TAMPering with RLBench: Enabling joint developments in Task and Motion Planning and Reinforcement Learning research

Alexander Dürr[1], Jacek Malec[1], and Elin A. Topp[1]

***Abstract*— The field of robotics, spanning task and motion planning (TAMP), hierarchical reinforcement learning (HRL), and neuro-symbolic AI, faces challenges in handling complex long-horizon tasks with sparse rewards. Although planning approaches show potential, scalability is limited by the lack of accurate world models and symbolic abstractions. More reliable data are needed to support learning these representations and unifying fragmented subcommunities. This paper presents an enhanced simulation platform, built on RLBench, designed to meet the need for efficient data generation. While RLBench was created purely for reinforcement learning (RL) research, our simulator generates a richer variety of data required for the research fields TAMP, RL, and neuro-symbolic AI, supporting the study of symbolic and composable representations, multimodal inputs, and hierarchical abstractions. Our platform supports the evaluation of generalizable and interpretable world models, addressing key data generation challenges in robotics. This can foster collaboration between fragmented research areas and contributes to the development of robust and scalable systems for robotic planning.**

I. INTRODUCTION

Robotics research, particularly long-horizon tasks with sparse rewards, faces several technical challenges that complicate the development of robust and scalable solutions. End-to-end learning methods struggle to form reliable abstractions and world models in these complex tasks, while planning-based approaches, although more effective in such scenarios, are limited by the lack of accurate symbolic representations of real-world environments. This bottleneck has hindered the effectiveness of planning methods and their application to real-world tasks.

An important avenue of research focuses on learning symbolic and composable abstractions that bridge different levels of robotic tasks, from task and motion planning (TAMP) to high-level symbolic reasoning and hierarchical reinforcement learning (HRL). However, progress in this area has been constrained by the scarcity of high-quality and diverse data sets. The ability to quickly generate a large and high-quality data set from simulated various environments and tasks with flexible abstractions is crucial for advancing research on abstraction learning and planning.

This paper introduces an enhanced simulation platform, built on RLBench [1], that addresses these limitations by significantly improving data generation capabilities. Our platform, as illustrated in Figure 1, offers a richer and more diverse set of task configurations, enabling us to explore key challenges related to abstraction learning, task distribution,

[1]All authors are with the Department of Computer Science, Lund University LTH, 22100 Lund, Sweden `{alexander.durr,jacek.malec,elin_a.topp}@cs.lth.se`

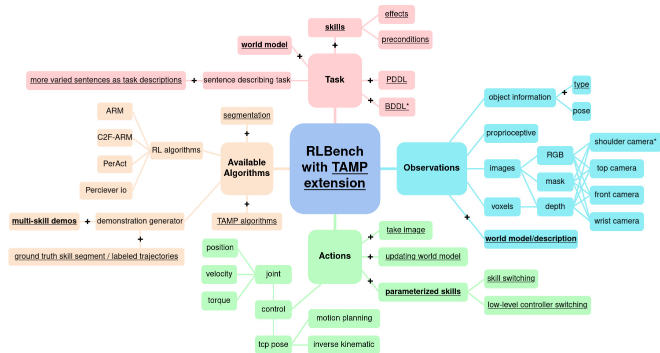


Fig. 1: Properties and capabilities of RLBench and the capabilities added by our TAMP RLBench extension.

and integration of multimodal data (e.g., language, vision, and sensorimotor data streams). The observation space is extended by abstract observations. The action space allows parameterized skills with skill and low-level controller switching. TAMP-friendly skills are described for the task with preconditions and effects. PDDL is available, and the implementation of BDDL is ongoing work. World models can be defined, and tasks can be built with specific properties. The generated demonstrations are more varied with multiple underlying skills. The simulator is designed to unlock new research possibilities, allowing testing of symbolic representations, composable models, and hierarchical abstractions across a wide variety of robotic tasks.

Our platform aims at several challenges in TAMP and RL, as discussed in [2], [3], [4], [5].

Sparse Rewards and Long-Horizon Tasks: Many robotic tasks span long time horizons with sparse rewards, making it difficult for traditional end-to-end learning methods to effectively learn appropriate abstractions. The delayed or unclear feedback often slows down the development of models capable of reasoning over extended periods.

Absence of Accurate World Models: The effectiveness of planning-based methods hinges on the availability of accurate world models, often represented as symbolic abstractions that encode the dynamics of the environment. Without reliable abstractions, even advanced planning methods struggle to perform well in real-world applications. Developing systems that can autonomously learn these representations is a major challenge.

Need for Generalizable Abstractions: Generalization remains a significant issue in robotic planning. Abstractions learned in one task or domain often fail to transfer to new, unseen tasks, reducing the robustness of planning systems. Developing composable abstractions that can be reused across multiple domains is vital to achieve long-term autonomy and adaptability in robotics.

Hierarchical Planning and Task Decomposition: Hierarchical approaches, which decompose tasks into subtasks operating at various levels of abstraction, are increasingly necessary to manage the complexity of real-world tasks. However, effectively learning and leveraging these hierarchies remains a challenge, as systems must be able to identify appropriate levels of abstraction and coordination.

