UNIVERSITY OF LUXEMBOURG

PhD-FSTM-2025-122
Faculty of Science, Technology and Medicine

DISSERTATION

Defense held on 19 December 2025 in Luxembourg
to obtain the degree of

DOCTEUR DE L'UNIVERSITÉ DU LUXEMBOURG
EN INFORMATIQUE

by

Andrej ORSULA

Robot Learning Beyond Earth

Enabling Adaptive Autonomy in Space

UNIVERSITY OF LUXEMBOURG

Affidavit

I declare that this thesis:

- is the result of my own work. Any contribution from any other party, and any use of generative artificial intelligence technologies have been duly cited and acknowledged;
- is not substantially the same as any other that I have submitted, and;
- is not being concurrently submitted for a degree, diploma or other qualification at the University of Luxembourg or any other University or similar institution except as specified in the text.

With my approval I furthermore confirm the following:

- I have adhered to the rules set out in the University of Luxembourg's Code of Conduct and the Doctoral Education Agreement (DEA)[1], in particular with regard to Research Integrity.
- I have documented all methods, data, and processes truthfully and fully.
- I have mentioned all the significant contributors to the work.
- I am aware that the work may be screened electronically for originality.

I acknowledge that if any issues are raised regarding good research practices based on the review of the thesis, the examination may be postponed pending the outcome of any investigation of such issues. If a degree was conferred, any such subsequently discovered issues may result in the cancellation of the degree.

Approved on 2025-09-19

[1] If applicable (DEA is compulsory since August 2020)

Preface

The dream of expansion beyond our home planet has always been a powerful motivator, yet it is a vision defined by immense engineering challenges. This thesis is the embodiment of a research journey driven by a fundamental question: How do we build the adaptive autonomy that robots need to operate reliably in the remote and unstructured environments of space? This thesis details my effort to answer that question.

My research path followed two interconnected directions. The first was the creation of a virtual playground where robots could learn from a near-infinite diversity of challenges. The second involved developing and applying a learning methodology that empowers robots with the physical intelligence needed to master the complex tasks they will face in extraterrestrial environments. This work presents the design of the combined framework and the principles of learning it enables.

This endeavor would not have been possible without the guidance of my advisors, the discussions with my colleagues, and the support of my family and friends. It is my sincere hope that these contributions will serve as a small but meaningful step toward building the truly autonomous systems that will one day help us explore and build new worlds.

Defense Committee

Committee Members:	Prof. Holger Voos
	Prof. Keenan Albee
	Dr. Claudio Semini
Co-Supervisor:	Dr. Carol Martinez
Supervisor:	Prof. Miguel Olivares-Mendez

All models are wrong, but some are useful.

— George E. P. Box

Abstract

The growing ambition for a sustainable human presence beyond Earth requires autonomous robotic systems capable of reliable operation in extreme and unpredictable conditions. However, developing such autonomy is hindered by the scarcity of extraterrestrial data, the prohibitive cost of hardware testing, and the critical sim-to-real gap. This thesis confronts these obstacles by challenging the conventional pursuit of a singular high-fidelity digital twin. Instead, it proposes a paradigm of diversity over fidelity, where true robotic robustness is achieved not by perfecting one simulation, but by learning to master a massive distribution of scenarios.

To enable such a vision, this work introduces the Space Robotics Bench, a comprehensive open-source simulation framework for robot learning in space that combines scalable parallelization with an integrated procedural engine for the on-demand generation of diverse mission-relevant applications. Building on this foundation, a model-based reinforcement learning methodology is leveraged to acquire robust control policies that can adapt to novel situations.

Experimental validation demonstrates that the principle of procedural diversity yields policies capable of mastering a wide range of mission-critical capabilities, extending from planetary landing and resilient traversal on unstructured deformable terrains to high-precision assembly and tool-aware manipulation. These efforts culminate in the successful zero-shot sim-to-real transfer of a learned policy to a physical rover.

Ultimately, this thesis delivers a new paradigm for the development and validation of learning-based autonomy. By contributing a powerful open-source toolkit and a validated methodological blueprint, this work establishes a scalable pathway for developing and verifying the adaptive robotic systems that will be essential for our multiplanetary future.

Contents

Figures

Tables

Publications

Core Publications

I 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, pp. 4112–4119. doi: 10.1109/IROS47612.2022.9981661.

II 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, pp. 357–364. doi: 10.1109/iSpaRo60631.2024.10688111.

III A. Orsula, A. Richard, M. Geist, M. Olivares-Mendez, and C. Martinez, "Towards Benchmarking Robotic Manipulation in Space," in *Conference on Robot Learning Workshop on Mastering Robot Manipulation in a World of Abundant Data*, 2024.

IV A. Orsula, M. Geist, M. Olivares-Mendez, and C. Martinez, "Advancing Adaptive Autonomy through Procedural Space Environments," in *International Astronautical Congress*, 2025.

V A. Orsula, M. Geist, M. Olivares-Mendez, and C. Martinez, "Sim2Dust: Mastering Dynamic Waypoint Tracking on Granular Media," in *International Conference on Space Robotics*, 2025. doi: 10.48550/arXiv.2508.11503.

VI A. Orsula, M. Geist, M. Olivares-Mendez, and C. Martinez, "Learning Tool-Aware Adaptive Compliant Control for Autonomous Regolith Excavation," in *Symposium on Advanced Space Technologies in Robotics and Automation*, 2025. doi: 10.48550/arXiv.2509.05475.

VII A. Orsula, M. Geist, M. Olivares-Mendez, and C. Martinez, "Space Robotics Bench: Robot Learning Beyond Earth," *arXiv:2509.23328*, 2025, doi: 10.48550/arXiv.2509.23328.

Secondary Publications

VIII K. R. Barad, A. Orsula, A. Richard, J. Dentler, M. Olivares-Mendez, and C. Martinez, "Grasp-O: A Generative System for Object-Centric 6-DoF Grasping of Unknown Objects," in *Springer Proceedings in Advanced Robotics*, 2024, pp. 280–285. doi: 10.1007/978-3-031-76428-8_52.

IX K. R. Barad, A. Orsula, A. Richard, J. Dentler, M. A. Olivares-Mendez, and C. Martinez, "GraspLDM: Generative 6-DoF Grasp Synthesis Using Latent Diffusion Models," *IEEE Access*, vol. 12, pp. 164621–164633, 2024, doi: 10.1109/ACCESS.2024.3492118.

X P. Liu et al., "A Retrospective on the Robot Air Hockey Challenge: Benchmarking Robust, Reliable, and Safe Learning Techniques for Real-world Robotics," in *Advances in Neural Information Processing Systems*, 2024, pp. 9690–9726.

XI M. El Hariry, A. Orsula, M. Geist, and M. Olivares-Mendez, "RL-AVIST: Reinforcement Learning for Autonomous Visual Inspection of Space Targets," in *International Astronautical Congress*, 2025.

Glossary

Space

ESA **– European Space Agency**: An intergovernmental organization of member states dedicated to the research and exploration of space. 3

EVA **– extravehicular activity**: An activity performed outside a spacecraft in a space environment, often involving tasks like assembly, maintenance, and scientific research. 4, 18

ISAM **– in-space servicing, assembly, and manufacturing**: A paradigm of space operations focused on servicing existing satellites, assembling large structures in orbit, and manufacturing components directly in space. 20

ISRU **– in-situ resource utilization**: The practice of collecting, processing, and utilizing resources found on-site in an extraterrestrial environment to support future robotic and human missions. 3, 20, 36, 85

ISS **– International Space Station**: A modular space station in low Earth orbit, serving as a microgravity and space research laboratory. 3, 18, 61

JAXA **– Japan Aerospace Exploration Agency**: The Japanese national agency responsible for space exploration, research, and development. 3

NASA **– National Aeronautics and Space Administration**: An agency of the United States responsible for the civil space program and research. 3, 5, 20

V&V **– verification and validation**: A process used to ensure that a system meets specifications and fulfills its intended purpose, particularly in safety-critical applications. 8, 33, 50

cislunar: The region of space between the Earth and the Moon. 4

microgravity: A condition in which objects are perceived to be weightless, typically experienced in orbital environments. 4, 13, 19, 34, 51

regolith: A layer of loose superficial deposits covering solid rock. It includes dust and soil found on the surface of the Moon, Mars, and other celestial bodies. 3, 4, 5, 8, 15, 19, 20, 21, 43, 49, 51, 56, 77, 85, 87, 88, 97

terramechanics: The study of the interaction between vehicles and terrain, particularly for off-road traversal on deformable surfaces like sand or regolith. 21

Machine Learning

BC **– behavior cloning**: A basic form of imitation learning where a policy is trained in a supervised manner to directly mimic the state-action pairs from a demonstration dataset. 28

DR **– domain randomization**: A technique used to bridge the sim-to-real gap by training a policy in a simulation with a wide range of randomized parameters. 10, 11, 13, 30, 31, 33, 39, 43, 45, 46, 49, 55, 60, 62, 68, 74, 77, 81, 94

DreamerV3: A model-based reinforcement learning algorithm that learns a latent world model from observations and uses this model to train its actor-critic networks entirely within imagined trajectories. 10, 27, 64, 65, 66, 67, 68, 71, 73, 78, 79, 80, 82, 87, 90

IL **– imitation learning**: A type of machine learning where an agent learns to perform a task by observing and imitating demonstrations from a human or another controller. 28, 61

LSTM **– long short-term memory**: A type of recurrent neural network architecture that is capable of learning long-term dependencies in sequential data. 78

LfD **– learning from demonstration**: A machine learning paradigm where an agent learns to perform tasks by observing demonstrations. It can include techniques like imitation learning and behavior cloning. 17, 28, 98

MBRL **– model-based RL**: A class of reinforcement learning algorithms that employ a model of the environment dynamics. 10, 12, 13, 23, 27, 35, 46, 63, 65, 67, 70, 92, 94, 95

MDP **– Markov decision process**: A mathematical framework for modeling decision-making in situations where outcomes are at least partially under the control of an agent. It is the formal foundation for most reinforcement learning problems. 17, 24, 25, 27, 40

POMDP **– partially observable MDP**: An extension of the Markov decision process where the agent does not directly observe the full state of the environment but instead receives an observation that may be noisy or incomplete. 25, 64, 78, 87

PPO **– Proximal Policy Optimization**: An actor-critic reinforcement learning algorithm that constrains policy updates to a small region to promote stable learning by optimizing a clipped surrogate objective function. 10, 26, 64, 65, 66, 67, 78, 79

RL **– reinforcement learning**: A type of machine learning where an agent learns to make decisions by taking actions in an environment to maximize a cumulative reward signal. 6, 7, 8, 10, 12, 13, 14, 15, 17, 25, 26, 28, 29, 34, 36, 37, 39, 40, 42, 46, 49, 60, 63, 64, 67, 68, 71, 75, 76, 78, 79, 80, 82, 96

RNN **– recurrent neural network**: A class of neural networks designed for processing sequential data by maintaining a hidden state that captures information about previous inputs. 68

***SAC* – Soft Actor-Critic**: An off-policy actor-critic reinforcement learning algorithm based on the maximum entropy framework, which aims to maximize both the expected return and the entropy of the learned policy. 10, 26, 42, 64, 65, 66

***TD3* – Twin Delayed Deep Deterministic Policy Gradient**: An off-policy actor-critic algorithm that addresses value overestimation and instability by employing a pair of critic networks and delayed policy updates. 10, 26, 64, 67, 78

***TQC* – Truncated Quantile Critics**: An off-policy actor-critic algorithm that extends Soft Actor-Critic with a truncated quantile distribution to improve stability and robustness. 42

actor-critic: A class of reinforcement learning algorithms that use two separate entities in the form of an actor that selects actions based on the current policy and a critic that evaluates the actions by estimating the value of the resulting state. This architecture results in more stable and efficient learning. 25, 26, 42, 68

agent: The learner and decision-maker in a reinforcement learning problem. It perceives its environment and takes actions to maximize a cumulative reward, also known as return. 2, 4, 6, 7, 8, 10, 12, 13, 15, 21, 24, 25, 27, 28, 29, 30, 31, 34, 35, 37, 39, 40, 41, 43, 45, 46, 47, 49, 53, 60, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 78, 79, 80, 81, 83, 85, 86, 87, 88, 89, 90, 91, 92, 94, 97

end-to-end: A learning approach where a model is trained to directly map raw inputs to outputs without intermediate representations or feature extraction. This is often used in complex applications where traditional feature engineering is impractical. 12, 40, 53, 69, 83, 84, 92

environment: In reinforcement learning, the world in which the agent exists and interacts. It receives actions from the agent and returns new states and rewards. 7, 8, 9, 10, 24, 25, 26, 27, 28, 29, 30, 31, 34, 35, 36, 37, 38, 39, 40, 45, 46, 47, 48, 49, 50, 51, 54, 55, 56, 57, 58, 59, 62, 63, 64, 65, 67, 68, 69, 74, 75, 76, 77, 78, 81, 82, 86, 90, 94, 97

episode: A single sequence of interactions between an agent and its environment, starting from an initial state and ending in a terminal state. 30, 39, 60, 77, 87

generalization: The ability of a learned policy to perform effectively in scenarios that were not encountered during its training. Opposite of overfitting and a core focus of this thesis. 2, 6, 7, 8, 9, 10, 11, 12, 13, 15, 31, 33, 35, 36, 48, 53, 62, 64, 66, 68, 73, 74, 75, 76, 78, 81, 88, 92, 95, 96

latent space: A lower-dimensional representation of high-dimensional data learned by a machine learning model, often used to capture the essential features of the data while discarding noise and redundancy. 27, 42

off-policy: A category of reinforcement learning algorithms that can update the current policy using data collected from any policy, typically stored in a replay buffer. 26, 42, 64, 78

offline RL: A paradigm in reinforcement learning where the agent learns a policy from a static dataset of interactions, without any active exploration or online data collection. 28, 29

on-policy: A category of reinforcement learning algorithms that update the current policy using only data collected while acting with that same policy. 26, 64, 78

overfit: A modeling error that occurs when a function is too closely fit to a limited set of data points. In reinforcement learning, it results in a policy that performs well on its training environment but fails to generalize to new situations. 8, 10, 31, 34, 39, 45, 66, 90, 95

policy: In reinforcement learning, the strategy used by an agent to determine which action to take in a given state. In deep learning, this function is typically approximated by a neural network. 2, 5, 6, 7, 8, 9, 10, 12, 13, 15, 24, 25, 26, 27, 28, 29, 30, 31, 34, 36, 37, 39, 40, 42, 43, 45, 46, 47, 49, 53, 54, 60, 63, 65, 66, 68, 69, 72, 75, 76, 77, 78, 80, 81, 82, 83, 84, 85, 87, 90, 91, 92, 94, 97, 98

reward function: A function that defines the goal in a reinforcement learning problem. It provides a numerical signal to the agent at each step to indicate the immediate desirability of its actions. 43, 78, 87, 91

sample efficiency: A measure of how much data an algorithm requires to learn a task. Algorithms with high sample efficiency can learn effective policies from a relatively small number of environmental interactions. 10, 26, 27, 65, 68, 78, 79, 80

self-play: A training method where an agent improves by playing against other agents, often previous versions of itself. It is a powerful technique for discovering robust and general strategies. 13, 91

sim-to-real: The process and challenge of transferring a policy or model trained in a simulation environment to a physical system operating in the real world. 2, 7, 12, 13, 15, 17, 29, 32, 33, 35, 36, 44, 45, 46, 48, 49, 51, 62, 74, 75, 79, 82, 84, 92, 94, 95, 96, 98

value function: A function that estimates the expected cumulative reward, also called return, from a given state or state-action pair. It is used to evaluate the quality of states or actions in reinforcement learning. 25, 26, 28, 42

world model: A learned model of environment dynamics used in model-based reinforcement learning. It can predict future states and rewards given the current state and an action. 27, 46, 66, 68, 73, 85, 93, 94

zero-shot transfer: The process of deploying a model or a policy on a new task or in a new environment, without any additional training or fine-tuning on data from the newly encountered scenario. 12, 43, 78, 81, 97

