Towards Benchmarking Robotic Manipulation in Space

Andrej Orsula
University of Luxembourg
Luxembourg
andrej.orsula@uni.lu

Antoine Richard
University of Luxembourg
Luxembourg
antoine.richard@uni.lu

Matthieu Geist
Cohere
Canada
matthieu@cohere.com

Miguel Olivares-Mendez
University of Luxembourg
Luxembourg
miguel.olivaresmendez@uni.lu

Carol Martinez
University of Luxembourg
Luxembourg
carol.martinezluna@uni.lu

Abstract: As humanity ventures deeper into space, the demand for autonomous robotic systems capable of performing complex manipulation sequences is becoming increasingly critical. This work introduces a set of tasks designed to explore learning-based approaches in the context of space robotics while emphasizing the need for generalization and adaptability. The benchmark leverages procedural generation and parallel simulation environments to expose agents to a wide range of scenarios across different domains of space. Preliminary results highlight the challenges posed by procedural variability and underscore the importance of evaluating generalization capabilities in the design of the benchmark. The source code is publicly available at github.com/AndrejOrsula/space_robotics_bench.

Keywords: Benchmark, Robotic Manipulation, Space Robotics

1 Introduction

Robotic manipulation plays a crucial role in the future of space exploration, with applications ranging from planetary outpost construction [1] to orbital assembly and servicing [2, 3]. With the increasing ambition of envisioned space missions, the demand for autonomous robotic systems capable of executing complex task sequences grows rapidly. However, the domain of space introduces several unique challenges due to the presence of harsh environmental conditions that are coupled with significant computational constraints and limited availability of human intervention as a result of communication delays. Addressing these challenges requires the development of adaptive and robust control strategies that can generalize across a wide range of scenarios, making space robotics an attractive domain for exploring learning-based approaches.

Recent advances in robot learning have shown promise in acquiring general-purpose manipulation skills through high-capacity models trained on large-scale datasets [4]. However, there is a substantial gap between the available terrestrial datasets and the requirements of robots operating in space. Moreover, the safety-critical nature of space systems and the limited availability of laboratory setups for training and validation make collecting such datasets infeasible. On the other hand, simulation-based benchmarks have played a crucial role in advancing robot learning research, particularly in the field of reinforcement learning (RL) [5, 6]. Although several benchmarks have been introduced for manipulation, they mainly focus on the table-top setting [7] or assembly of predefined objects in a controlled environment [8]. Similarly, standalone frameworks for applying RL to space robotics have been developed for applications such as planetary and orbital navigation [9, 10]. However, the lack of standardized benchmarks tailored for space robotics has limited the progress in exploring and developing learning-based approaches for extraterrestrial applications.

8th Conference on Robot Learning (CoRL 2024), Munich, Germany.

In this work, we introduce the first steps towards benchmarking robotic manipulation in space by presenting a novel simulation-based framework designed for exploring and evaluating learning algorithms in space-relevant scenarios. The benchmark focuses on key manipulation problems that are considered to be essential for the future of space exploration and sustainable presence in space while exposing agents to a wide range of scenarios by leveraging procedural generation and parallel simulation instances. Consequently, generalization is emphasized with the goal of contributing to the development of adaptive systems capable of operating under the challenging conditions of space.

2 Structure and Design

This benchmark is an integral part of the Space Robotics Bench initiative, which aims to provide a standardized framework for evaluating robotic systems in space-relevant scenarios. The initiative is driven by the need to develop generalizable and robust algorithms that can adapt to the uncertainties and complexities of space environments. The manipulation benchmark focuses on single-arm scenarios in which agents are required to interact with their surroundings in the SE(3) space using a general-purpose robotic arm equipped with a mechanical gripper. All tasks are designed to be robot-agnostic, allowing researchers to experiment with different systems without extensive reconfiguration. The benchmark is compatible with the Gymnasium API for seamless integration with learning frameworks while also exposing a ROS 2 interface to simplify interoperability with external systems for their iterative development and validation. Furthermore, a containerized Docker environment is provided to ensure reproducibility and ease of deployment across different platforms.