Interpretable and Explainable Abstractions: The growing importance of interpretable and explainable abstractions, particularly in human-robot interaction contexts, presents another challenge. It is critical that learned abstractions are both understandable and verifiable by humans to ensure trust and predictable behavior in robotic systems. This challenge becomes even more pronounced when dealing with complex models that incorporate natural language or other high-level representations.

Data Scarcity and Simulation Limitations: High-quality, diverse data sets are essential for developing generalizable models. However, many existing robotic simulation platforms lack the flexibility and richness needed to support learning across a wide range of tasks. This limitation restricts the scope of research, especially in tasks requiring complex or composable abstractions.

By addressing these challenges, our enhanced simulation platform offers a critical resource to advance the field of robotic planning and learning, supporting the development of more generalizable, interpretable, and scalable models[1].

II. RELATED WORK

An excellent overview of TAMP, RL and other AI on robots is provided by Kroemer et al. in their comprehensive review of robot learning for manipulation tasks [6]. Our goal is to develop a simulator benchmark that meets the needs of multiple research disciplines. In this section, we explore related publications to identify their requirements and discuss existing simulators and benchmarks.

Task and Motion Planning (TAMP): Significant advances in TAMP have demonstrated the potential of symbolic representations and hierarchical models to enhance robotic reasoning and generalize task planning across environments through discovery [7]. A key challenge in this domain is to develop robust world models that generalize to various tasks and environments. For the environment, egocentric and object-centric representations have been explored as effective methods to enable task transfer [8], [9], [10]. To support this, a simulator must facilitate both approaches, offering different tasks using the same set of objects. Furthermore, the simulator environment should have hierarchical concepts such that visual concept learning can be done. On a real robot, this has been shown in dialogue and human-robot interaction, enabling abstract reasoning [11]. The needs of these methods for egocentric and object-centric tasks, as well as hierarchical concepts, emphasize that a suitable simulator must allow for symbolic learning and task abstraction.

Reinforcement Learning (RL): RL approaches have explored learning reward functions [12], through inverse reinforcement learning [13], or language grounding to map instructions to reward functions [14]. This highlights the importance of simulators that provide predefined reward functions for ground-truth comparison and language-driven tasks with automatic generation of different levels of granular language instructions. Furthermore, See-SPOT-Run (SSR) and GLIB present models that improve RL sample efficiency and task transfer, demonstrating the importance of models that facilitate exploration and efficient task transfer [15], [16]. These advancements necessitate simulators capable of handling complex, multi-step tasks.

Skill Learning and Representation: Skill policies and their representation have been extensively studied through various approaches [17], [18], including probabilistic segmentation of demonstrations [19], logical program-based policies [20], and diffusion models [21]. The discovery and composition of skills is an important area of focus, and particularly when demonstrations are given as continuous trajectories, segmentation is required. Autonomously generated skill libraries were achieved through the combined learning of segmentation and movement primitives [22]. For example, Skill Machines demonstrate how previously learned skills can be combined logically and temporally to solve new tasks efficiently [23]. These approaches highlight the need for simulators capable of segmenting and labeling skill demonstrations. With segment ground-truth labels an additional evaluation metric is available and it allows to run ablation studies to examine the algorithms for better skill discovery.

Transition Models: In model-based approaches, learning accurate transition models is essential to predict the outcomes of actions. Transition models have been learned through relational frameworks and neuro-symbolic approaches [16], [24], [14]. A simulator must support the learning of transition models by providing environments that allow for object interaction and observable as well as hidden state changes, with the flexibility to adjust the granularity of state transitions.

Hierarchical Planning: Research on hierarchical planning and control has progressed significantly with the further development of visuomotor skills as low-level controllers [11], [25]. Approaches with task-level planners can be used for long-range tasks, where agents' capabilities to perform up to 60 actions in a single task have been demonstrated [24]. There is a pressing need for a simulator that supports long-horizon tasks and allows seamless integration of visuomotor skills.

Hybrid State and Action Spaces: Although much research has focused on discrete low-dimensional spaces, more recent work has addressed hybrid, continuous, and high-dimensional spaces [15], [10], [26]. To compare and evaluate approaches, it is necessary that simulators can generate multimodal data and provide tasks at varying levels of granularity to accommodate low-dimensional to high-dimensional,

[1]The code and instructions can be found on the github project page: `https://alexanderdurr.github.io/TAMPering-with-RLBench`

continuous and discrete decision-making.

Benchmarks and Simulators: Several benchmarks and simulators have been proposed, including LIBERO [27], AI2-THOR [28], and BEHAVIOR-100 [29], which are used by the research community to evaluate task performance in simulated and real (recorded) environments [26], [24], [30]. However, none of these simulators offers an integrated platform that brings together fragmented research efforts from different fields into a single cohesive benchmark. To address this gap, our proposed simulator enable us to combine implementations and share them easily, on a platform that can be simulated and re-created in reality, supporting sim2real experiments, and allowing for cross-disciplinary comparisons.