Robotics

ATE – **average tracking error**: A measure of performance for reference tracking problems, defined as the average linear or angular distance between the reference trajectory and the actual trajectory followed by the robot. 79, 83

DoF – **degree of freedom**: The number of independent parameters that define the configuration of a mechanical system. For a robotic arm, it typically refers to the number of its actuated joints. 13, 21, 23, 39

EE – **end-effector**: The tool at the end of a robotic arm that is designed to interact with its environment, such as grippers, scoops, and drills. 5, 6, 22, 23, 37, 40, 43, 52, 53, 70, 71

IK – **inverse kinematics**: A computational method used in robotics to determine the joint configuration necessary to place the end-effector at a specific position and orientation in Cartesian space. 9, 10, 23, 53, 70, 71, 88, 90

IMU – **inertial measurement unit**: An electronic device that measures the linear acceleration, angular rate, and sometimes orientation by using a combination of accelerometers, gyroscopes, and magnetometers. 53

Jacobian: A matrix of first-order partial derivatives that relates the joint velocities of a manipulator to the linear and angular velocities of its end-effector. 23

OSC – **operational space control**: A control methodology for robotic manipulation that formulates the equations of motion and control laws directly in the task space. It allows for the explicit specification of the end-effector compliance via virtual stiffness and damping parameters. 10, 12, 13, 23, 47, 53, 63, 70, 71, 73, 87, 94, 95

SE(3) – **special Euclidean group in 3D**: The mathematical group representing rigid body motions in 3D space through a combination of all possible translations and rotations. 21, 71, 87

Cartesian space: A coordinate system in which the position of a point is defined by its distances from a set of perpendicular axes. In robotics, it is often used to describe the position and orientation in 3D space. 37, 43, 64

compliance: The ability of a robotic system to yield or deflect in response to external forces, which is a critical property for physical interaction. 9, 10, 23, 70, 71, 72, 73, 85, 87, 88, 92, 94, 95, 96

damping: A property of a control system to resist motion proportional to its velocity. In compliant control, it is used to dissipate energy and prevent oscillations. 10, 13, 23, 71, 87, 92, 94

embodiment: The physical form and structure of a robot that includes its morphology, sensors, and actuators. The embodiment fundamentally influences how an agent can interact with and learn about its environment. 15, 51, 53, 63, 72, 87

kinematics: A branch of mechanics that describes the motion of objects without considering the forces that cause the motion. 37, 40, 53

octree: A hierarchical data structure in which each internal node has exactly eight children that partition a 3D space through a recursive subdivision. 37, 40, 41, 42, 43, 46

stiffness: A property of a control system to resist displacement caused by an external force. In compliant control, high stiffness results in rigid behavior and faster response. 10, 13, 23, 71, 72, 87, 92, 94

task space: The space in which a robotic manipulator operates, defined by the position and orientation of its end-effector. 6, 23, 37, 46, 70

teleoperation: The direct control of a remote system by a human operator. In space, this is often hindered by significant communication delays. 4, 19, 28, 61, 62

Software

API – **application programming interface**: A set of protocols for building software applications that define how different software components should interact. 11, 31, 39, 50, 58

Blender: A free and open-source computer graphics software used for modeling, sculpting, animation, simulation, texturing, and rendering of 3D content. It is utilized in this thesis for both manual and procedural asset generation. 34, 38, 57, 59

Gymnasium: A widely adopted standard interface and library for reinforcement learning environments. All applications developed in this thesis are compliant with Gymnasium. 11, 50, 54, 62

Isaac Sim: A scalable robotics simulator developed by NVIDIA. It serves as the core backend for the Space Robotics Bench. 31, 49, 55

PBR – **physically-based rendering**: A computer graphics approach that seeks to render images in a way that models the flow of light in the real world, resulting in more consistent visual appearances. 31, 37, 38, 58

PCG – **procedural content generation**: The programmatic and algorithmic creation of data, such as 3D models and environments. It is used in this thesis to generate diverse training scenarios at scale. 10, 11, 12, 13, 14, 15, 33, 34, 37, 38, 45, 46, 49, 52, 56, 60, 62, 65, 68, 74, 76, 81, 86, 88, 94

***ROS* – Robot Operating System**: A framework and set of tools for writing robot software. It is a popular standard for middleware in the robotics research community. 11, 32, 39, 50, 60, 61, 62

***SRB* – Space Robotics Bench**: The open-source simulation framework developed in this thesis, designed for robot learning research in diverse and procedurally generated space environments. 13, 15, 29, 33, 36, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 60, 61, 62, 63, 64, 66, 67, 69, 72, 73, 74, 75, 76, 85, 92, 93, 94, 95, 96, 98

SimForge: The open-source procedural engine developed in this thesis for generating diverse 3D assets and environments that are particularly tailored for space robotics. 14, 15, 57, 58, 59, 60, 73, 86

***USD* – Universal Scene Description**: A file format and framework for the interchange of 3D computer graphics data. It is the native format for NVIDIA Isaac Sim. 58

middleware: A framework that acts as a communication bridge between different software components. 50, 62

1 Introduction

Space, once a realm of mythology and distant contemplation, is rapidly evolving into a dynamic domain for scientific discovery, sustained human presence, and resource exploitation as humanity actively architects its multiplanetary future. The vision of permanent settlements on the Moon and Mars is no longer confined to science fiction [1]. It has become a tangible objective pursued by a global consortium of space agencies and pioneering private ventures. This new era is driven by ambitious missions that aim to unlock the secrets of our solar system, harness extraterrestrial resources, and pave the way for human expansion into the universe [2], [3], [4].

Embarking on this transformative journey beyond Earth requires more than human courage and ingenuity alone. The environment is extreme, and the communication distances are vast. Consequently, sophisticated robotic systems are emerging not merely as tools but as indispensable enablers of this expansion. They are tasked with diverse operations from scouting planetary surfaces and assembling orbital megastructures to maintaining life-support systems in off-world outposts. The sheer scale and complexity of these missions demand more than mechanically proficient machines. The impracticality of constant human oversight mandates a fundamental shift towards autonomous systems capable of intelligent decision-making and adaptive behavior. Yet, traditional robotic control paradigms, which rely on pre-programmed execution under nominal conditions, are insufficient when confronted with the dynamic, unstructured, and often poorly understood environments of space [5]. This capability gap highlights an urgent need for data-driven control approaches that can learn from experience to develop robust strategies and adapt to unforeseen challenges.

Unlocking the full potential of robot learning in space necessitates more than just algorithmic advancements. The current landscape is characterized by a significant scarcity of relevant data, the prohibitive cost of comprehensive technology demonstrations, and limited access to representative simulations. These constraints render many terrestrial development methodologies impractical for the unique requirements of extraterrestrial missions. This thesis directly

confronts these critical challenges by proposing a new paradigm of achieving robustness not through the pursuit of a single digital twin, but through the mastery of immense procedural diversity. It introduces a comprehensive framework centered on this principle that leverages large-scale simulation and data-efficient learning to achieve the generalization required for space operations. The core of this work lies in establishing a scalable and diverse virtual testbed that facilitates the efficient training and systematic benchmarking of autonomous systems across a wide range of space-relevant scenarios. Building upon this foundational framework, the thesis further explores the applications of a novel robot learning methodology designed to equip agents with the robust, compliant, and generalizable behaviors required for complex tasks across diverse robotic platforms and application domains. The complete procedural paradigm, from scenario generation to sim-to-real validation, is conceptually illustrated in Figure 1.1.

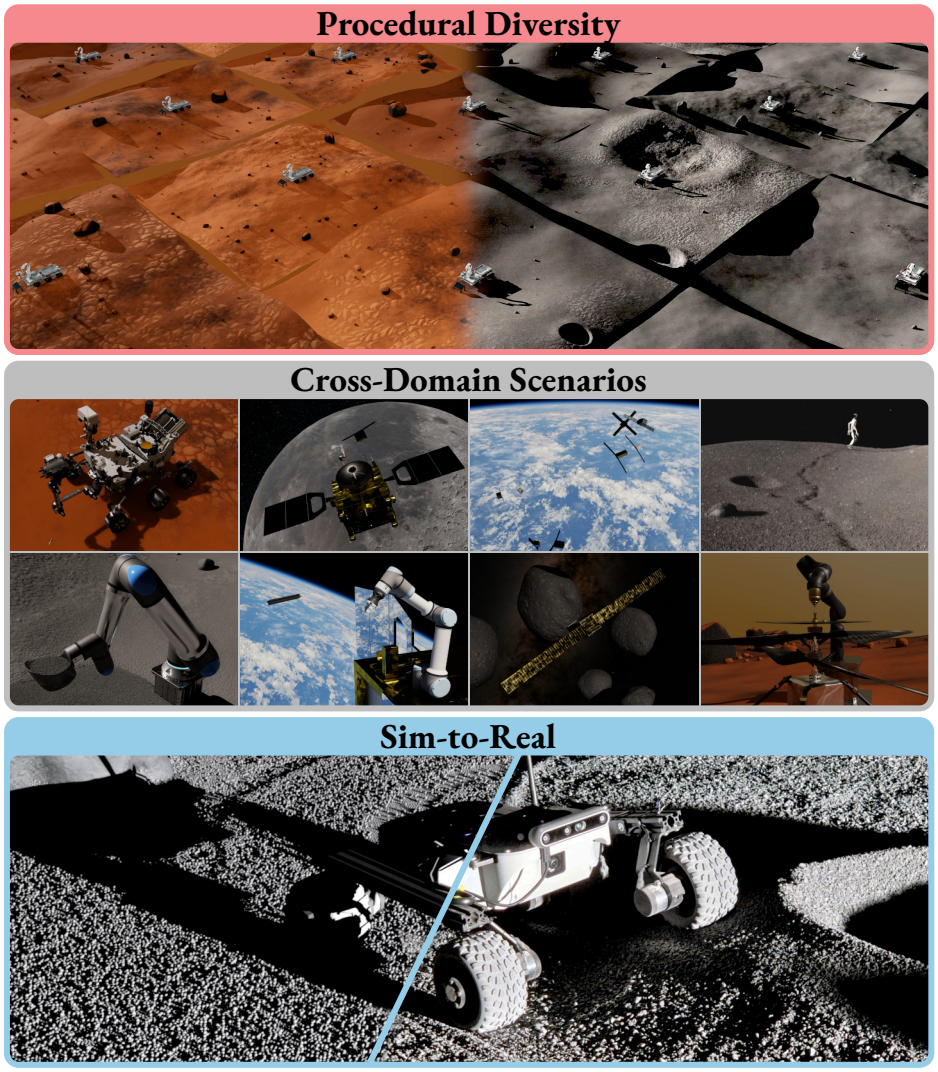
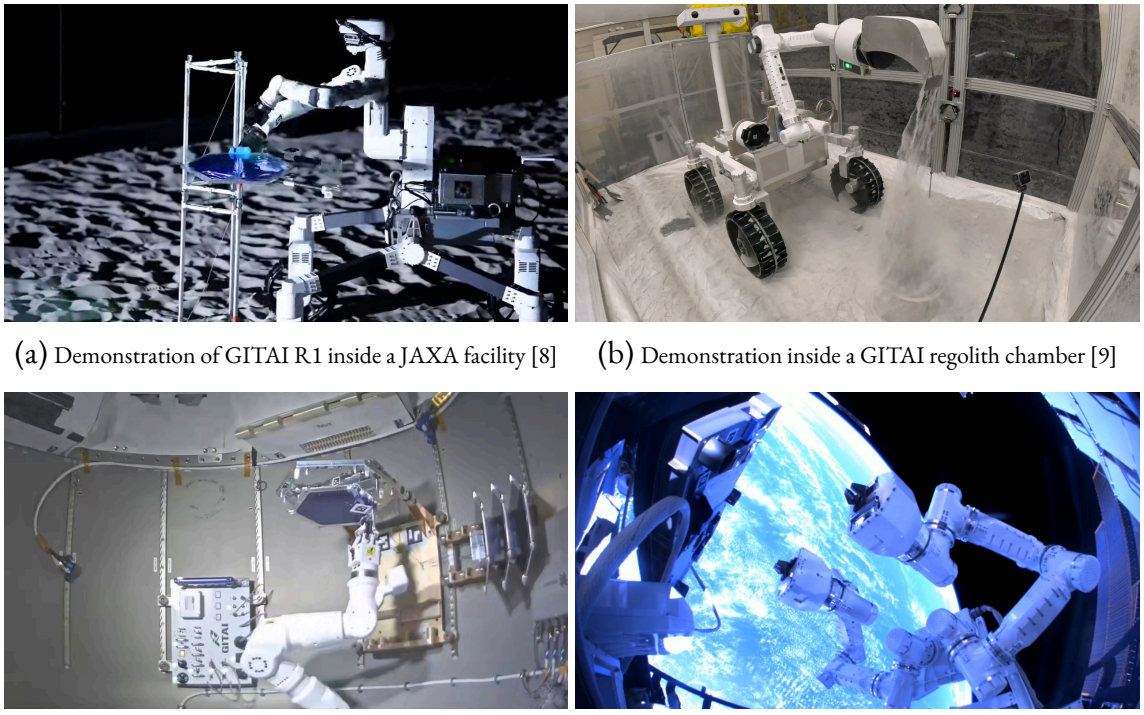


Figure 1.1 – The procedural paradigm at the core of this thesis leverages programmatically generated scenarios for large-scale policy training in a parallelized simulation, which is a methodology validated by its successful zero-shot sim-to-real transfer to a physical robot.

