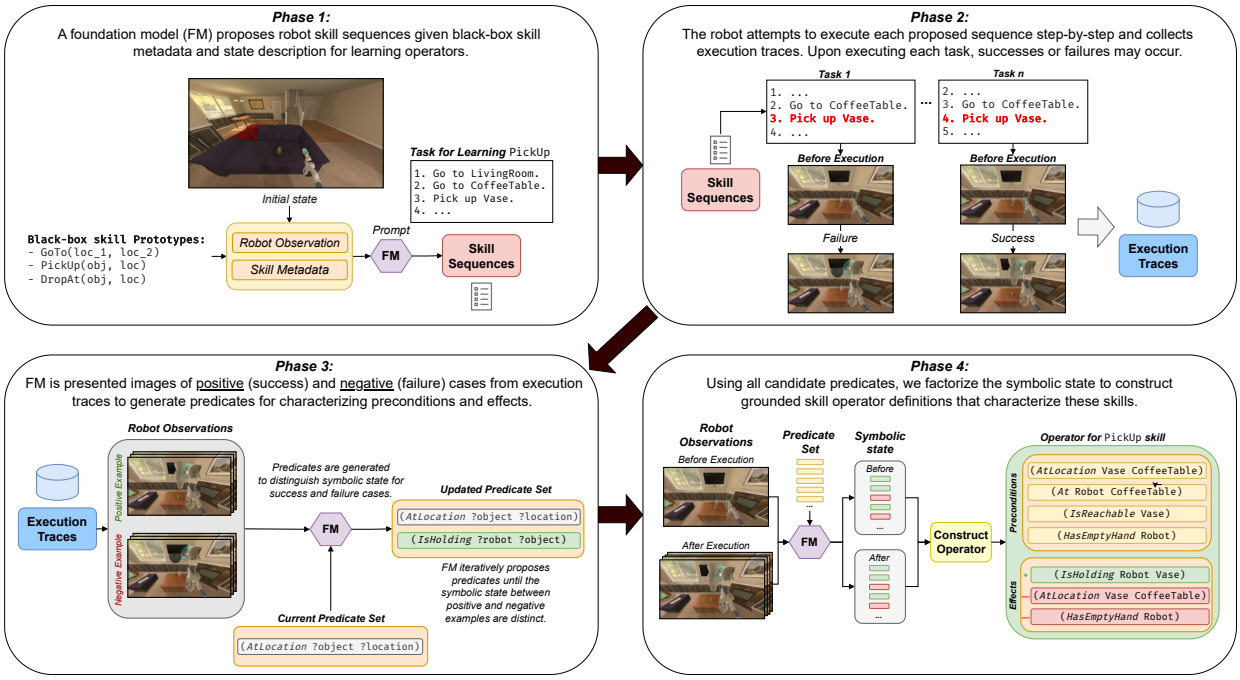

Fig. 2: Overview of SKILLWRAPPER: in (1), given a description of the robot's environment and metadata about its skills, a foundation model (FM) proposes skill sequences useful for representation learning. In (2), the robot attempts to execute the proposed sequences, collecting the initial and final state of each action as images stored in a database. In (3), this database is presented to the FM as *contrastive pairs* (i.e., success and failure images as positive and negative examples) from which the FM will invent predicates to describe the symbolic state across all skills. Finally, in (4), the FM is used to infer the abstract states corresponding to the states before and after each successful execution trace. The resulting abstract transitions are used to construct planning operators.

pairs of successful and failed skill executions, and *(4)* using these predicates to generate PDDL-style operators compatible by design with off-the-shelf classical AI planners.

Skill Sequence Proposal: Our system first queries a foundation model to generate skill sequences intended to explore the symbolic state space in a directed manner.

Predicate Invention: SKILLWRAPPER uses foundation models to generate interpretable predicates, along with their semantic meanings as English sentences. Predicate invention aims to generate predicates that distinguish state features responsible for successful or unsuccessful skill executions.

Operator Learning by Clustering: Using the collected dataset of skill execution traces, SKILLWRAPPER evaluates the truth value of each predicate at every traced state, inducing a dataset of abstract state transitions. The operator learning algorithm identifies the effects and preconditions of the potentially multiple subgoal options corresponding to each skill [2].

Planning with Learned Operators: Having "wrapped" the black-box skills in corresponding learned operators, SKILLWRAPPER can solve task planning problems conveniently specified using natural language and images (Figure 1). To convert the multi-modal task specification into an initial PDDL state, the system queries a foundation model to classify whether each predicate holds given the current state description, in a way similar to existing work [13].

III. EXPERIMENTS

We demonstrate the capabilities of the SKILLWRAPPER system using preliminary simulation experiments in the ManipulaThor [14, 15] environment. We prompt GPT-4o [16] with egocentric observations to evaluate the truth value of each predicate. For predicate invention and skill sequence proposal, we utilize o1-preview [17]. We provide the simulated robot with three high-level actions: `PickUp(obj, loc)`, `DropAt(obj, loc)`, and `GoTo(loc1, loc2)`. These high-level actions function as black-box skills by executing a deterministic sequence of low-level motions.

In total, we collected data from five skill sequences consisting of 80 image-based states and 40 transitions, of which 24 skill executions were successful. Given the dataset of environment transitions and the abstract state space induced by the invented predicates, SKILLWRAPPER learned ten operators to model the three high-level skills. Once the system has learned the human-interpretable predicates and operators, we use a foundation model to formulate an unseen task planning problem. We present an example multi-modal task specification using natural language and images at the top of Figure 1. Given the learned abstract transition model and the task planning problem inferred from the above task specification, the task planner returned the plan shown at the bottom of Figure 1.