In summary, while significant progress has been made in task and motion planning, reinforcement learning, and skill representation, there remains a need for a comprehensive simulator benchmark that can unify these fragmented efforts. By supporting symbolic learning, multimodal state-action spaces, long-horizon tasks, and skill composition, the proposed simulator will provide a powerful platform to advance research in these fields.

III. TOOL DESCRIPTION

Our enhanced simulation platform, built on RLBench, addresses critical challenges in data generation for robotics research by offering a flexible and scalable environment for a wide range of tasks. The improvements focus on increasing the diversity, volume, and quality of the available data, supporting the exploration of research questions in TAMP and RL. The extension is designed such that new RL developments based on RLBench alone can be seamlessly integrated and run with our extended RLBench version, with the benefit that TAMP research can benefit from new RL algorithms and vice versa, enabling synergies through combination of developments in both research fields.

A non-exhaustive list of TAMP and RL research problems is listed in Table I, which can be explored with the modified version of RLBench.

Figure 2 shows how and from what demonstrations are generated in RLBench and what is required. RLBench utilizes simple moves that stop the robot almost completely at each waypoint and successively actuate the gripper. This limits the research basically to one skill (move, then actuate gripper) and does not allow interesting TAMP research. Highlighted in blue are the main elements that were changed to allow for a more diverse set of demonstrations, with several skills and ground-truth information needed for TAMP research. In the following subsections, we will discuss these elements in the context of which research they enable.

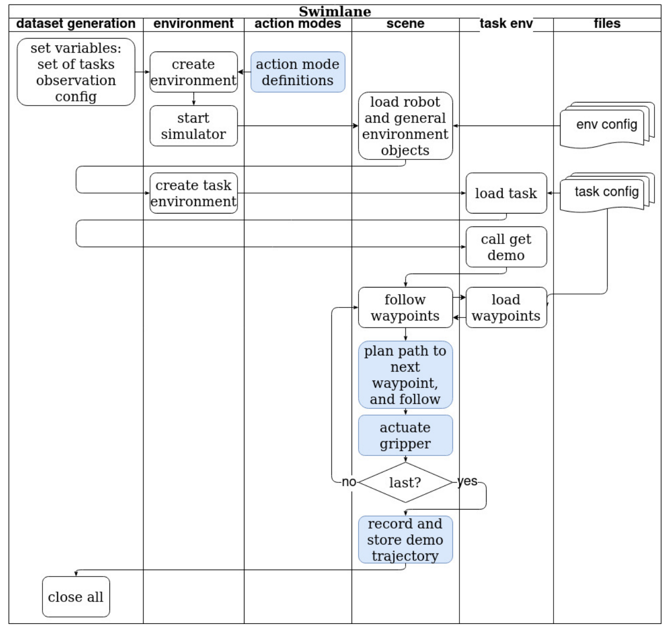


Fig. 2: Swimlane of RLBench dataset generation and its main modifications to achieve multi-skill demonstrations.

A. Comprehensive Data Generation for Long-Horizon Tasks

The simulator is designed for rapid and scalable data generation, particularly suited for long-horizon tasks with sparse rewards. This is inherent to RLBench and its underlying simulator CoppeliaSim [32]. By simulating a diverse array of task environments with many possible variations, the platform enables the generation of large datasets with diverse datapoints.

The comparison of RLBench with other simulators, and how our extension improves on RLBench, is presented in Table II, comparing the available data types. Our extension for more data types is essential for learning segmentations, abstractions, symbolic and composable representations, as well as skill and visual encodings. This is achieved by modification of the observation space by additional data types, modules that interface with CoppeliaSim to get these additional data, modules that augment the task description in language through LLM interfaces, and modules that generate PDDL descriptions through the task files. All this is done with minimal impact on existing RLBench interfaces.

B. Flexible Environment Configurations for Model Training

Our platform supports highly configurable task environments, allowing dynamic modification of configurations and variations. This flexibility enables experimentation with various symbolic representations, world models, and planning algorithms.

We add the ability to describe the environment through a world model. A declared world model and an initialized task environment can be used to create a world description through automatic instantiation. During the agent's interaction with the environment, updates of the world description are done through assigning the correct values in the description. Instantiation and updates are achieved by reading the state of the environment through an interface with the simulator.