1.1 Motivation

The human ambition to explore and establish a sustainable presence beyond Earth is a powerful catalyst for technological innovation. Future space missions envisioned by agencies as well as commercial entities are predicated on complex operations conducted in environments far removed from direct human intervention. For instance, the scientific exploration of the Moon and Mars by ESA and NASA requires autonomous rovers capable of traversing vast, hazardous terrains to independently collect diverse samples and perform sophisticated in-situ analyses [6]. Concurrently, robotic manipulators are becoming indispensable for building and maintaining orbital infrastructure, including next-generation space stations and advanced telescopes [3], [4]. These systems will need to assemble large structures from smaller components, extend the operational lives of existing satellites through servicing [7], and tackle the escalating problem of space debris. The development and deployment of such systems is underway through technology demonstrations of robots like those developed by GITAI shown in Figure 1.2.

Furthermore, a cornerstone of sustainable presence is the ability to utilize the land through in-situ resource utilization (ISRU) [12]. This entails robots capable of excavating regolith, extracting water ice, and converting indigenous materials into vital resources for construction, propellant production, and life support. Ultimately, as our presence extends further into



(a) Demonstration of GITAI R1 inside a JAXA facility [8]

(b) Demonstration inside a GITAI regolith chamber [9]

(c) Demonstration of GITAI S1 inside the ISS [10]

(d) Demonstration of GITAI S2 outside the ISS [11]

Figure 1.2 – Technology demonstrations of GITAI for in-space operations that illustrate the manipulation skills required for both planetary and orbital domains.

space, robots will play a crucial role in supporting astronaut activities. They will assist with extravehicular activities (EVAs), undertake hazardous tasks, and perform routine maintenance to improve crew safety and mission productivity. These applications share a common, critical need for robots that can operate with minimal human supervision and adapt to novel situations. This thesis is deeply motivated by addressing this capability gap and advancing robot learning techniques tailored to the unique challenges of space robotics.

1.1.1 Rise of Robots Beyond Earth

The expansion of human endeavor into space depends critically on a new generation of robotic autonomy. The sheer scale of interplanetary missions introduces substantial communication delays that range from seconds for lunar operations to many minutes for Mars. This renders real-time teleoperation impractical for intricate tasks. Even within cislunar space, continuous human oversight is a scarce and valuable resource that limits the scope and pace of operations. Consequently, robots are no longer mere extensions of human capability. They are becoming essential and independent agents tasked with navigating and acting in these remote domains.

However, sending robots beyond Earth immerses them in environments far more demanding and less forgiving than any terrestrial setting. Extraterrestrial domains are often poorly characterized. They present highly unstructured terrains where unexpected events are the norm. These events can range from sudden micrometeoroid impacts to the gradual degradation of hardware. The success of pioneering missions like the Perseverance rover on Mars showcases the potential of robotic exploration. It also highlights current limitations because its operations are constrained by meticulous planning cycles and delayed communication [2]. Robots in these conditions must therefore possess the intrinsic ability to perceive their surroundings, make intelligent decisions in the face of uncertainty, and adapt their strategies without constant human guidance.

These operational demands are compounded by a unique combination of environmental extremes. Robots must endure drastic temperature fluctuations and persistent cosmic radiation. Orbital environments introduce the complexities of vacuum and microgravity, which fundamentally alter object dynamics and heat dissipation. Planetary surfaces present different gravity magnitudes that influence locomotion and manipulation in non-intuitive ways. Abrasive regolith and rock fragments can damage mechanical systems and alter interaction dynamics, as shown in Figure 1.3. The safety-critical nature of space missions, where failures can lead to catastrophic losses, further underscores the need for exceptionally robust systems. All of this must be achieved under severe constraints on available energy, computational power, and communication bandwidth.

Figure 1.3 – Damage to the wheel of the NASA Curiosity rover on Mars. This image highlights the challenges posed by hazardous planetary terrains, where sharp rocks can cause significant hardware degradation over time [13].

1.1.2 Versatility of General-Purpose Robotics

The dynamic and evolving demands of future space missions make the deployment of highly specialized, single-task robots both logistically and economically unfeasible. In a prospective lunar base, a single robotic system may be tasked with excavating regolith, assembling habitat modules, and later deploying scientific instruments. Developing, launching, and maintaining a distinct robot for each activity would introduce prohibitive complexity and cost. This operational reality gives rise to modular robotic architectures. An example is a common mobile platform equipped with a manipulator and various end-effectors (EEs).

This hardware versatility exposes a more profound software challenge. A physically adaptable robot remains functionally inert if its control system cannot accommodate a new tool or an unfamiliar task. Manually engineering control policies for every hardware configuration and mission objective is a brittle and unscalable strategy. The key to unlocking true versatility lies not in pre-programming a robot for one specific mission but in equipping it with a learned set of fundamental, transferable skills. Instead of a monolithic policy for a complete construction project, a truly general-purpose robot must master foundational primitives such as reliable navigation, grasping, and precision insertion. This approach abstracts a complex problem to its underlying physical components. For instance, a learned peg-in-hole sequence is a foundational motion applicable to structural assembly, spacecraft docking, and electrical connector mating.

Achieving this level of transferability, as depicted in Figure 1.4, requires that learned skills are fundamentally agnostic to hardware specifics. The methodology in this thesis pursues this goal by learning control policies in the robot's task space. For instance, the high-level policy would command EE motion rather than specific joint torques. This ensures a policy can be deployed across manipulators with different kinematic structures. By leveraging standardized or abstract sensory representations, the system also becomes invariant to the precise type or placement of its sensors. This approach transforms a general-purpose robot from a mere collection of reconfigurable parts into a truly adaptable agent capable of applying its learned knowledge to novel challenges. This adaptable intelligence is the cornerstone of sustainable and scalable in-space operations. The pursuit of generalization does not exclude the need for specialization. Safety-critical sequences like a planetary landing will likely continue to depend on platform-specific policies. Yet, the vision of general-purpose versatility is to build a broad foundation of composable skills to solve the vast majority of tasks encountered during a mission.

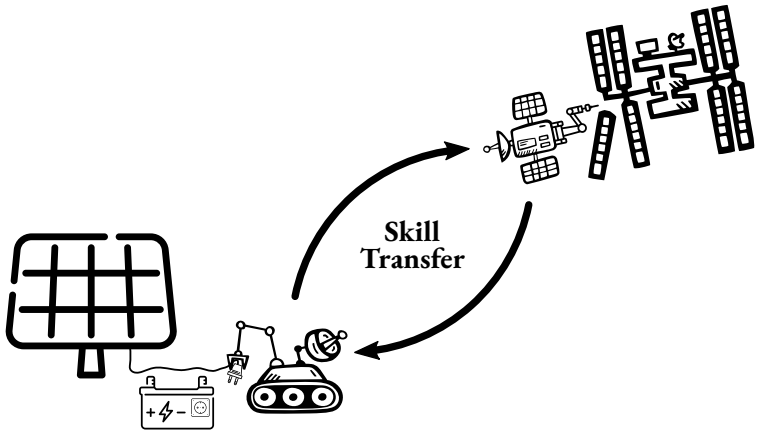


Figure 1.4 – The conceptual idea of acquiring skills that can be transferred among different application domains of space robotics. A single learned skill, such as peg-in-hole insertion, could be deployed to diverse tasks across planetary and orbital environments, enabling a general-purpose robot to adapt to various mission objectives.

1.1.3 Promise of Learning-Based Autonomy

The quest for versatile and truly autonomous space robots necessitates a departure from traditional control paradigms that depend on precise environmental models and pre-programmed behaviors. These methods inherently struggle when confronted with the unmodeled dynamics and fundamental uncertainties of extraterrestrial operations. Robot learning offers a transformative pathway to provide robots with the adaptive intelligence required for these challenges [14]. Among the available paradigms, reinforcement learning (RL) has gained popularity due to its suitability for solving sequential decision-making problems [15]. By learning from direct interaction with its surroundings, an RL agent can acquire a sophisticated understanding of physics and task dynamics.

This process embodies two capabilities that are critical for space missions. First, RL agents can discover non-intuitive yet effective strategies for interacting with their environment. An example is learning to use environmental contacts to guide a misaligned part into place. These emergent behaviors are often difficult, if not impossible, for a human to explicitly program. Second, and central to the promise of this thesis, is achieving generalization. When trained across thousands of diverse simulation instances, an RL policy should become invariant to irrelevant details such as the exact shape of a rock or the intensity of ambient illumination, and instead distill the underlying principles of the task. Although the application of advanced learning techniques to space robotics is still limited, their success in tackling complex terrestrial problems and even super-human challenges signals their profound potential [16], [17]. This thesis argues that the synergy of modern RL algorithms with a comprehensive and diverse simulation framework is the key to transforming this promise into demonstrable, reliable autonomy for the next generation of space exploration.

1.1.4 Role of High-Fidelity Simulation

The entire promise of robot learning for space is contingent upon high-fidelity simulation as a single enabling technology. The extreme cost, safety-critical nature, and logistical impossibility of conducting the millions of trials required for modern robot learning make the virtual world the only feasible training ground. Terrestrial analog facilities, while valuable, cannot fully replicate the unique physics of different gravitational fields or the vast topographies of other worlds. Simulation provides the necessary safe, cost-effective, and scalable environment to develop and validate the next generation of autonomous systems.

The success of this paradigm relies on bridging the critical gap between the virtual and real worlds. This is not a simple sim-to-real problem but a more complex sim-to-lab-to-space challenge. A policy must first prove its efficacy by transferring from a virtual environment to a physical robot in a laboratory. It must then be robust enough to generalize to the far more unstructured conditions of its final extraterrestrial deployment. This dual challenge exposes the inadequacy of many existing space-grade simulators for robot learning. Traditionally, these tools were designed for mission verification, with the aim of testing a pre-determined plan under specific, nominal conditions [18], [19]. They were not built for the open-ended discovery and generalization that RL requires. To train a truly adaptive agent, a simulation must be more than a mere digital twin of a single scenario. It must become a universe of possibilities. It must expose the agent to a near-infinite variety of environmental conditions, object configurations, and potential failure modes. The lack of a simulation framework that integrates these learning-centric principles with high-fidelity space physics represents a major bottleneck to progress. Addressing this critical void is the primary focus of the research presented in this thesis.

1.2 Problem Statement

While the motivation for autonomous space robotics is compelling, its realization is obstructed by a set of fundamental and interconnected challenges. The successful deployment of learning-based systems beyond Earth is not just an engineering effort. It is a scientific problem that lies at the intersection of generalization, standardization, and safety. This thesis is constructed to directly address the following critical problem areas that currently impede progress in the field.

1.2.1 Challenge of Generalization

The central technical problem confronting robot learning in space is the profound challenge of generalization. An autonomous agent must perform reliably not just in the specific conditions under which it was trained. It must perform across the full, unpredictable spectrum of scenarios it will encounter during a mission. Extraterrestrial environments represent the ultimate out-of-distribution test case. A rover trained on simulated Martian terrain must contend with real-world regolith whose mechanical properties differ from any training sample. A manipulator arm learning to assemble a structure must handle components whose dimensions may have subtly changed due to thermal expansion or launch-induced vibrations.

This demand for robustness is where many contemporary robot learning approaches falter. Policies trained on limited or static datasets, even those of high fidelity, tend to overfit to the particularities of their training environment [20]. They may learn to exploit subtle visual or physical artifacts in the simulation that do not exist in reality. Consequently, when deployed, such policies often exhibit brittle behavior and fail catastrophically in the face of even minor novelty [21]. This discrepancy between performance in training and performance in operation reveals a critical generalization gap. Closing this gap is the foremost challenge. A policy that cannot generalize beyond its training data is fundamentally unsuitable for the unstructured and evolving nature of space.

1.2.2 Lack of Standardized Benchmarks

The challenge of generalization is exacerbated by a significant infrastructural problem, namely the absence of standardized benchmarks tailored for robot learning in space. Scientific progress in fields like computer vision and RL is historically driven by the availability of common testbeds that allow researchers to rigorously compare methodologies, reproduce results, and build upon each other's work. The domain of space robotics currently lacks such a unifying platform.

Existing space simulators like GMAT [18] and Basilisk [19] focus primarily on astrodynamics while being typically employed for mission verification and validation (V&V) of pre-planned

trajectories. They are often proprietary, narrow in scope, and ill-suited for the unique demands of robot learning research, which requires thousands of diverse and randomized simulations in parallel. Recent efforts have produced learning-focused simulators for specific tasks like rover navigation [22] or spacecraft rendezvous [23], but a comprehensive platform is missing. Conversely, established robot learning benchmarks, such as RLBench [24], Meta-World [25], FurnitureBench [26], and ManiSkill3 [27], are overwhelmingly terrestrial. They focus on tabletop manipulation or indoor navigation tasks. Their underlying physics and operational constraints do not capture the complexities of variable gravity, orbital dynamics, or large-scale unstructured terrains. This lack of a suitable and accessible testbed raises the barrier to entry for researchers and stifles innovation. Without a common ground for evaluation, the community cannot effectively measure progress toward achieving robust, generalizable autonomy.