2.1 Procedural Generation

Procedural generation is a powerful technique for creating diverse and realistic environments without relying solely on static datasets that might be difficult to obtain and limited in scope. This approach allows for the generation of nearly an infinite number of unique environments, which has been widely adopted in the gaming industry but remains largely underutilized in the fields of robotics and space exploration. While existing RL benchmarks leverage procedural generation [6], they focus on 2D games rather than realistic 3D scenarios. However, procedural generation also has the advantage of being able to produce assets of various dimensions at a configurable level of detail, enabling each asset to be tailored for the specific objective and performance requirements of the task. Furthermore, this mechanism can contribute towards curriculum learning by gradually increasing the difficulty of the task through the generation of more complex assets.

We use Blender [11] to generate a wide range of procedural assets showcased in Figure 1. Following the workflow established in prior work [12, 13], the mesh geometry of each asset is produced through Blender's Geometry Nodes, which is a node-based system that provides parametric control over the creation, manipulation, and modification of arbitrary geometry and data types. At the same time, the appearance of the assets is defined using Shader Nodes that enable the creation of complex visual materials that can leverage procedural textures and mapping techniques. By introducing

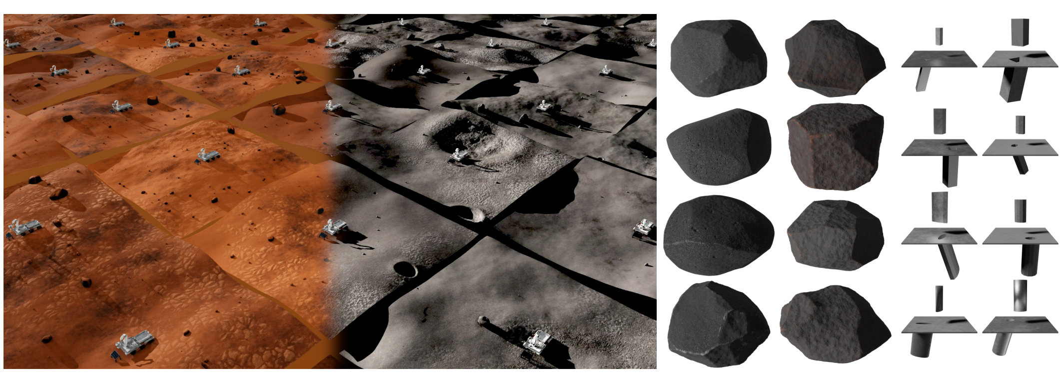


Figure 1: Examples of procedurally generated terrains, rocks, and peg-in-hole assembly modules.

randomness and variation within these pipelines, a diverse set of unique assets can be generated by simply adjusting the initial pseudo-random seed. The geometry of each asset can then be directly imported into a number of robotics simulators, while the visual material is baked into a set of PBR textures at a configurable level of detail to ensure a realistic visual appearance and runtime performance. Our benchmark fully automates this process to seamlessly generate unique assets for each environment at runtime. Not only does this approach enhance the diversity of scenarios, but it also eliminates the need to store large datasets, making the benchmark more accessible and scalable.

2.2 Parallel Environments

The benchmark is built on top of NVIDIA Isaac Sim through the Isaac Lab framework [14] to leverage its parallelized physics and rendering capabilities. Therefore, each task can be simulated across multiple parallel instances, significantly accelerating workflows such as hyperparameter tuning, synthetic data generation, and online learning. Moreover, each environment instance can benefit from procedural generation to ensure a unique experience that contributes towards overall domain randomization. The interoperability with Isaac Lab also provides compatibility with a wide array of pre-configured robots and sensors, allowing researchers to experiment with different configurations.

In addition to facilitating online learning and iterative development, the framework also supports teleoperation for the collection of human demonstrations through various interfaces. This feature is targeted towards imitation learning and offline RL, where diverse demonstrations in procedural environments can be used to bootstrap the learning. However, no dataset is currently available while the benchmark is under active development and feedback from the community is being collected.

2.3 Task Formulation

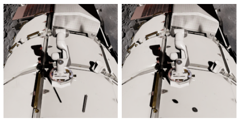
We formulate all tasks as a partially observable Markov decision process to encapsulate the sequential decision-making under uncertainty. To encourage generalization across different robotic platforms, we do not make any assumptions about the specific kinematic configuration of the robot. Instead, the agent interacts with the environment through the motion of the end-effector in the form of desired linear and angular velocities, which are mapped to joint commands via differential inverse kinematics. Similarly, the state of the gripper is controlled through a robot-agnostic binary action.