Second, given constraints on the world description, task environment configurations, and variations can be launched in line with the constrained world description, allowing the

TABLE I: Research problem formulations and assumptions

Research problems	Approach	Given	Hidden (only for evaluation)	Related work
Learning state and action abstractions	SAT solver	meta-problem: set of classical planning problems		[7]
Learning portable representations	learn symbolic vocabulary & operators and learn to instantiate	transition data		[8], [10]
Learning concepts	unsupervised learning system based on pre-trained vision and language models	full observation space, language annotations		[11]
Learning skill segments	supervised learning	temporal sequence of proprioception, ground truth skill segments/labels	none	
	unsupervised learning	temporal sequence of proprioception	ground truth skill segments/labels	
Learning skill preconditions and effects	supervised learning	full observation space, ground truth skill segments/labels, ground truth preconditions and effects	none	
	unsupervised learning	full observation space, ground truth skill segments/labels	ground truth preconditions and effects	
Combined learning of skill segments and conditions	unsupervised learning	full observation space	ground truth skill segments/labels, ground truth pre- and postconditions	[19], [22]
Learning motions from demonstrations	encoding or generative methods (VAE, GAN)	temporal sequence of proprioception	skill used to generate demo	[31]
Learning behavior	immitation learning	full observation space, logical-program elements		[20]
Learning compositions	controller composition by priority order	environment interaction, full observation space		[9]
Learning task transfer	RL	environment interaction, full observation space		[15]
Learning efficient exploration	RL	environment interaction, full observation space		[16]
Learning reward functions	RL	environment interaction, full observation space, rewards		[12], [13]
	grounding	full observation space, rewards, language description		[14]
Learning transition models	diverse	full observation space		[14], [16], [24]

TABLE II: Comparison of Data Produced by RLBench, LIBERO, AI2-THOR, and BEHAVIOR-100

Benchmark	**Simulated**	**Real-World Data**	**RGB Data**	**Depth Data**	**Object Mask Data**	**Semantic/instance segmentation images**	**Pose Data**	**Grasping Data**	**Language Annotations**	**Proprioception**	**Primitive-segmented trajectories**	**PDDL: domain & problem file**	**BDDL: activity definition file**
RLBench	✓	✗	✓	✓	✓	✓	✓	✗	✓	✓	✗	✗	✗
LIBERO	✓	✗	✓	✗	✗	✗	✓	✓	✓	✓	✗	✗	✓
AI2-THOR	✓	✗	✓	✓	✗	✗	✓	✗	✗	✗	✗	✗	✗
BEHAVIOR-100	✓	✗	✓	✓	✓	✓	✓	some	✗	some	✗	✗	✓
RLBench with TAMP extension (our)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	ongoing work

study of specific task cases.

These adaptable configurations are crucial for environment modeling and understanding if and how learned abstractions agree with ground-truth abstractions used during environment generation.

Robust and versatile abstractions are important to enable their use across different environments and tasks, for generalization, and to ensure that learned behavioral models using these abstractions are robust as well. This is achieved by world models with the ability to model and describe several tasks and environments of RLBench.

C. Multimodal Data Collection for Abstraction Learning

The platform is designed to collect multimodal data, including visual, linguistic, and proprioceptive data streams. This feature is vital for modern learning approaches using transformers [33]. The visual data are unchanged compared to RLBench and contain RGB and depth data from several cameras with different poses. We increase the amount and variation of this linguistic data through modules interfacing with language models for task descriptions, world descriptions, and skill descriptions. For more varied task descriptions, at task environment creation time, an LLM can be queried to give more similar sentences. The variation of proprioceptive data is increased by enabling multi-skill demonstrations, compared to RLBench's single-skill demonstrations. In RLBench, the action type is required and fixed because it is needed to create the environment, and the selection of action type equals one type of skill/movement. To enable the use of several skills, we add to the action module a new type of action that lets us/the agent change the skill at run-time. This is done through a skill-parameter and a list of parameters for the selected parameterized skill.

Apart from multimodal transformer approaches, these data are useful for the refinement of symbolic world models, particularly in the context of neuro-symbolic AI, where symbolic representations are integrated with neural networks processing sensory data. The availability of multimodal data improves research on models that incorporate natural language or vision-language pre-trained models, providing a valuable resource for studying complex robotic tasks.

D. Support for Hierarchical Control and Action Abstractions

The simulator is optimized for hierarchical task learning, allowing the investigation of action and state abstractions at multiple levels of granularity. This capability facilitates the study of the interaction between low-level motor control and high-level symbolic planning, supporting the development of hierarchical reinforcement learning methods and hybrid controllers.

With the additional type of action with skill-parameter and an associated skill description in PDDL format, as well as a world model-based world description, a planner or high-level controller can be used for skill and sub-goal selection. This selection is forwarded through the skill-parameter of the action type, and through the skill's list of parameters an available low-level controller can be selected.

Second, the observation space of our RLBench extension allows for a mix of high-level/abstract observations of the environment's state, as well as low-level/raw observations, such as image or sensor data. This is achieved by extending the configuration of the available observation space and additional modules to extract abstract information. Thus, it is possible to explore how different levels of abstraction affect RL algorithms.

Overall, our enhanced simulator provides a critical resource for advancing research in robotic planning and learning, offering comprehensive data generation, flexible configurations, and support for multimodal and hierarchical abstraction learning. These capabilities enable the research community to push the boundaries of current robotic systems and explore new frontiers in planning and abstraction.

IV. RESULTS

The results section presents the findings of our experiments to substantiate our claims of the tool description and its properties. We identified the following key areas.