1.2.3 Safety-Critical Nature of Space Operations

Finally, all solutions must be developed under the immense pressure of the safety-critical nature of space operations. Robotic systems in space are multi-million or even multi-billion dollar assets, and mission failure is not an acceptable outcome. A policy that is not robust or generalizable is inherently unsafe. An autonomous system that executes unpredictable, jerky, or overly rigid motions poses a direct threat to itself, to other mission-critical hardware, and potentially to human astronauts.

This problem is particularly acute for the contact-rich manipulation tasks that are central to this thesis, such as assembly and excavation. Traditional kinematic controllers like inverse kinematics (IK) are often brittle when unexpected contact occurs, potentially generating large forces that damage the robot or its environment. True safety in interaction requires not just positional accuracy but also physical compliance. The robot must be able to gracefully yield to unmodeled forces and adapt its physical behavior in response to contact. The problem, therefore, extends beyond mere task success. It is a question of how to learn policies that are not only effective but also inherently smooth, stable, and compliant, thereby ensuring the safety and integrity of the entire mission.

1.3 Research Questions

The preceding problem statement gives rise to a set of fundamental research questions that this thesis aims to answer. These questions are designed to systematically deconstruct the overarching challenge of achieving adaptive autonomy in space. They address the issues of generalization, learning methodology, and algorithmic limitations in a structured manner. The investigation of these questions forms the core intellectual pursuit of this work.

Research Question 1

How can simulation environments be leveraged to effectively train robotic policies that generalize across the diverse and unpredictable conditions of space?

At the heart of this thesis lies the challenge of generalization. Training a robot directly in space is infeasible, so simulation is the only viable alternative. This question explores the very nature of the simulation required for this purpose, moving beyond the idea of a single, high-fidelity digital twin. Instead, it investigates the construction of a virtual training ground that actively fosters robust and adaptable policies. The central hypothesis is that exposure to immense diversity is the key to generalization. This leads to an inquiry into the most effective techniques for generating this diversity, focusing on a combination of procedural content generation (PCG) for creating a near-infinite variety of physical assets [28] and extensive domain randomization (DR) of physical and visual parameters [29]. By training an agent within this constantly shifting world, the aim is to force it to learn the underlying principles of a task, rather than to overfit to the superficial details of any single scenario.

Research Question 2

What learning methodologies and control representations unlock the adaptive and compliant behaviors necessary for complex operations in space?

This question addresses the learning process itself. It probes which algorithmic and control-theoretic choices are best suited to producing the kind of behavior needed for space robotics. The inquiry compares different RL paradigms, particularly model-free approaches like Proximal Policy Optimization (PPO) [30], Twin Delayed Deep Deterministic Policy Gradient (TD3) [31], and Soft Actor-Critic (SAC) [32] with model-based RL (MBRL) approaches like DreamerV3 [33], to determine which offers superior sample efficiency and generalization capabilities. The question then delves into the representation of actions, challenging the sufficiency of standard IK for contact-rich tasks. It specifically explores operational space control (OSC) as a framework for providing software-defined compliance [34]. The core of this inquiry is to determine whether an agent can learn not only where to move but also how to physically interact with its environment by dynamically modulating its own stiffness and damping. The goal is to discover a synergistic combination of learning algorithm and control representation that yields policies that are not just successful, but also inherently adaptive and compliant.

Research Question 3

What are the key failure modes and limitations of state-of-the-art robot learning algorithms when faced with the unstructured complexities of space-relevant scenarios?

This final question adopts a critical and empirical perspective, seeking to systematically identify the breaking points of current state-of-the-art techniques. By developing a standardized benchmark, this thesis provides the means to rigorously probe the capabilities of existing algorithms. The investigation aims to answer practical questions: How does performance degrade with increasing environmental complexity? What is the performance gap between learning from privileged state compared to raw pixel data? At what point do long-horizon tasks with sparse rewards become intractable? By identifying these failure modes, this research aims to provide a clear assessment of where the current state-of-the-art stands, establishing a concrete roadmap for future work.

1.4 Research Objectives

To answer the stated research questions, this thesis pursues a set of specific and actionable research objectives. These objectives form a structured plan for the development, implementation, and validation of the concepts explored in this work.

Research Objective 1

Design and implement an open-source simulation framework for robot learning in space.

The first objective is to create the necessary infrastructure for this research. This involves the design and implementation of a novel, open-source simulation framework built to address the limitations of existing tools. The framework must be highly parallelizable, integrate PCG and DR as core features, and provide standardized interfaces like the Gymnasium API [35] and Robot Operating System (ROS) 2 [36] to make it accessible to the broader research community.

Research Objective 2

Establish a standardized suite of benchmark tasks within the developed simulation framework to rigorously evaluate the generalization and adaptation capabilities of robot learning algorithms in space-relevant scenarios.

Building upon the core framework, the second objective is to populate it with a suite of meaningful and challenging benchmark tasks. This involves carefully designing tasks that test the specific capabilities required for future space missions, from mobility and manipulation to complex, long-horizon assembly. The objective is to create a standardized testbed that allows for the systematic evaluation of learning algorithms to quantify performance, measure generalization, and identify failure modes.

Research Objective 3

Investigate and develop a learning methodology for robust adaptive control.

With the framework and benchmark in place, the third objective is to develop a learning methodology capable of producing adaptive and compliant behaviors. This involves a comparative analysis of RL algorithms and an exploration of control representations suited for contact-rich interaction. The goal is to formulate a blueprint for learning adaptive control that combines the strengths of world-modeling through MBRL agents with the stability and expressiveness of learned compliant control via OSC.

Research Objective 4

Validate the proposed framework and methodology via sim-to-real transfer in a terrestrial analogue facility.

The final objective is to bridge the gap between simulation and physical reality. This entails validating the key principles and learned policies on physical robotic hardware within specialized terrestrial analogue facilities, such as the LunaLab [37]. The goal is to demonstrate successful zero-shot transfer from the simulation framework to real robots. This will provide a validation of the entire approach and its potential applicability to future space missions.

1.5 Key Contributions

This thesis makes several significant contributions to the fields of robot learning and space robotics. These contributions are organized into primary and secondary categories. The primary contributions represent the core, novel advancements that form the central pillars of this work.

1.5.1 Primary Contributions

The research presented in this thesis yields contributions that directly address the stated research objectives and provide novel solutions to the core problems identified.

The foundation of this work was laid through **Publication I**, which introduced several novel concepts for robot learning in space. This research explored a simulation-centric approach to learn an end-to-end RL policy from high-dimensional visual inputs and established the feasibility of zero-shot transfer to a physical robot. This initial work demonstrated the potential of procedural asset generation and highlighted the importance of PCG for enhancing the generalization capabilities of learned policies.

Following this, the systematic investigation of PCG and DR across multiple RL algorithms in **Publication II** demonstrated their effectiveness in enhancing generalization. The findings provide strong empirical evidence that exposure to a wide distribution of training data is a critical strategy for closing the generalization gap.

The primary artifact of this thesis is the Space Robotics Bench (SRB), first conceptualized in **Publication III** before being fully introduced in **Publication VII**. It is a comprehensive open-source simulation framework designed specifically for robot learning in space. SRB provides a standardized and highly configurable platform that integrates high-fidelity physics with extensive DR and streamlined PCG pipelines, as proposed in **Publication IV**. Its sim-to-real capabilities were further validated in **Publication V** by demonstrating the successful transfer of learned policies to a rover operating in a dusty lunar-analogue facility.

A key methodological contribution is the development of a novel learning methodology for adaptive compliant control, as detailed in **Publication VI**. This methodology integrates MBRL with OSC to enable robots to learn to dynamically adjust their physical interaction properties. The work demonstrates that an agent can learn to modulate its own stiffness and damping in response to environmental interactions, which is essential for safe and reliable operations in space.

1.5.2 Secondary Contributions

In the course of pursuing the primary objectives, this research also produced several valuable secondary contributions. Leveraging the SRB framework, a system for autonomous visual inspection of orbital targets was developed in a collaborative effort presented in **Publication XI**. This application served as a powerful external validation of the framework's utility and modularity, demonstrating that the core MBRL methodology could be effectively applied by other researchers to control a spacecraft for complex 6-degree of freedom (DoF) maneuvers around larger targets under microgravity.

Participation in the Robot Air Hockey Challenge 2023 resulted in a novel application of a MBRL algorithm to a highly dynamic and competitive task. The approach leveraged self-play to acquire a robust policy and ultimately achieved second place, followed by a successful sim-to-real deployment. A retrospective analysis of all solutions is presented in **Publication X**.

Beyond these applications, this thesis provides a broad contribution to the research community by establishing extensive algorithmic benchmarks. Although minuscule in comparison to large-scale terrestrial datasets [38], the evaluation performed on SRB tasks represents one of the largest studies of RL algorithms on space-relevant problems to date, highlighting the strengths and weaknesses of different learning paradigms. Finally, the research is underpinned by significant open-source contributions. These include the SRB framework itself with its hundreds

of unique assets and robot configurations, framework-agnostic SimForge engine for PCG, and various utilities for robotics and RL research.

1.6 Thesis Outline

The remainder of this thesis is structured to systematically build upon the motivation and problems articulated in this introduction. The chapters are organized to first establish the necessary background, then present the core contributions, and finally, provide a thorough validation and discussion of the results. The logical progression of these core themes is illustrated in Figure 1.5.

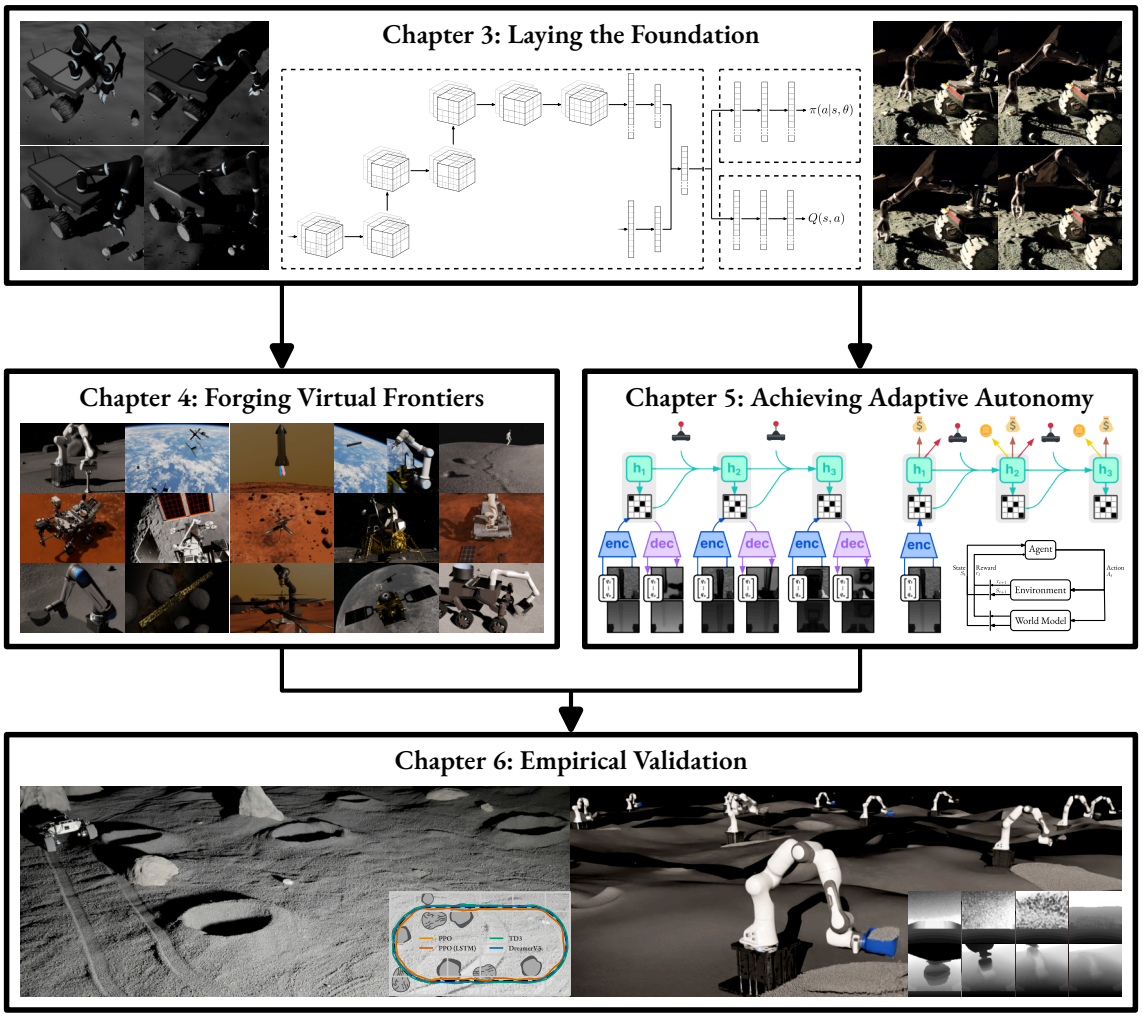


Figure 1.5 – A visual representation of the thesis structure, mapping the main chapters to their core themes and showing the progression from foundational work to the final conclusions.

A comprehensive review of the related work and foundational concepts that underpin this thesis is provided in Chapter 2. It covers the domain of space robotics, the principles of robot control, the paradigm of robot learning with a focus on RL and the sim-to-real challenge, and the existing ecosystem of robotics simulation. This chapter concludes by synthesizing this information to precisely define the research gap that this work addresses.

Chapter 3 details the initial investigation from **Publication I** that laid the groundwork for the main contributions of this thesis. It describes a simulation-centric approach to learning robotic grasping on the Moon and establishes the core concepts of PCG and the use of 3D observations. The chapter recounts the successful sim-to-real transfer of a learned policy to a physical robot in a terrestrial analogue facility. These outcomes served as the foundation for the subsequent research, providing critical insights that highlighted the limitations of existing infrastructure and set the stage for the development of a more comprehensive framework.

SRB, the core artifact of this thesis, is presented in Chapter 4. This chapter offers a deep dive into the design philosophy, architecture, and capabilities of the open-source simulation framework. It details the suite of benchmark tasks, the diverse robotic fleet, the procedural generation engine SimForge, and the integrations with the broader robotics and machine learning ecosystems.

Chapter 5 presents the methodological contribution, revolving around a novel learning methodology for achieving adaptive autonomy. This chapter details the investigation into different RL paradigms and control representations. It presents the algorithmic baselines, demonstrates the benefits of world modeling, and introduces the framework for learning compliant manipulation. It also explores the influence of robot embodiment and tool-awareness on learned strategies.