Multi-modal observations are often required to provide the agent with a comprehensive understanding of the environment. All tasks have access to proprioceptive observations in the form of end-effector pose, using 6D rotation encoding [15], and a normalized gripper state. Visual observations are provided in the form of RGB and depth images captured by two cameras, one of which is wrist-mounted, and the other perceives the scene from the base of the robot. Methods that exploit the availability of privileged information are supported by separately providing access to the ground truth state of the environment, which contains task-specific information such as the relative position of objects and targets. Additional sensory inputs, such as force-torque readings, are also included as part of the state because not all robotic systems are equipped with such sensors. Furthermore, observations whose dimensionality changes based on the specific setup, such as the number of joints, are also available but exposed separately due to their dynamic nature.

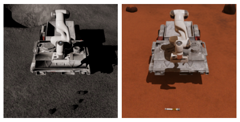
3 Domains and Tasks

The benchmark introduces four initial tasks that are designed to represent various aspects of robotic manipulation in space, including visuomotor coordination, fine motor skills, adaptability to dynamic environments, and long-horizon planning. Meanwhile, the challenges posed by each task are tailored to reflect a specific domain of space by configuring the dynamics, such as gravity, and visual appearance, such as skybox, to represent the characteristics of either the Moon, Mars or an orbital space station. Finally, all tasks implement a composite reward function as a weighted sum of task-specific sub-goals, which are designed to guide the agent towards the final objective through intermediate steps, such as reducing the proximity to the target and successful grasping.

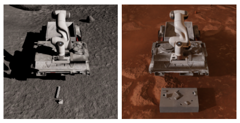
The tasks are as follows:



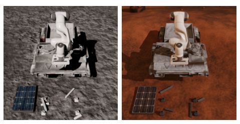
Debris Capture represents the task of stabilizing and capturing free-floating debris that is tumbling toward a robotic arm mounted on a space station. The debris can originate either from a dataset or a procedural pipeline, and the agent must stabilize it to prevent further orbital drift.



Sample Collection simulates the process of gathering physical materials from planetary surfaces. The agent is required to grasp objects and place them into a cargo bin of a rover on which the robotic arm is mounted. The objects can be either procedurally generated rocks or sample return tubes.



Peg-in-Hole Assembly poses the essential challenge of inserting a peg into a hole with a high degree of precision. The agent must grasp each peg, align it with its corresponding hole, and successfully insert it. Every peg-in-hole pair can be either from a dataset or procedurally generated.



Solar Panel Assembly extends the peg-in-hole task into a challenging long-horizon assembly sequence. The agent is required to assemble a solar panel by inserting four pegs of varying lengths into their holes, which is followed by attaching a solar cell to the assembled support structure.

4 Preliminary Benchmarking Results

We conduct a preliminary benchmarking study to evaluate the performance of RL agents on the introduced tasks. Model-based DreamerV3 by Hafner et al. [16] was selected due to its promising results with high-dimensional inputs and general robustness to hyperparameters across various domains, while its learned world model is suitable for providing insights about the design of the benchmark. It is configured to encode both proprioceptive and visual observations (64×64 px) using a model with a total capacity of 43M parameters. The agents were trained across the tasks in 128 parallel environments for 10M steps with a training ratio of 64. Furthermore, the sample collection and peg-in-hole assembly tasks are analyzed using both dataset-based and procedurally generated assets to evaluate the impact of diversity on the generalization capabilities.

Once trained, the agents were evaluated over 200 episodes in their corresponding tasks. Sample collection using static assets in the form of sample return tubes achieved a success rate of 98.5%, while the peg-in-hole assembly task reached 92.0%. However, the success rate dropped to 43.5% for collecting procedural rock samples, and the peg-in-hole assembly task with procedurally generated modules remained unsolved. Similarly, success in the solar panel assembly task was never achieved as the agent struggled to insert more than one peg of the support structure. These preliminary results suggest that the diversity introduced by procedural generation poses a significant challenge for the agents, highlighting the importance of generalization in the design of the benchmark. Furthermore, the complexity of long-horizon tasks requires additional exploration of the training dynamics, model capacity, and reward shaping to facilitate the process of learning.