A. Data compatibility with related work methods

As shown in Table I, related works in the field of TAMP and RL examine different research problems and use different types of data. Our extended RLBench version provides the data types shown in Table II, which covers most of the related work discussed in Section II. The implementation of BDDL is ongoing work. To try out an existing method and its algorithm implementation, one has to pre-process the extended RLBench data to be in a format to be consumed by the algorithm.

B. Segmentation

Some related works, such as [19], [33], [34] require or learn segmented trajectories. In Figure 3 we show the segmentation method[2] for the functioning of [33], [34], which defines keyframes at points in time when the arm is stopped or the gripper changes states (open/closed). In contrast to the single-skill demos (move, then actuate gripper) used in these works, the shown demonstration was generated with two parameterized skills; one of which moves the robot smoothly, without stopping, through an approach waypoint. Since the other approaches assume simple full-stop moves between waypoints, they would fail by crashing into the object at an angle, due to the wrong approach direction. This approach only works on demonstrations that have these properties, which is in the above cases conveniently generated through the demonstration generating algorithm. This approach will fail if the skills do not require stopping the arm, do not rely on a motion planner, and are generated through other means (e.g., by human demonstrations). Our version of RLBench allows for ground truth segmentation by labeling trajectories at demo generation time, and enables learning trajectory segmentation through supervised means, or for evaluation of unsupervised learned segmentation models, as shown in [19].

[2]https://github.com/stepjam/ARM/blob/main/arm/demo_loading_utils.py#L21

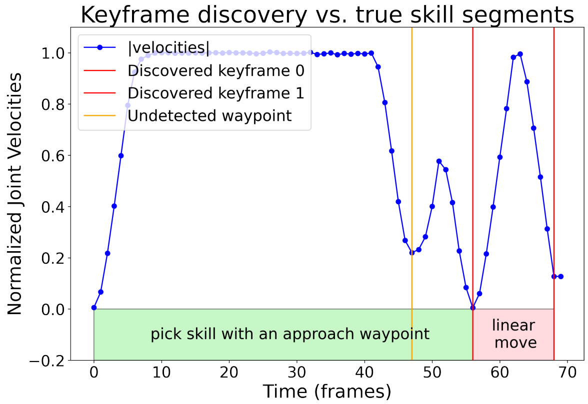


Fig. 3: Comparison between keyframe discovery and ground truth skill segments when multiple types of skills are present in one demonstration. Norm of all joint velocities (blue), showing stops of the robot near the x-axis. Keyframes discovered (red) by the related work method [34], assuming a single specific skill. Undetected waypoint (orange) for approach required for success. Boxes showing ground-truth segments of pick skill (green) and retraction (red).

C. TAMP modifications maintain RL algorithm applicability

To enable the synergies between RL and TAMP developments, we demonstrate that existing RL algorithms work on the extended RLBench version and show that the version does not break developments based purely on RLBench. We run RL algorithms [35] in the gym environment and the C2F-ARM algorithm [34] on the extended demonstration data. Both algorithms work on the modified data without any alterations. This is trivial since the python objects `Demo` and `Observation` contain all original RLBench data that are read by these methods, while the additional data is ignored, see an example `Observation` instance in Listing 1.

```
vars(demo._observations[0])

{'left_shoulder_rgb': None,
 'left_shoulder_depth': None,
 'left_shoulder_mask': None,
 'left_shoulder_point_cloud': None,
 :
 'joint_velocities': array([ 7.62939453e-05, ...]),
 'joint_positions': array([ 4.80415110e-07, ...]),
 'joint_forces': array([-2.20681336e-02,  ...]),
 'gripper_open': 1.0,
 'gripper_pose': array([ 2.78499901e-01, ...]),
 'gripper_matrix': array([[-9.70800757e-01, ...]]),
 'gripper_joint_positions': array([0.03999685, ...]),
 'gripper_touch_forces': array([ 1.78904529e-03, ...]),
 'task_low_dim_state': array([ 2.79052228e-01, ...]),
 'misc': {'left_shoulder_camera_extrinsics': array([[
      1.73648179e-01,  ...]]),
  'left_shoulder_camera_intrinsics': array
       ([[-175.8385604, ...]]),
  'left_shoulder_camera_near': 0.00999999977...,
  'left_shoulder_camera_far': 3.200000047683716,
  :
  }
```

Listing 1: Demonstration data.

D. Simulation to reality

To apply in-simulation-learned policies on the real robot, also known as sim2real, the data and setup need to be aligned between simulation and reality. In Figure 4 we show visual examples of the alignment of simulated and real vision data. The cameras were calibrated and their poses (position and orientation) determined through a developed workflow [36]. This allows us to get visual data with the same perspective and distance as in the standard RLBench environment, and makes re-training of pre-trained models done on RLBench unnecessary. The values of the Intel RealSense depth data stream are scaled to match the range of RLBench.