The empirical validation for the proposed framework and methodology is provided in Chapter 6. This chapter presents a series of in-depth case studies on key applications, including adaptive traversal and tool-aware regolith excavation, to provide robust evidence of the generalization and adaptation capabilities of the developed agents.

Finally, Chapter 7 summarizes the findings of this research. It revisits the research questions and objectives outlined in this chapter and demonstrates how they have been addressed and fulfilled. The chapter discusses the broader implications of this work, acknowledges its limitations, and proposes promising directions for future research.

Throughout the chapters, this thesis synthesizes a body of research that has been presented in a series of peer-reviewed publications. To maintain a clear and cohesive narrative focused on the central scientific arguments, the following content presents the primary methodologies and principal findings of this work. Consequently, certain implementation details, such as exhaustive hyperparameter lists, minor variations in experimental setups, and supplementary results, have been omitted for brevity. For a complete and detailed account of any specific study, the reader is respectfully directed to the original publication.

2

Background and Related Work

This chapter provides the foundational context necessary to situate the contributions of this thesis within the broader landscape of space robotics and robot learning. It begins by exploring the domain of space robotics, tracing its historical evolution and outlining the unique challenges that motivate the need for advanced autonomy. Next, it reviews the fundamental principles of robot control for both mobility and manipulation, with a particular focus on the distinction between kinematic and compliant control strategies. The chapter then delves into the paradigm of robot learning. It introduces the mathematical framework of the Markov decision process (MDP) and discusses the core concepts of RL, learning from demonstration (LfD), and the critical sim-to-real challenge. Finally, it surveys the ecosystem of robotics simulation and highlights the role of procedural generation and standardized benchmarks in driving progress. The chapter culminates in a synthesis that clearly identifies the research gap this thesis aims to fill.

2.1 Domain of Space Robotics

Space robotics is a specialized field of robotics concerned with the design, development, and operation of robotic systems capable of functioning in extraterrestrial environments. These systems are instrumental in performing tasks that are too dangerous, repetitive, or precise for humans. They also operate in locations that are entirely inaccessible to direct human presence. From the earliest automated probes to the sophisticated rovers currently exploring Mars, robots have been fundamental to humanity's quest to understand and venture into the cosmos.

2.1.1 Evolution of Automation in Space

The history of space robotics is deeply intertwined with the history of space exploration itself. The first robotic systems in space were simple, automated spacecraft that followed pre-programmed trajectories and executed simple commands [39]. Manipulation tasks were

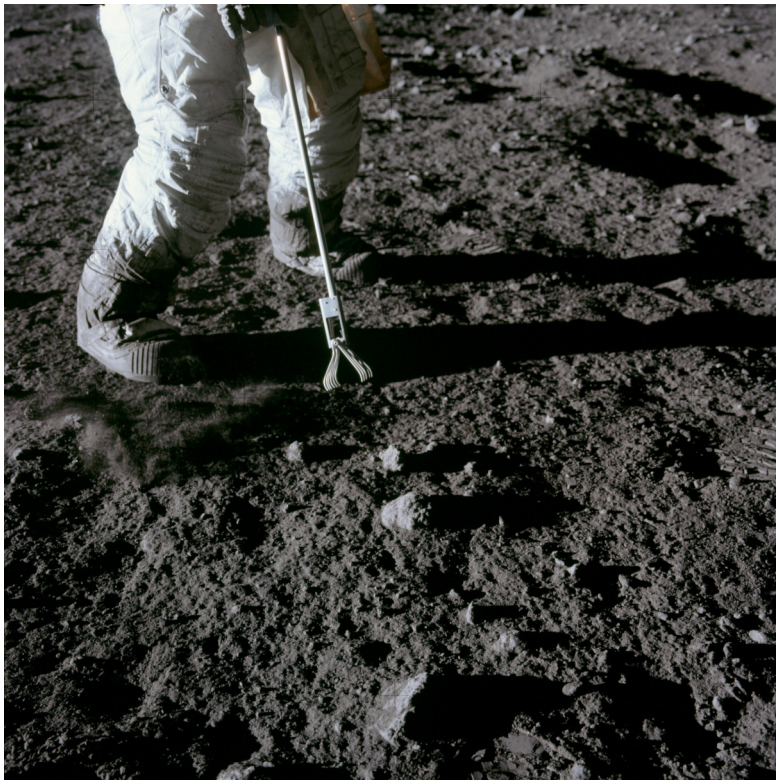


Figure 2.1 – A photograph of the Apollo 12 mission displaying the use of a manual tool for collecting lunar rock samples [42].

initially performed by human astronauts using hand tools, as illustrated in Figure 2.1. These early missions relied entirely on human dexterity for complex interactions, highlighting the capability gap that modern autonomous robotics aims to fill. Over the decades, the level of automation has steadily increased. The Viking landers featured a robotic arm capable of scooping Martian soil [40]. The actions of this arm were meticulously sequenced and uploaded from Earth. The Sojourner rover represented a significant step forward with its limited ability to autonomously navigate around obstacles [41].

More recent missions, such as the Spirit, Opportunity, and Curiosity rovers, have demonstrated progressively more sophisticated autonomous navigation and instrument placement capabilities [43]. The Perseverance rover, currently active on Mars, leverages advanced vision-based navigation to traverse challenging terrain more quickly than its predecessors [2]. In orbit, the Canadarm on the Space Shuttle and its successor, Canadarm2 on the ISS, have been crucial for satellite deployment, station assembly, and supporting EVAs. These systems, however, still rely heavily on human operators to guide their actions. The evolution has been one of gradual delegation. It moved from pre-programmed sequences to teleoperated control, and finally to supervised autonomy, where high-level goals are commanded from Earth [5]. The next logical step in this evolution is the transition to truly adaptive autonomy. This transition is the central focus of this thesis.

2.1.2 Unique Challenges of Extraterrestrial Environments

The environments beyond Earth present a set of challenges that are fundamentally different and more severe than those encountered in any terrestrial setting. These challenges affect every aspect of a robot's design and operation. Communication latency is a primary driver for autonomy. The round-trip communication latency to the Moon is a few seconds. This is manageable for high-level supervision but disruptive for direct control. However, the latency increases significantly with distance, which renders real-time teleoperation impossible for any intricate task. This delay eliminates the possibility for an Earth-based operator to perform reactive control. An unexpected event, such as a manipulator encountering an obstacle or a tool getting stuck, would require an immediate response that cannot wait for a signal from Earth.

The physical environment itself is hostile. Extreme temperature variations between illuminated and shadowed areas can cause thermal stresses on materials and electronics. Persistent exposure to cosmic and solar radiation can degrade components over time, leading to unexpected hardware failures [44]. This necessitates the use of radiation-hardened electronics, which are often generations behind their terrestrial counterparts in computational power. This creates a challenging paradox. The need for greater autonomy demands more onboard computation, while the environment restricts the available computational resources.

In orbital environments, the vacuum affects heat dissipation and can lead to phenomena like cold welding. Microgravity fundamentally alters the dynamics of all physical interactions. An object that is released does not fall but drifts, and any force exerted by a manipulator will produce an equal and opposite reaction on its spacecraft base. As illustrated in Figure 2.2, planetary surfaces are often covered in fine, abrasive regolith that can infiltrate mechanical joints,

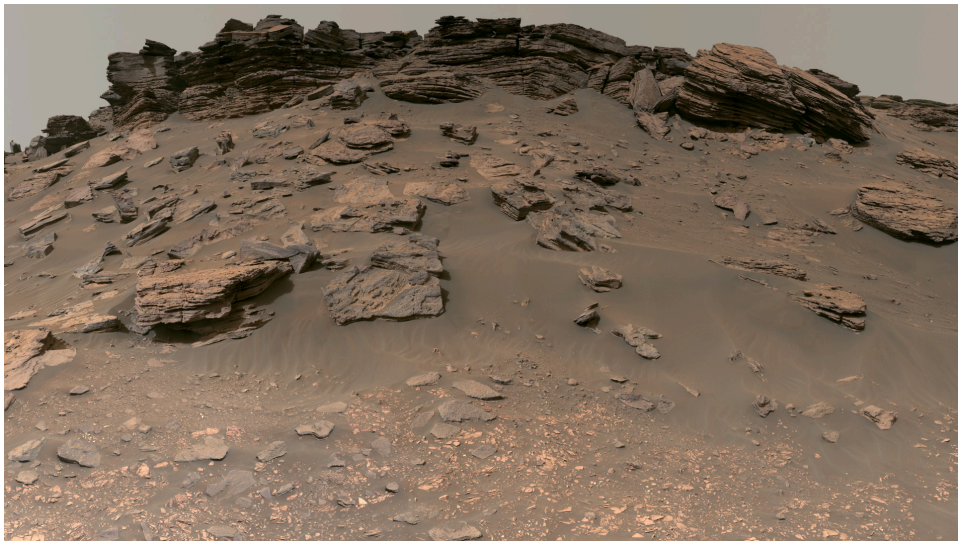


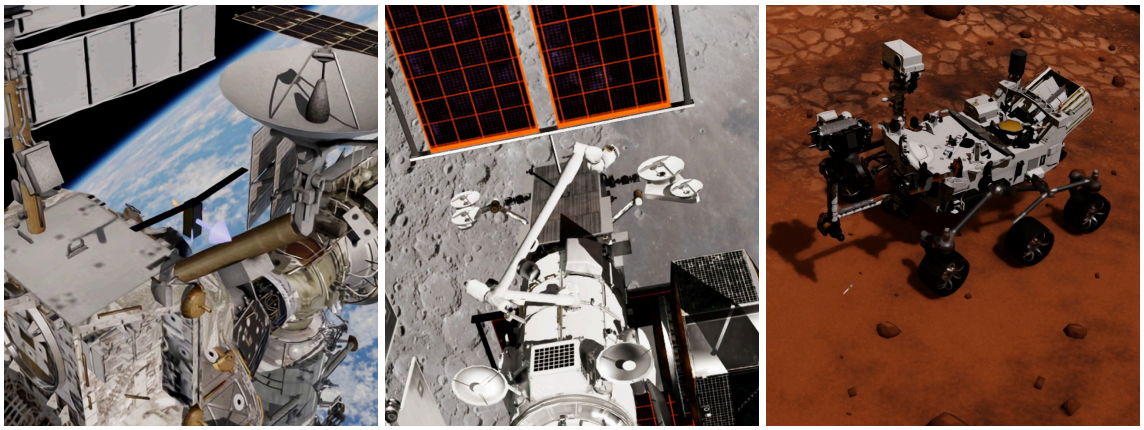
Figure 2.2 – The rugged landscape of Mars' Jezero Crater delta, where rovers must navigate through a hazardous mix of steep inclines, loose regolith, and boulder fields [45].

obscure sensors, and create unpredictable traction conditions for rovers. Furthermore, these environments are largely unstructured and poorly mapped. A robot must constantly contend with novelty and uncertainty in a domain where every component is a safety-critical asset.

2.1.3 Vision for Future Robotic Missions

The ambition of future space missions necessitates a leap in robotic capabilities. The vision extends far beyond simple exploration to encompass large-scale construction, industrial-style resource utilization, and permanent human habitation. Initiatives like NASA's Artemis program aim to establish a sustainable human presence on the Moon. This goal relies on robots to perform preparatory work such as site surveying, habitat construction, and infrastructure deployment before the arrival of astronauts [1]. A cornerstone of this vision is ISRU [12], where robots will excavate regolith to extract resources like water ice and minerals. These resources can then be used for construction, propellant production, and life support.

In orbit, the concept of in-space servicing, assembly, and manufacturing (ISAM) is gaining traction [4]. This involves robots that can refuel and repair existing satellites to extend their operational lifespan, assemble large structures like telescopes or solar power stations from components launched separately, and even manufacture parts directly in space [3], [7]. For deep space exploration, a new generation of autonomous science laboratories on wheels or propellers will be needed to explore ocean worlds or methane lakes, where communication is severely limited, and mission risk is high [46]. The complexity of these missions may require heterogeneous teams of robots working collaboratively to achieve common goals [47]. In all these envisioned futures, as depicted in Figure 2.3, robots are not just tools but are foundational infrastructure. They will work persistently and adaptively to create and maintain a human foothold beyond Earth. This vision can only be realized if these robotic systems are endowed with a level of autonomy that far surpasses the state of the art today.



(a) Orbital inspection (b) Autonomous manipulation (c) Planetary exploration

Figure 2.3 – The vision for future robotic missions across multiple domains.

2.2 Foundation of Robot Control

The ability of a robot to perform useful work in its environment is predicated on its capacity for controlled motion. Robot control is the discipline concerned with calculating the necessary actuator commands to achieve a desired state or behavior. This section provides a brief overview of the fundamental control concepts relevant to this thesis. It focuses on the distinct but often intertwined challenges of mobility and manipulation.

2.2.1 Mobility

Mobility refers to a robot's ability to move its entire body through an environment. For space robotics, this encompasses a wide range of platforms with vastly different control requirements that are heavily influenced by the operational domain. The approaches to achieving stable and efficient locomotion on a planetary surface are fundamentally different from those required for maneuvering in the vacuum of space.

Planetary mobility presents challenges of traction, stability, and navigation over unstructured terrain. The control of wheeled rovers often involves kinematic models like skid-steer or Ackermann steering. These models map desired chassis velocities to individual wheel speeds. Their effectiveness is complicated by the interaction with deformable terrain, a field known as terramechanics [48]. The unpredictable nature of wheel slip on loose regolith makes precise odometry and path following a persistent challenge. Legged robots, such as quadrupeds or humanoids, offer the potential for greater mobility in extreme terrain but introduce a more complex control problem. Due to their higher dimensionality and the need to maintain dynamic stability, their control is typically hierarchical. A high-level planner determines footstep locations while a low-level controller calculates the joint torques or positions required to execute stable locomotion [49]. A third mode of planetary mobility is aerial. The Ingenuity helicopter on Mars demonstrated the feasibility of flight in a thin atmosphere. Its control involves precisely modulating rotor speeds to generate the necessary lift and thrust for controlled flight, which offers a unique vantage point for reconnaissance and exploration.