IV. CONCLUSION

This paper formulates the problem of actively learning interpretable, abstract representations of black-box skills. We propose SKILLWRAPPER as a solution that exploits the multi-modal reasoning capabilities of foundation models. We demonstrate our approach using a proof-of-concept example in a simulated mobile manipulation setting. In ongoing and future work, we plan to compare our approach to alternative methods for active relational abstraction learning.

References

[1] D. McDermott, M. Ghallab, A. Howe, C. Knoblock, A. Ram, M. Veloso, D. Weld, and D. Wilkins, "PDDL – The Planning Domain Definition Language," CVC TR-98-003/DCS TR-1165, Yale Center for Computational Vision and Control, Tech. Rep., 1998.

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

[3] T. Silver, R. Chitnis, N. Kumar, W. McClinton, T. Lozano-Pérez, L. Kaelbling, and J. B. Tenenbaum, "Predicate Invention for Bilevel Planning," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 10, pp. 12 120–12 129, Jun. 2023.

[4] 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.

[5] M. Han, Y. Zhu, S.-C. Zhu, Y. N. Wu, and Y. Zhu, "InterPreT: Interactive Predicate Learning from Language Feedback for Generalizable Task Planning," in *Robotics: Science and Systems (RSS)*, 2024.

[6] N. Gopalan, E. Rosen, G. Konidaris, and S. Tellex, "Simultaneously Learning Transferable Symbols and Language Groundings from Perceptual Data for Instruction Following," in *Robotics: Science and Systems (RSS)*, 2020.

[7] M. Ahn, A. Brohan, N. Brown, Y. Chebotar, O. Cortes, B. David, C. Finn, C. Fu, K. Gopalakrishnan, K. Hausman, A. Herzog, D. Ho, J. Hsu, J. Ibarz, B. Ichter, A. Irpan, E. Jang, R. J. Ruano, K. Jeffrey, S. Jesmonth, N. Joshi, R. Julian, D. Kalashnikov, Y. Kuang, K.-H. Lee, S. Levine, Y. Lu, L. Luu, C. Parada, P. Pastor, J. Quiambao, K. Rao, J. Rettinghouse, D. Reyes, P. Sermanet, N. Sievers, C. Tan, A. Toshev, V. Vanhoucke, F. Xia, T. Xiao, P. Xu, S. Xu, M. Yan, and A. Zeng, "Do As I Can, Not As I Say: Grounding Language in Robotic Affordances," in *Proceedings of The 6th Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, vol. 205. PMLR, 14–18 Dec 2022, pp. 287–318.

[8] D. Driess, F. Xia, M. S. M. Sajjadi, C. Lynch, A. Chowdhery, B. Ichter, A. Wahid, J. Tompson, Q. Vuong, T. Yu, W. Huang, Y. Chebotar, P. Sermanet, D. Duckworth, S. Levine, V. Vanhoucke, K. Hausman, M. Toussaint, K. Greff, A. Zeng, I. Mordatch, and P. Florence, "PaLM-E: An Embodied Multimodal Language Model," in *Proceedings of the 40th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 202. PMLR, 23–29 Jul 2023, pp. 8469–8488.

[9] J. Liang, W. Huang, F. Xia, P. Xu, K. Hausman, B. Ichter, P. Florence, and A. Zeng, "Code as policies: Language model programs for embodied control," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 9493–9500.

[10] J. Wu, R. Antonova, A. Kan, M. Lepert, A. Zeng, S. Song, J. Bohg, S. Rusinkiewicz, and T. Funkhouser, "TidyBot: personalized robot assistance with large language models," *Autonomous Robots*, vol. 47, no. 8, p. 1087–1102, Nov. 2023.

[11] S. S. Raman, V. Cohen, I. Idrees, E. Rosen, R. Mooney, S. Tellex, and D. Paulius, "CAPE: Corrective Actions from Precondition Errors using Large Language Models," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 2024, pp. 14 070–14 077.

[12] M. Helmert, "The Fast Downward Planning System," *Journal of Artificial Intelligence Research*, vol. 26, pp. 191–246, 2006.

[13] B. Liu, Y. Jiang, X. Zhang, Q. Liu, S. Zhang, J. Biswas, and P. Stone, "LLM+P: Empowering Large Language Models with Optimal Planning Proficiency," *arXiv preprint arXiv:2304.11477*, 2023.

[14] E. Kolve, R. Mottaghi, W. Han, E. VanderBilt, L. Weihs, A. Herrasti, D. Gordon, Y. Zhu, A. Gupta, and A. Farhadi, "AI2-THOR: An Interactive 3D Environment for Visual AI," *arXiv*, 2017.

[15] K. Ehsani, W. Han, A. Herrasti, E. VanderBilt, L. Weihs, E. Kolve, A. Kembhavi, and R. Mottaghi, "ManipulaTHOR: A Framework for Visual Object Manipulation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 4497–4506.

[16] OpenAI, "Hello GPT-4o," 2024, accessed: May 17, 2025. [Online]. Available: https://openai.com/index/hello-gpt-4o/

[17] ——, "Introducing OpenAI o1-preview," 2024, accessed: May 17, 2025. [Online]. Available: https://openai.com/index/introducing-openai-o1-preview/