The proprioceptive data from the simulated robot, generated by CoppeliaSim's physics engine, and the proprioceptive data of the real robot are sufficiently matching in position and velocity. Torque readings show different characteristics, but can be mitigated through advanced sim2real algorithms [37], which augment simulated torque readings to make them appear more realistic during simulated training. Torque is inherently difficult to match since each real robot differs in torque behavior due to a plethora of environmental factors (version, age, temperature/friction, mount). If the use of torque observations is required to solve the task at hand, then realistic noise during learning in simulation is a good option to allow for transfer of behavior.

V. DISCUSSION

Our simulator provides a powerful tool to address several key challenges in robotic planning and abstraction learning. By generating large and diverse datasets, the platform facilitates exploration of critical objectives such as soundness, completeness, and planning efficiency, as possible goals for abstraction learning. Its flexibility enables experimentation with various levels of abstraction across different task distributions, supporting the development of effective planning strategies.

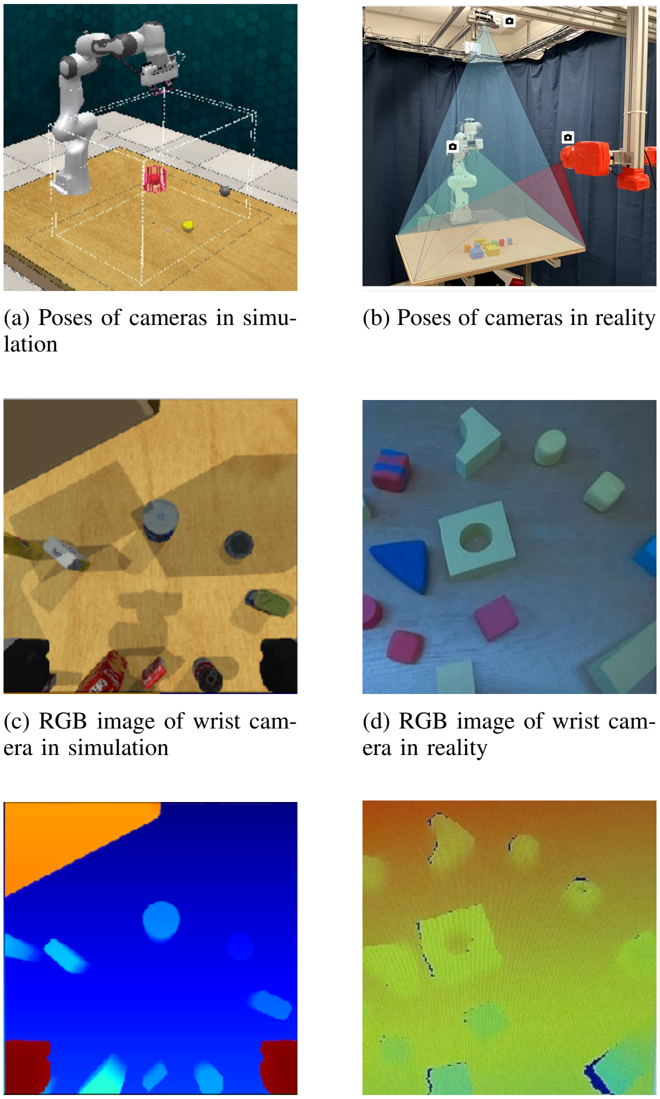
In addition, the simulator's multimodal data capabilities, combined with integration with large language models (LLMs) and vision-language models (VLMs), facilitate the creation of interpretable and explainable abstractions. This feature allows to study how symbolic representations can be communicated effectively to humans. Furthermore, the compatibility of the platform with pre-trained models enables us to leverage these models for abstraction learning.

Overall, the simulator provides a flexible and scalable environment to tackle critical challenges such as developing generalizable abstractions, improving hierarchical planning, and enhancing neuro-symbolic AI. Its ability to simulate diverse tasks and environments supports the development of robust world models essential for long-term autonomy in robotics.

VI. CONCLUSIONS

This paper presents an enhanced simulation platform that significantly improves data generation for robotics research, with particular relevance to task and motion planning, hierarchical reinforcement learning, and neuro-symbolic AI. By

simulated setup **real setup**



(a) Poses of cameras in simulation

(b) Poses of cameras in reality

(c) RGB image of wrist camera in simulation

(d) RGB image of wrist camera in reality

(e) Depth image of wrist camera in simulation

(f) Depth image of wrist camera in reality

Fig. 4: Comparison of RLBench setup and its replication in reality.

providing a rich, flexible, and scalable data source, the simulator enables the tackling of critical questions surrounding the learning of symbolic and composable representations for robotic planning.

The platform addresses key challenges in robotics, such as sparse rewards, long-horizon tasks, and the need for generalizable and interpretable abstractions, pushing the boundaries of current robotic systems. Its advanced data generation capabilities lay the foundation for developing robust, scalable, and autonomous robotics, making it a vital tool to advance the field and to foster collaboration between fragmented research areas. This, in turn, helps bridge the gap between learning and planning approaches.