In orbital environments, mobility is governed by orbital mechanics and the principles of rocket propulsion. Spacecraft control involves the precise firing of thrusters to achieve a desired change in velocity. This is used for fundamental operations like station-keeping, trajectory correction, and orbital insertion, while more complex operations include rendezvous and docking. These tasks require an agent to maneuver a spacecraft to approach and precisely match the full 6-DoF state in the SE(3) of a target object. The challenge is magnified when the target is uncooperative, such as a piece of tumbling debris, as this requires the chasing spacecraft to predict its motion and execute a complex intercept trajectory. In all orbital maneuvers, the control problem is constrained by fuel optimization, as propellant is a finite and mission-critical resource. Across

all these diverse platforms, the objective of the mobility controller is to accurately track a desired trajectory for the robot's base while navigating the specific physical constraints of its domain.

2.2.2 Manipulation

Manipulation involves the control of a robotic arm and its EE to interact with objects in the environment. This is the primary focus of the complex tasks addressed in this thesis, such as assembly and excavation. The core problem of manipulator control is to move the EE to a desired pose in space by actuating the manipulator's joints. The approaches for solving this problem can be broadly categorized into kinematic and compliant control. A quintessential example of a contact-rich manipulation task is the peg-in-hole problem, illustrated in Figure 2.4. Its apparent simplicity contrasts with a complex control challenge that is fundamental to nearly all assembly operations. It has been a subject of extensive research for decades [50], [51], [52].

Figure 2.4 – The peg-in-hole task is a common problem in robotics, fundamental to a vast range of operations from industrial manufacturing to everyday interactions like connector inserting. Its successful execution requires precise control over contact forces and alignment.

Kinematic Control

Kinematic control focuses on the geometry of motion without considering the forces or torques that cause it. The most common problem in this domain is IK. Given a desired EE pose, IK calculates the corresponding set of joint angles required to achieve that pose. For manipulators with many DoF, there can be multiple or even infinite solutions to this problem.

A common approach for real-time control is differential IK, which relates EE velocities to joint velocities through a matrix known as the Jacobian [53]. By inverting or taking the pseudo-inverse of the Jacobian, a controller can compute the necessary joint velocities to achieve a desired EE velocity. This method forms the basis for many standard robot control interfaces. However, a key limitation of purely kinematic control is its rigidity. It commands the robot to achieve a specific position or velocity without regard for external forces. If the robot's path is obstructed or it makes an unexpected contact, a rigid kinematic controller will continue to drive the motors. This can potentially generate large forces that could damage the robot or its environment. This brittleness makes it ill-suited for tasks involving contact-rich interactions in unstructured settings, a limitation that became evident in the foundational study of this thesis **Publication I**.

Compliant Control

Compliant control addresses the limitations of rigid kinematic control by allowing a robot to safely interact with its environment. Instead of commanding a strict position, a compliant controller specifies a relationship between the robot's position and the external forces it experiences. This enables the robot to adjust its control when it encounters resistance. This mimics the natural compliance of human motion.

One powerful framework for implementing compliant behavior is OSC [34], which formulates the control problem directly in the robot's task space. It allows the designer to specify the desired dynamic behavior of the EE as if it were a mass-spring-damper system. By setting the controller gains for stiffness and damping, one can define how the EE should react to external forces. Low stiffness allows the robot to be very compliant and easily moved by external contacts, whereas high stiffness makes it behave more rigidly. OSC provides a principled way to manage forces during interaction. This makes it highly suitable for contact-rich tasks like assembly and excavation. As explored in **Publication VI**, this thesis demonstrates that the ability to dynamically modulate these compliance parameters through MBRL provides a powerful mechanism for achieving adaptive and robust manipulation [54], [55].

2.3 Paradigm of Robot Learning

Robot learning is a subfield of machine learning that focuses on endowing robots with the ability to acquire new skills and adapt their behavior through experience [14]. It represents a fundamental departure from traditional programming, where a robot's behavior is explicitly defined by a human engineer. Instead of following a fixed set of instructions, a learning robot improves its performance on a task over time by interacting with its environment or by observing demonstrations. This paradigm is particularly well-suited for the unstructured and unpredictable domains of space robotics, as it is impossible to pre-program a robot for every possible contingency.

2.3.1 Markov Decision Process

Many problems in robot learning are formalized using the mathematical framework of the Markov decision process (MDP) [15]. An MDP provides a model for sequential decision-making in an environment where outcomes are at least partially under the control of an agent. An MDP is defined by a set of states $\mathcal{S}$, a set of actions $\mathcal{A}$, and a set of rewards $\mathcal{R}$. The dynamics of the MDP are defined by a probability distribution p:

$$p(s', r \mid s, a) \tag{2.1}$$

This function specifies the probability of transitioning to state s' and receiving reward r, given that the agent was in state s and took action a at timestep t. The capital letters denote random variables, while the lowercase letters denote their specific values.

The agent's goal is to learn a policy, denoted as $\pi(a \mid s)$, which defines a mapping from states to a probability of selecting each possible action. The objective is to find an optimal policy π^* that maximizes the expected cumulative sum of future rewards, known as the return. For an ongoing task, the discounted return at timestep t is defined as:

$$G_t = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \tag{2.2}$$

where $\gamma \in [0, 1]$ is the discount factor. This factor determines the present value of future rewards. Values closer to 0 prioritize immediate rewards while values closer to 1 emphasize long-term gains. The core assumption is the Markov property, which states that the state S_t provides all necessary information for the agent to make an optimal decision, independent of the history of states, actions, and rewards that came before it. For example, in a robotic manipulation task, the full state might include the precise joint angles of the arm, the poses of all objects in the scene, and their mutual interaction forces.

However, in many real-world robotics problems, the agent does not have access to the complete state of the environment. This situation is known as partial observability and is more accurately described by a partially observable MDP (POMDP). A POMDP extends the MDP with a set of observations and an observation probability distribution $O(o \mid s', a)$. This gives the probability of receiving observation o after the agent took action a and the environment transitioned to state s'. In a POMDP, the agent receives an observation, such as an image from a camera, which may be a noisy or incomplete representation of the true state. It must often rely on a history of past observations to infer the underlying state and make informed decisions, a core challenge addressed in **Publication II**.

2.3.2 Reinforcement Learning

Reinforcement learning (RL) is a learning paradigm centered on the MDP framework. It is concerned with how an intelligent agent should take actions in an environment to maximize cumulative reward, without being explicitly told which actions to take [15]. The learning process is driven by trial and error. The agent explores the environment by taking actions and observing the resulting states and rewards. It uses this feedback to update its policy and gradually converges towards an optimal strategy. The fundamental interaction loop of RL is depicted in Figure 2.5.

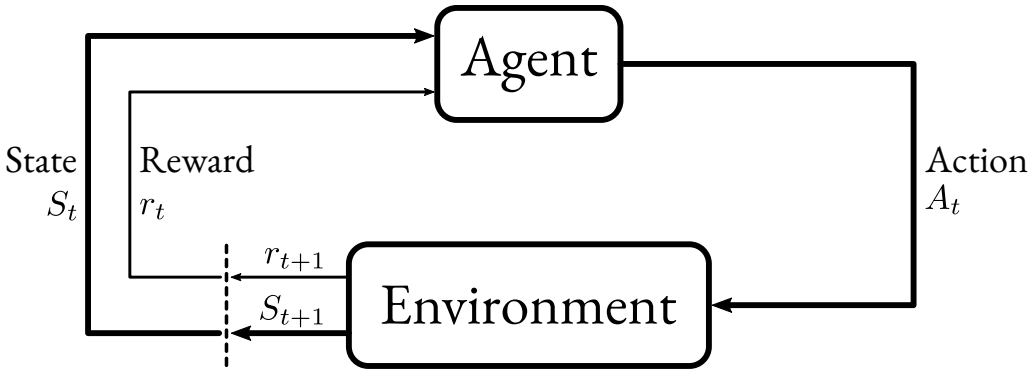


Figure 2.5 – The fundamental interaction loop of RL. The agent takes an action in the environment, which transitions to a new state and emits a reward. The agent uses this information to update its policy and inform future actions.

Modern RL algorithms are often combined with deep neural networks as function approximators, giving rise to the field of deep RL. This allows agents to learn complex policies directly from high-dimensional sensory inputs and to control robots with high-dimensional action spaces. Many prominent deep RL algorithms employ an actor-critic architecture, illustrated in Figure 2.6. In this setup, an actor network learns a parameterized policy $\pi_{\theta(a \mid s)}$, while a critic network learns a value function to evaluate the actor's actions by estimating their long-term

value. One common value function is the action-value function, which estimates the expected return for taking action a in state s and thereafter following policy π:

$$q_\pi(s, a) = E_\pi[G_t \mid S_t = s, A_t = a] \tag{2.3}$$

The critic, with parameters φ, learns an approximation $q_{\varphi(s,a)} \approx q_{\pi(s,a)}$. Its evaluation is then used to update the actor's parameters θ, typically by moving them in the direction of improved performance.

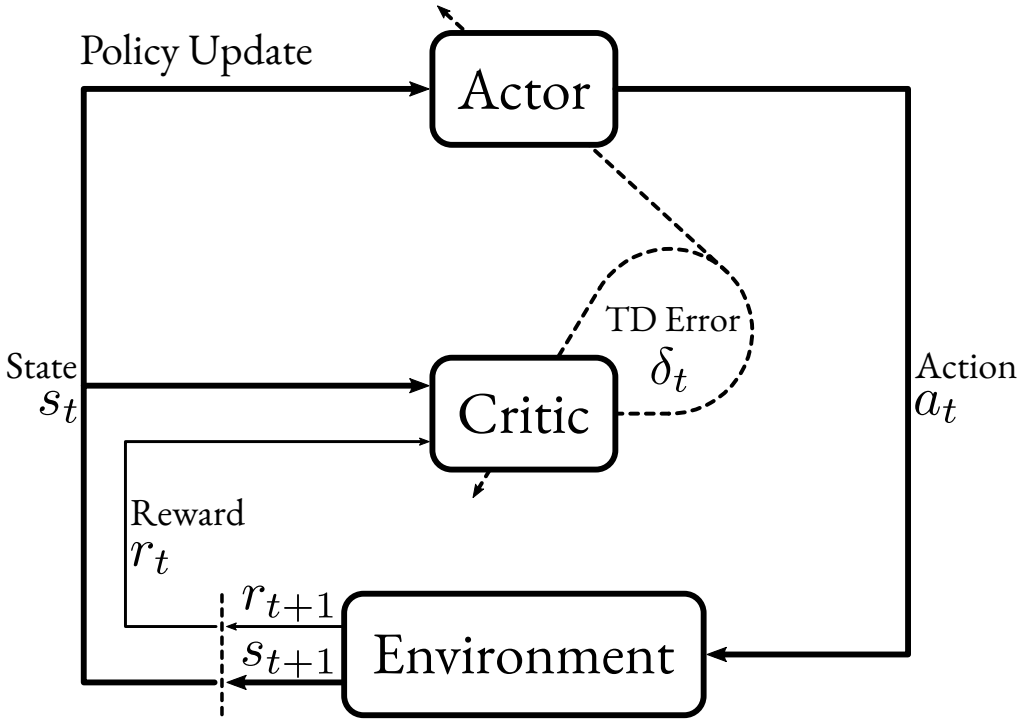


Figure 2.6 – The interaction loop of actor-critic RL. The actor (policy) selects an action, and the critic (value function) evaluates the quality of that action. The critic's feedback is then used to update the actor, guiding it toward more rewarding behaviors.

The algorithms can be broadly categorized as model-free or model-based. Model-free algorithms learn a policy and/or a value function directly from experience without explicitly learning a model of the environment's dynamics. They are often robust and can achieve high performance. Prominent model-free methods include on-policy algorithms like PPO [30], known for their training stability, and off-policy algorithms like TD3 [31] and SAC [32], which often exhibit greater sample efficiency. However, a common drawback of model-free approaches is that they can require millions or even billions of environmental interactions to learn a complex task.

Model-Based Reinforcement Learning

Model-based RL (MBRL) approaches aim to improve sample efficiency by employing a model of the environment's dynamics alongside the policy [56]. A learned world model, parameterized by ψ, is a function p_ψ that approximates the true dynamics p of the MDP:

$$(s', r) \sim p_\psi(\cdot \mid s, a) \tag{2.4}$$

This learned model acts as a surrogate for the real environment. It allows the agent to predict the consequences of its actions without having to physically execute them. The agent can use this model for planning or to generate large quantities of simulated experience to train its policy, as illustrated in Figure 2.7. Research in the area of MBRL has demonstrated significant gains in data efficiency on robotic control problems through algorithms like PILCO [57].

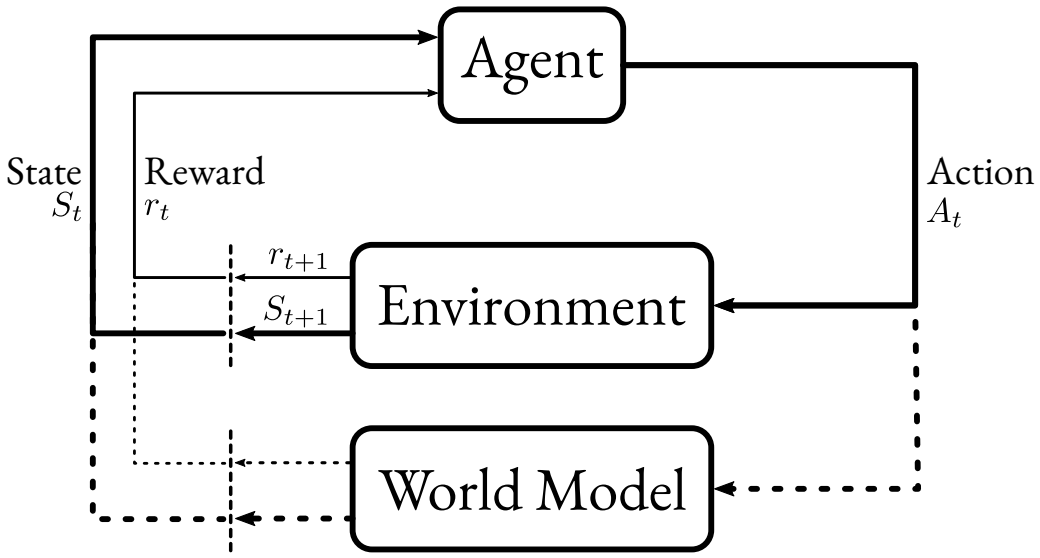


Figure 2.7 – The interaction loop of MBRL. The agent learns a world model from real-world interactions. It then uses this model for planning or to generate imagined trajectories, which are used to efficiently update the policy without requiring further real-world interaction.