5 Conclusion and Future Work

The benchmark introduced in this work aims to provide a standardized framework for exploring learning-based approaches in the domain of space robotics. By focusing on key manipulation challenges, this benchmark introduces a set of tasks that are essential for future space missions. The preliminary results highlight the importance of evaluating generalization capabilities in the design of the benchmark, as the diversity introduced by procedural generation poses a significant challenge for the agents. Future work will focus on addressing the current limitations by expanding the available tasks with more complex scenarios while emphasizing mobile manipulation that would explore the peculiarities of dynamic space environments. Parametric robot configurations will be introduced to evaluate robustness and generalization capabilities across various systems. Furthermore, elaborate evaluation metrics shall be defined to provide a comprehensive understanding of the performance across different tasks and domains before establishing thorough benchmarking baselines.

References

[1] National Aeronautics and Space Administration. Artemis Plan: NASA's Lunar Exploration Program Overview, 2020.

[2] X. Zhihui, L. Jinguo, W. Chenchen, and T. Yuchuang. Review of in-space assembly technologies. *Chinese Journal of Aeronautics*, 34(11), 2021.

[3] European Space Agency. ESA moves ahead with In-Orbit Servicing missions, 2023.

[4] Open X-Embodiment Collaboration. Open X-Embodiment: Robotic Learning Datasets and RT-X Models. *arXiv preprint arXiv:2310.08864*, 2023.

[5] S. Tunyasuvunakool, A. Muldal, Y. Doron, S. Liu, S. Bohez, J. Merel, T. Erez, T. Lillicrap, N. Heess, and Y. Tassa. dm_control: Software and tasks for continuous control. *Software Impacts*, 6, 2020. ISSN 2665-9638.

[6] K. Cobbe, C. Hesse, J. Hilton, and J. Schulman. Leveraging Procedural Generation to Benchmark Reinforcement Learning. *arXiv preprint arXiv:1912.01588*, 2019.

[7] S. James, Z. Ma, D. Rovick Arrojo, and A. J. Davison. RLBench: The Robot Learning Benchmark & Learning Environment. *IEEE Robotics and Automation Letters*, 2020.

[8] M. Heo, Y. Lee, D. Lee, and J. J. Lim. FurnitureBench: Reproducible Real-World Benchmark for Long-Horizon Complex Manipulation. In *Robotics: Science and Systems*, 2023.

[9] A. Mortensen and S. Bøgh. RLRoverLab: An Advanced Reinforcement Learning Suite for Planetary Rover Simulation and Training. In *International Conference on Space Robotics*, 2024.

[10] M. El-Hariry, A. Richard, and M. Olivares-Mendez. RANS: Highly-Parallelised Simulator for Reinforcement Learning based Autonomous Navigating Spacecrafts. *arXiv preprint arXiv:2310.07393*, 2023.

[11] Blender Development Team. Blender 4.2 LTS, 2024.

[12] A. Orsula, S. Bøgh, M. Olivares-Mendez, and C. Martinez. Learning to Grasp on the Moon from 3D Octree Observations with Deep Reinforcement Learning. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2022.

[13] A. Orsula, M. Geist, M. Olivares-Mendez, and C. Martinez. Leveraging Procedural Generation for Learning Autonomous Peg-in-Hole Assembly in Space. In *International Conference on Space Robotics*, 2024.

[14] M. Mittal, C. Yu, Q. Yu, J. Liu, N. Rudin, D. Hoeller, J. L. Yuan, R. Singh, Y. Guo, H. Mazhar, A. Mandlekar, B. Babich, G. State, M. Hutter, and A. Garg. Orbit: A Unified Simulation Framework for Interactive Robot Learning Environments. *IEEE Robotics and Automation Letters*, 8(6), 2023.

[15] Y. Zhou, C. Barnes, J. Lu, J. Yang, and H. Li. On the Continuity of Rotation Representations in Neural Networks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

[16] D. Hafner, J. Pasukonis, J. Ba, and T. Lillicrap. Mastering Diverse Domains through World Models. *arXiv preprint arXiv:2301.04104*, 2023.