Ongoing work includes improving trajectory segmentation methods to enable skill learning, including preconditions and effects, improvements to the extension to enable more TAMP developments to be transferred (such as BDDL), and the transfer of in-simulation learned policies to transfer to the real robot setup. Future work is the exploration of combinations of RL and TAMP algorithms in one system to extend our existing efforts [38], and learning skill-primitive representations through encoding and generative methods [31].

VII. ACKNOWLEDGMENT

The computations and data handling were enabled by resources provided by the National Academic Infrastructure for Supercomputing in Sweden (NAISS), partially funded by the Swedish Research Council through grant agreement no. 2022-06725.

This work was partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation.

REFERENCES

[1] S. James, Z. Ma, D. R. Arrojo, and A. J. Davison, “Rlbench: The robot learning benchmark & learning environment,” *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3019–3026, 2020.

[2] K. Zhang, E. Lucet, J. A. D. Sandretto, S. Kchir, and D. Filliat, “Task and motion planning methods: applications and limitations,” in *ICINCO 2022-19th International Conference on Informatics in Control, Automation and Robotics*. SCITEPRESS-Science and Technology Publications, 2022, pp. 476–483.

[3] M. Mansouri, F. Pecora, and P. Schüller, “Combining task and motion planning: Challenges and guidelines,” *Frontiers in Robotics and AI*, vol. 8, p. 637888, 2021.

[4] G. Dulac-Arnold, N. Levine, D. J. Mankowitz, J. Li, C. Paduraru, S. Gowal, and T. Hester, “Challenges of real-world reinforcement learning: definitions, benchmarks and analysis,” *Machine Learning*, vol. 110, no. 9, pp. 2419–2468, 2021.

[5] J. Ibarz, J. Tan, C. Finn, M. Kalakrishnan, P. Pastor, and S. Levine, “How to train your robot with deep reinforcement learning: lessons we have learned,” *The International Journal of Robotics Research*, vol. 40, no. 4-5, pp. 698–721, 2021.

[6] O. Kroemer, S. Niekum, and G. Konidaris, “A review of robot learning for manipulation: Challenges, representations, and algorithms,” *Journal of machine learning research*, vol. 22, no. 30, pp. 1–82, 2021.

[7] A. Curtis, T. Silver, J. B. Tenenbaum, T. Lozano-Pérez, and L. Kaelbling, “Discovering state and action abstractions for generalized task and motion planning,” in *Proceedings of the AAAI conference on artificial intelligence*, vol. 36, no. 5, 2022, pp. 5377–5384.

[8] S. James, B. Rosman, and G. Konidaris, “Learning portable representations for high-level planning,” in *International Conference on Machine Learning*, 2020.

[9] M. Sharma, J. Liang, J. Zhao, A. LaGrassa, and O. Kroemer, “Learning to compose hierarchical object-centric controllers for robotic manipulation,” in *Conference on Robot Learning*, 2020.

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

[11] W. Gu, A. Sah, and N. Gopalan, “Interactive visual task learning for robots,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 9, 2024, pp. 10 297–10 305.

[12] E. C. Williams, N. Gopalan, M. Rhee, and S. Tellex, “Learning to parse natural language to grounded reward functions with weak supervision,” in *2018 ieee international conference on robotics and automation (icra)*. IEEE, 2018, pp. 4430–4436.

[13] M. Cockcroft, S. Mawjee, S. James, and P. Ranchod, “Learning options from demonstration using skill segmentation,” in *2020 International SAUPEC/RobMech/PRASA Conference*. IEEE, 2020, pp. 1–6.

[14] N. Gopalan, M. Littman, J. MacGlashan, S. Squire, S. Tellex, J. Winder, L. Wong, *et al.*, "Planning with abstract markov decision processes," in *Proceedings of the International Conference on Automated Planning and Scheduling*, vol. 27, 2017, pp. 480–488.

[15] A. Hundt, A. Murali, P. Hubli, R. Liu, N. Gopalan, M. Gombolay, and G. D. Hager, ""good robot! now watch this!": Repurposing reinforcement learning for task-to-task transfer," in *5th Annual Conference on Robot Learning*, 2021. [Online]. Available: https://openreview.net/forum?id=Pxs5XwId51n

[16] R. Chitnis, T. Silver, J. B. Tenenbaum, L. P. Kaelbling, and T. Lozano-Pérez, "Glib: Efficient exploration for relational model-based reinforcement learning via goal-literal babbling," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 13, 2021, pp. 11 782–11 791.

[17] M. Mayr, F. Rovida, and V. Krueger, "Skiros2: A skill-based robot control platform for ros," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2023, pp. 6273–6280.

[18] E. A. Topp, M. Stenmark, A. Ganslandt, A. Svensson, M. Haage, and J. Malec, "Ontology-based knowledge representation for increased skill reusability in industrial robots," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 5672–5678.

[19] R. Lioutikov, G. Neumann, G. Maeda, and J. Peters, "Probabilistic segmentation applied to an assembly task," in *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*. IEEE, 2015, pp. 533–540.