Modern algorithms like DreamerV3 have demonstrated remarkable success by learning a compact, latent space model of the world directly from high-dimensional observations [33]. The agent can then learn a policy entirely within the imagined trajectories of its learned model. This process can be significantly faster and more data-efficient than interacting with the real or even a simulated environment. This ability to learn a predictive world model is particularly promising for space robotics. It provides a mechanism for planning and reasoning that is crucial for tackling complex, long-horizon tasks. This thesis heavily leverages the MBRL paradigm due to its potential for efficient learning and its inherent suitability for adaptation.

2.3.3 Learning from Demonstration

While RL offers a powerful framework for discovering behaviors through exploration, the process can be slow and undirected, especially for tasks with sparse rewards or complex action spaces where successful outcomes are rare. Learning from demonstration (LfD) provides an alternative or complementary approach that leverages existing data to accelerate and guide the learning process [58]. Instead of relying solely on trial and error, the agent learns by observing how a task should be performed. This data can be collected from human teleoperation, scripted policies, or the rollouts of a previously trained agent.

Imitation Learning

The most direct form of LfD is imitation learning (IL). In this paradigm, the problem is framed as a supervised learning task where the agent learns a policy that mimics the actions of an expert. The simplest approach, behavior cloning (BC), trains a policy to map states to actions from expert trajectories. Given its simplicity, BC has been successfully applied to complex robotic manipulation tasks [59]. Several works have also combined it with RL to learn contact-rich assembly skills [60], [61].

While straightforward, BC can suffer from issues like covariate shift. This occurs when small errors in the learned policy cause the agent to drift into states not seen in the expert data. Once off the expert's trajectory, the agent has no data to guide its recovery, which often leads to compounding failures. More advanced IL methods exist to mitigate this issue. For example, some techniques involve querying the expert for corrective actions from states the agent has visited. A fundamental constraint of pure IL remains. The learned policy's performance is ultimately bounded by the quality and coverage of the expert demonstrations. The agent can only learn to replicate what it has seen and cannot discover novel or better strategies on its own.

Offline Reinforcement Learning

Offline RL bridges the gap between RL and IL. It aims to learn a policy from a fixed, pre-existing dataset of transitions without any further interaction with the environment. Unlike pure IL, it does not assume that the data comes from an expert. The dataset can be a mix of optimal, suboptimal, and purely random behaviors. Offline RL algorithms use dynamic programming or model-based methods to stitch together the best parts of the trajectories in the dataset. This allows them to learn a policy that can potentially outperform the best trajectory seen in the data.

This paradigm is highly relevant for space robotics, where online data collection is expensive or dangerous. The primary challenge in offline RL is distributional shift. This occurs when the learned policy favors actions that lead to out-of-distribution states for which the value function

is not well-defined, often leading to erroneously high value estimates. Modern offline RL algorithms mitigate this issue using various forms of policy constraints or conservatism. This ensures the agent primarily utilizes actions and states that are well-supported by the dataset. The progress in this area is fueled by the availability of large-scale robotics datasets [38]. Although not investigated in this work, the capability of SRB to collect teleoperated data and record trajectories from trained RL agents provides the necessary infrastructure for future research into creating similar datasets for the space domain.

2.3.4 Sim-to-Real Challenge

A central premise of this thesis is that simulation provides the only feasible environment for training the complex robotic skills required for space. Yet, the real-world applicability is the ultimate goal. This reliance on simulation introduces a fundamental obstacle known as the sim-to-real gap. This gap refers to the discrepancy between the simulated environment and the real world, as illustrated in Figure 2.8 from **Publication I**. A simulation is always an approximation of reality, no matter how high its fidelity. Differences in visual appearance, object dynamics, friction, sensor noise, and actuator behavior can cause a policy that performs perfectly in simulation to fail completely when deployed on a physical robot. Bridging this sim-to-real gap is one of the most significant challenges in modern robot learning research.

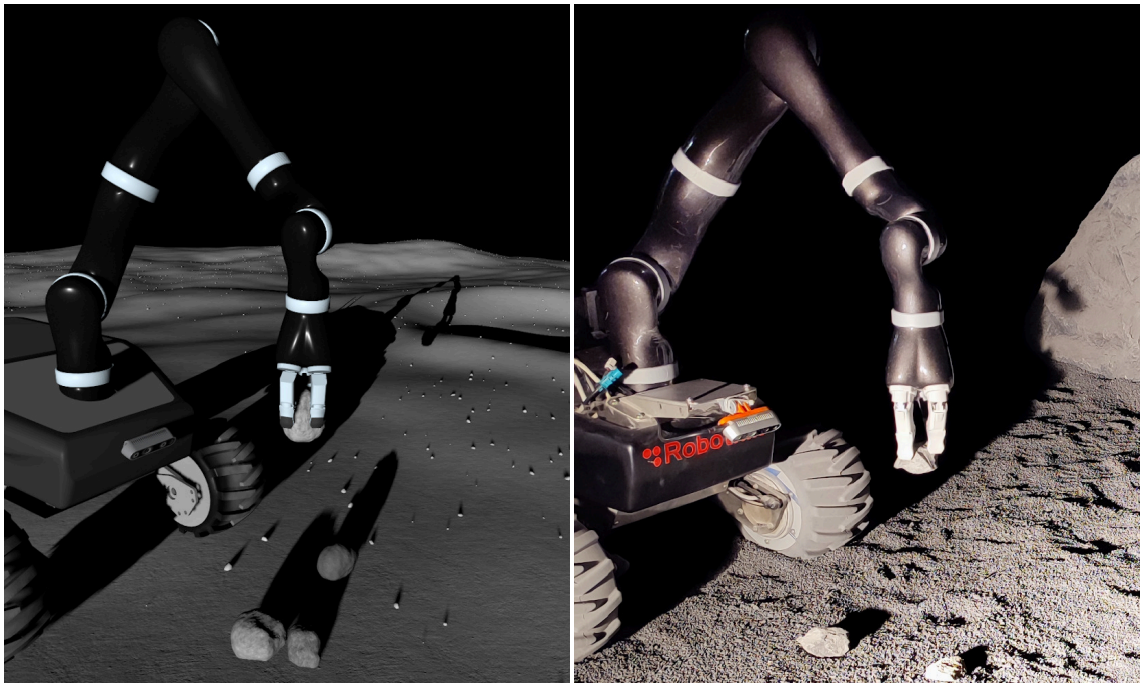


Figure 2.8 – A conceptual illustration of the sim-to-real gap, taken from the foundational study of **Publication I**. A policy trained exclusively in a simulation domain may fail when deployed in the real world due to discrepancies in physics, visuals, and other unmodeled effects. The goal of sim-to-real transfer is to develop policies that are robust to this gap.

Domain Adaptation

Domain adaptation techniques aim to make the simulation more closely resemble the real world or to make the learned policy less sensitive to the differences between the two. This can involve a range of methods. For example, system identification can be used to measure the physical parameters of a real robot and its environment, such as joint friction or actuation delays. These parameters can then be used to fine-tune the simulator's physics engine [21]. Other approaches involve fine-tuning a policy that was pre-trained in simulation with a small amount of data collected from the real world. While these methods can be effective, they often require access to the target real-world domain during the training process. This dependency on real-world data makes them challenging to apply in space robotics, where the target environment is remote, poorly characterized, and largely inaccessible.

Domain Randomization

Domain randomization (DR) takes a different and, in many ways, opposite approach. Instead of trying to create a perfect replica of one specific real-world scenario, it aims to make the simulation so diverse that the real world appears to the policy as just another variation [29]. This is achieved by systematically randomizing the parameters of the simulation during training, as shown in Figure 2.9. This can include randomizing visual aspects like lighting conditions, textures, and camera positions. It also includes randomizing physical properties like object masses, friction coefficients, and actuator dynamics.

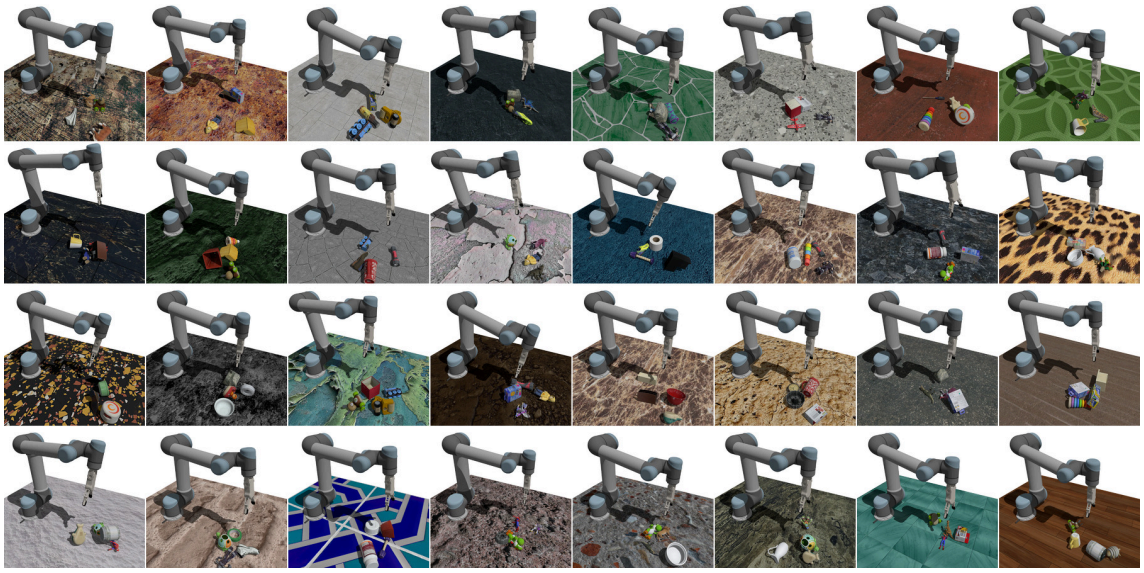


Figure 2.9 – An example of DR, where each training episode features a unique combination of graspable object models, textures, and illumination conditions. This encourages the agent to learn a generalizable grasping strategy [62].

By exposing the agent to a vast range of conditions, DR forces the policy to become robust to these variations. The agent learns to ignore superficial and irrelevant features of the environment and instead focuses on the underlying physical principles required to solve the task. This process acts as a form of implicit regularization that discourages the policy from a tendency to overfit to simulation-specific artifacts. More advanced techniques even allow the agent to learn to make the simulation more challenging for itself, further enhancing robustness [63]. Although domain knowledge is beneficial, this approach does not require any data from the target real-world environment. It is thus highly suitable for space applications where conditions are unpredictable. The methodology in this thesis relies heavily on DR as a core strategy for achieving robust generalization.

2.4 Ecosystem of Robotics Simulation

The development and validation of robotic systems are increasingly dependent on simulation. Simulators provide a safe, cost-effective, and scalable environment for prototyping algorithms, training learning-based agents, and testing system behavior under a wide range of conditions. This section surveys the landscape of available robotics simulators, discusses the transformative potential of procedural generation, and highlights the critical role of standardized benchmarks in the advancement of the field.

2.4.1 Proprietary and Open Source Landscape

The ecosystem of robotics simulation is diverse. It encompasses both proprietary commercial software and community-driven open-source projects. Each category presents distinct advantages and trade-offs. Open-source simulators like Gazebo [64] and physics engines such as MuJoCo [65] and DART [66] offer great flexibility, transparency, and accessibility. They allow researchers to modify and extend the software to suit their specific needs, which fosters a collaborative development environment. However, they have sometimes lagged behind their commercial counterparts in terms of rendering fidelity and computational performance.

In recent years, the line between these two categories has begun to blur. Powerful platforms have emerged that blend proprietary core technology with open access and APIs for the research community. Isaac Sim, the platform upon which the framework in this thesis is built, is a prominent example. It provides GPU-accelerated physics and physically-based rendering (PBR) capabilities that were largely exclusive to high-end proprietary software while also offering an open and extensible Python-based workflow. This shift has been instrumental in democratizing access to high-fidelity simulation. A comparison of several prominent simulators is provided in Table 2.1.

Table 2.1 – A comparison of prominent robotics simulators and game engines, qualitatively evaluating each platform on its primary strengths and limitations in the context of developing and validating learning-based autonomous systems.

Simulator	Key Strengths	Primary Limitations
CoppeliaSim[2]	• User-friendly	• Not optimized for scalability • Lower visual & physics fidelity
Drake[3]	• High-fidelity physics • Excellent for model-based control	• Not optimized for scalability
Gazebo[4]	• Large community • Close ROS integration	• Not optimized for scalability
Isaac Sim[5]	• Scalable • Realistic rendering	• Vendor-locked hardware requirements
MuJoCo[6]	• Large community • Scalable • High-fidelity physics	• Lower visual fidelity
PyBullet[7]	• User-friendly • Lightweight	• Lower visual fidelity
SAPIEN[8]	• Scalable • Realistic rendering	• Not general enough
Unity[9]	• Large community • Realistic rendering	• High integration effort for robotics
Unreal Engine[10]	• Large community • Realistic rendering	• High integration effort for robotics
Webots[11]	• User-friendly	• Not optimized for scalability • Lower visual fidelity

Despite the capabilities of these general-purpose platforms, they exhibit a profound terrestrial bias. Consequently, the challenge lies not merely in selecting a simulator but in constructing a domain-specific framework that leverages high-performance physics and rendering to generate the massive, diverse, and physically accurate experience that is essential for bridging the sim-to-real gap prevalent in space robotics.

[2]https://coppeliarobotics.com
[3]https://drake.mit.edu
[4]https://gazebosim.org
[5]https://developer.nvidia.com/isaac/sim
[6]https://mujoco.org
[7]https://pybullet.org
[8]https://sapien.ucsd.edu
[9]https://unity.com
[10]https://unrealengine.com
[11]https://cyberbotics.com

2.4.2 Space Simulation Frameworks

Simulation is a cornerstone of space mission development, traditionally used for high-fidelity verification of astrodynamics via tools like GMAT [18] and Basilisk [19]. However, space robotics introduces challenges that require physically realistic models of contact dynamics and complex terrains. To meet these needs, a new generation of specialized simulators has emerged, often developed by space agencies for specific missions, such as VIPER RSIM [67] or Astrobee Sim [68].