[20] T. Silver, K. R. Allen, A. K. Lew, L. P. Kaelbling, and J. Tenenbaum, "Few-shot bayesian imitation learning with logical program policies," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 06, 2020, pp. 10 251–10 258.

[21] A. Ajay, Y. Du, A. Gupta, J. B. Tenenbaum, T. S. Jaakkola, and P. Agrawal, "Is conditional generative modeling all you need for decision making?" in *The Eleventh International Conference on Learning Representations*, 2023. [Online]. Available: https://openreview.net/forum?id=sP1fo2K9DFG

[22] R. Lioutikov, G. Neumann, G. Maeda, and J. Peters, "Learning movement primitive libraries through probabilistic segmentation," *The International Journal of Robotics Research*, vol. 36, no. 8, pp. 879–894, 2017.

[23] G. N. Tasse, D. Jarvis, S. James, and B. Rosman, "Skill machines: Temporal logic skill composition in reinforcement learning," in *The Twelfth International Conference on Learning Representations*, 2024. [Online]. Available: https://openreview.net/forum?id=qiduMcw3CU

[24] 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)*. IEEE, 2022, pp. 4166–4173.

[25] A. Ajay, S. Han, Y. Du, S. Li, A. Gupta, T. Jaakkola, J. Tenenbaum, L. Kaelbling, A. Srivastava, and P. Agrawal, "Compositional foundation models for hierarchical planning," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[26] X. Jia, Q. Wang, A. Donat, B. Xing, G. Li, H. Zhou, O. Celik, D. Blessing, R. Lioutikov, and G. Neumann, "MaIL: Improving imitation learning with selective state space models," in *8th Annual Conference on Robot Learning*, 2024. [Online]. Available: https://openreview.net/forum?id=IssXUYvVTg

[27] B. Liu, Y. Zhu, C. Gao, Y. Feng, Q. Liu, Y. Zhu, and P. Stone, "Libero: Benchmarking knowledge transfer for lifelong robot learning," *arXiv preprint arXiv:2306.03310*, 2023.

[28] 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.

[29] S. Srivastava, C. Li, M. Lingelbach, R. Martín-Martín, F. Xia, K. E. Vainio, Z. Lian, C. Gokmen, S. Buch, K. Liu, S. Savarese, H. Gweon, J. Wu, and L. Fei-Fei, "Behavior: Benchmark for everyday household activities in virtual, interactive, and ecological environments," in *Proceedings of the 5th Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, A. Faust, D. Hsu, and G. Neumann, Eds., vol. 164. PMLR, 08–11 Nov 2022, pp. 477–490. [Online]. Available: https://proceedings.mlr.press/v164/srivastava22a.html

[30] N. Kumar, W. McClinton, R. Chitnis, T. Silver, T. Lozano-Pérez, and L. P. Kaelbling, "Learning efficient abstract planning models that choose what to predict," in *Conference on Robot Learning*. PMLR, 2023, pp. 2070–2095.

[31] Q. Yang, A. Dürr, E. A. Topp, J. A. Stork, and T. Stoyanov, "Variable impedance skill learning for contact-rich manipulation," *IEEE robotics and automation letters*, vol. 7, no. 3, pp. 8391–8398, 2022.

[32] E. Rohmer, S. P. N. Singh, and M. Freese, "Coppeliasim (formerly v-rep): a versatile and scalable robot simulation framework," in *Proc. of The International Conference on Intelligent Robots and Systems (IROS)*, 2013, www.coppeliarobotics.com.

[33] M. Shridhar, L. Manuelli, and D. Fox, "Perceiver-actor: A multi-task transformer for robotic manipulation," in *Proceedings of the 6th Conference on Robot Learning (CoRL)*, 2022.

[34] S. James, K. Wada, T. Laidlow, and A. J. Davison, "Coarse-to-fine q-attention: Efficient learning for visual robotic manipulation via discretisation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 13 739–13 748.

[35] A. Raffin, A. Hill, A. Gleave, A. Kanervisto, M. Ernestus, and N. Dormann, "Stable-baselines3: Reliable reinforcement learning implementations," *Journal of Machine Learning Research*, vol. 22, no. 268, pp. 1–8, 2021. [Online]. Available: http://jmlr.org/papers/v22/20-1364.html

[36] L. Jakobsson and O. Larsson, "Camera calibration for alignment of a real world setup to a simulated environment," 2024, Master Thesis, LTH, Lund University.

[37] A. Dürr, L. Neric, V. Krueger, and E. A. Topp, "Realeasy: Real-time capable simulation to reality domain adaptation," in *2021 IEEE 17th International Conference on Automation Science and Engineering (CASE)*. IEEE, 2021, pp. 2009–2016.

[38] M. Mayr, F. Ahmad, A. Duerr, and V. Krueger, "Using knowledge representation and task planning for robot-agnostic skills on the example of contact-rich wiping tasks," in *2023 IEEE 19th International Conference on Automation Science and Engineering (CASE)*. IEEE, 2023, pp. 1–7.