These mission-specific simulators are invaluable for their intended purpose of V&V, but are ill-suited for the broader needs of robot learning research. They are designed to model a static, singular reality with maximum fidelity, which is misaligned with the massive data generation and environmental diversity required by modern learning algorithms. In response, the research community has developed several learning-focused simulators for specific domains, including rover navigation [22], spacecraft rendezvous [23], and orbital manipulation [69]. While these platforms are crucial steps forward, they are often tightly coupled to their original research objectives and a single robotic platform, making it difficult to study generalization across different robots or applications. Table 2.2 summarizes this diverse landscape. SRB is designed for multi-domain autonomy with a focus on high scalability and extensibility, enabled by its combined PCG and DR approach for the sim-to-real challenge.

Table 2.2 – Comparison of existing space simulation frameworks.

Simulator	Primary Focus	Sim-to-Real	Scalability	Gymnasium
GMAT [18]	Orbital Mechanics	Digital Twin	Low	No
Basilisk [19]	Orbital Mechanics	Digital Twin	Medium	Yes [70]
VIPER RSIM [67]	Rover V&V	Digital Twin	Low	No
Astrobee Sim [68]	Free-Flyer V&V	Digital Twin	Low	No
Int-Ball2 Sim [71]	Free-Flyer V&V	Digital Twin	Low	No
HeliCAT-DARTS [72]	Rotorcraft V&V	Digital Twin	Low	No
EELS-DARTS [73]	Snake Robot V&V	Digital Twin	Low	No
OmniLRS [74]	Planetary Navigation	Photorealism	Medium	No
RLRoverLab [22]	Planetary Navigation	DR	High	Yes
RANS [23]	Spacecraft Navigation	DR	High	Yes
SpaceRobotEnv [69]	Orbital Manipulation	DR	Low	Yes
GraspPlanetary (**Publication I**)	Planetary Manipulation	DR+PCG	Low	Yes
SRB (**Publication VII**)	Multi-Domain	DR+PCG	High	Yes

2.4.3 Power of Procedural Content Generation

A key technique for enhancing the diversity and scalability of simulation environments is procedural content generation (PCG) , which encompasses a set of algorithms for creating data automatically rather than manually. While widely adopted in the gaming industry to create vast and varied game worlds, its application in robotics simulation is still emerging. Traditionally, robotic simulation environments have been built using manually created assets. This limits their variety and makes the creation of large-scale, diverse scenarios a labor-intensive process.

PCG offers a solution to this bottleneck. By using parametric pipelines, a near-infinite number of unique 3D assets can be generated on demand. This approach is not only efficient but also enables a level of environmental diversity that is unattainable through manual design. As argued throughout this thesis, diversity is a critical requirement for training truly generalizable robot learning policies [28]. By forcing the agent to succeed across a wide distribution of scenarios, PCG effectively regularizes the learned policy and prevents it from a tendency to overfit to simulation-specific artifacts. Furthermore, PCG can be used to facilitate curriculum learning, where the complexity of the generated content is gradually increased to guide the agent's learning process from simple to more challenging tasks [75]. The pipeline used in this thesis leverages Blender's Geometry Nodes to create a flexible and powerful system for on-demand asset creation, as introduced in **Publication IV**.

2.4.4 Role of Benchmarks in Driving Progress

Standardized benchmarks have played a crucial role in accelerating progress in machine learning. A benchmark provides a common set of tasks, environments, and evaluation metrics that allow researchers to directly compare the performance of their algorithms in a reproducible manner. This is particularly important in RL, where subtle differences in implementation can lead to significant variations in results, making reproducibility a persistent challenge [20].

In robotics, benchmarks like RLBench [24], Meta-World [25], ManiSkill3 [27], Robosuite [76], and RoboHive [77] have been instrumental in advancing research in terrestrial manipulation. Other benchmarks have focused on more specialized domains, such as long-horizon assembly in FurnitureBench [26], dexterous manipulation with RoboPianist [78], and humanoid locomotion via HumanoidBench [79]. While invaluable, these existing benchmarks are Earth-centric. They do not capture the unique challenges of space robotics, such as microgravity dynamics, unstructured planetary terrains, or the specific tasks relevant to space missions. This highlights a critical gap in the available tools for the community, a gap that **Publication VII** aims to fill. The development of a dedicated, open-source benchmark for space robotics is therefore a primary motivation for this thesis.

2.5 Synthesis and Research Gap

The preceding sections have traced the parallel and intersecting paths of space robotics, robot control, and robot learning. The synthesis of these fields reveals a clear and compelling need for a new approach to developing autonomous systems for space. The vision for future space exploration demands robots with a high degree of adaptive autonomy. Traditional control methods lack the necessary robustness for unstructured and remote environments. Robot learning, particularly deep MBRL, offers a promising paradigm for acquiring the required adaptive behaviors. Its application to space robotics is severely hampered by the sim-to-real challenge and the unique constraints of the domain.

This review of the state of the art exposes a critical research gap. The space robotics community has developed high-fidelity simulators but lacks an open platform designed for the data-intensive needs of modern robot learning [18], [19]. The robot learning community has produced powerful algorithms and benchmarks, but has focused on terrestrial applications that do not capture the unique physics of space [24], [26]. There is a clear need for a bridge between these two worlds.

This thesis aims to fill this gap by addressing three specific deficiencies. First, there is a lack of a comprehensive, open-source benchmark for robot learning in space that integrates diverse, procedurally generated environments with high-fidelity physics. Second, there is insufficient understanding of how to best train agents for robust generalization in contact-rich manipulation tasks, particularly with respect to learning compliant behaviors. Third, there is no systematic evaluation of state-of-the-art learning algorithms against a standardized suite of space-relevant tasks. This work directly confronts these issues by developing the necessary simulation infrastructure and a novel learning methodology to advance the state of adaptive autonomy for robots beyond Earth.

3

Laying the Foundation

The progress of this thesis towards achieving adaptive autonomy in space began not with a comprehensive benchmark but with an empirical investigation into a fundamental manipulation skill. This chapter details the initial research presented in **Publication I**, which served to validate the foundational hypotheses of this thesis. The work confronted a concrete challenge of teaching a robot to grasp unknown objects on the Moon. This task, while specific, served as a foundation for testing the core principles that underpin this entire thesis. The successful outcomes of this early work established the critical methodologies that would later be scaled and generalized into the full SRB and the adaptive control framework. It provided the first concrete evidence that a simulation-centric approach, grounded in procedural diversity and robust perception, could successfully bridge the sim-to-real gap for a complex space robotics task. This chapter recounts that foundational experiment, detailing its successes and exposing the limitations that motivated the more advanced work to come.

3.1 Core Concepts

The research focused on the task of vision-based robotic grasping of previously unknown objects in a simulated lunar environment. Grasping was selected as it is a prerequisite for a vast range of mission-critical operations, from sample collection and ISRU to the assembly and maintenance of infrastructure. The goal was not merely to solve this specific problem but to use it as a testbed for establishing a set of core principles for learning generalizable policies. Four key concepts were central to this initial investigation.

First, the work revolved around a simulation-centric development approach. This principle acknowledges the fundamental impossibility of collecting the vast datasets required for modern RL through physical trials in space. The prohibitive cost, safety-critical nature, and logistical constraints of extraterrestrial operations mandate that the virtual world serve as the primary training ground. This decision necessitated the development of a simulation that was not only physically and visually realistic but also sufficiently diverse to foster the robust generalization required for real-world deployment.

Second, to achieve this diversity, the research introduced the use of PCG for creating environmental assets. This concept directly confronts the limitations of training on static, finite datasets of 3D models. Such datasets fail to capture the sheer variability of an unknown environment like the lunar surface. By developing custom pipelines to synthesize a wide variety of terrains and rocks algorithmically, the training distribution becomes effectively infinite. This approach forces the agent to learn the general physical principles of grasping rather than memorizing the specific visual features of a limited set of objects.

Third, the research challenged the sufficiency of traditional 2D image-based observations for complex 3D manipulation. It proposed a novel approach based on 3D octree representations of the environment. The hypothesis was that a 3D data structure would provide a more natural and robust input for a policy that must reason about objects and movements in three-dimensional space. A 3D representation offers inherent invariance to camera viewpoint and sensor resolution. The octree was chosen specifically for its computational and memory efficiency compared to dense voxel grids [80], [81], [82].

Finally, to ensure the learned policy would be transferable across different robotic arms, the control was formulated in the robot's task space. A policy that learns to command individual joint angles is intrinsically tied to a specific kinematic structure. By instead commanding the desired displacement of the EE in Cartesian space, the policy becomes agnostic to the underlying kinematics of the manipulator. This decouples the high-level goal of the task from the low-level mechanics of the robot, a crucial step towards creating truly general-purpose and reusable skills. These four concepts, when integrated, formed the blueprint for the successful proof-of-concept experiment detailed in this chapter.

3.2 Simulation-Centric Approach

The entire development and training pipeline for this foundational study was centered within a custom simulation environment. This approach was not a matter of convenience but a necessity driven by the core challenges of RL. The millions of interactions required to train a deep neural network policy are impossible to collect on physical hardware, especially for a safety-critical and inaccessible domain like space. The primary goal, therefore, was to create a virtual world that was not just a replica of a single scenario but a diverse and challenging testbed for learning.

The environment was built using the Gazebo robotics simulator [64]. It was chosen for its open-source nature, which promotes reproducibility, and its mature and extensible architecture. For this contact-rich grasping task, physical realism was paramount. The simulation leveraged the DART physics engine for its stable and accurate handling of rigid-body dynamics [66]. Visual fidelity was achieved using the OGRE 2 rendering engine, which provided PBR

capabilities for more realistic lighting and material interactions. The resulting environment was designed to encapsulate the key characteristics of a mobile manipulation task on the lunar surface. It integrated a simulated rover with an articulated robotic arm and a stereo camera for visual perception.

3.2.1 Procedural Generation of Assets

A core innovation of this early work was the extensive use of PCG to create the environmental assets. This strategy directly addressed the inadequacy of existing 3D model datasets for this specific domain. While large-scale datasets of common objects exist, they are not representative of the natural, geological forms found on planetary surfaces. Manually creating thousands of unique rock and terrain models would be prohibitively time-consuming. PCG provided a scalable solution.

New pipelines were created using the Geometry Nodes feature of Blender. These node-based systems allowed for the programmatic and parametric creation of assets, as conceptualized in Figure 3.1. For the terrain, a flat 2D plane was programmatically displaced using a combination of procedural noise textures to create an uneven surface with features resembling impact craters. For the rocks, simple convex polyhedra were similarly displaced to generate a wide variety of shapes and sizes, including complex non-convex geometries. Each generated model was also assigned a random set of PBR material textures from a static dataset to increase visual diversity. This approach allowed for the creation of a nearly unlimited number of unique environments from a small set of procedural rules.

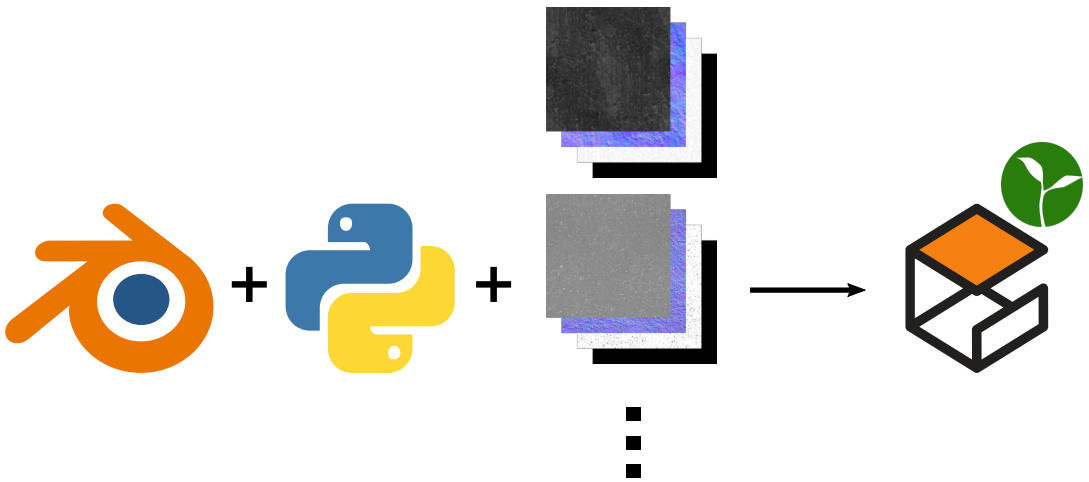


Figure 3.1 – The procedural generation pipeline for creating diverse lunar assets. It uses Blender's Geometry Nodes to generate terrains and rocks, which are then exported with PBR textures via automated Python scripts for direct use in the Gazebo simulator.

Extensive DR was applied during training. In each episode, the simulation randomized the pose of the rover, the camera extrinsics, the terrain model, the number and models of the rocks, their physical properties like density and friction, and even the direction of the simulated sunlight to mimic the harsh illumination conditions on the Moon [29]. This constant variation was essential for preventing the agent from a tendency to overfit and for encouraging the development of a robust, generalizable policy.

The simulated task involved a Summit XL-GEN mobile manipulator, a rover equipped with a 7-DoF Kinova Gen2 robotic arm and a three-finger gripper, as shown in Figure 3.2. The agent's task was to grasp one of the procedurally generated rocks scattered in its workspace and lift it to a specified height. Visual perception of the scene was provided by a simulated stereo camera, which is a sensor choice motivated by its prevalence on actual space rovers [43]. The camera produced both monochromatic images and depth maps of the scene from the perspective of the rover base.

To maintain focus on the core manipulation skill, the problem was constrained to the immediate workspace of the manipulator. Any prior movement of the rover was assumed to be handled by other means. The simulation environment provided a standardized interface compatible with modern RL frameworks, specifically following the OpenAI Gym API [83] and facilitated by the Gym-Ignition library [84]. This allowed for seamless integration with the learning algorithm and facilitated a structured, episodic formulation of the grasping task. Communication between the simulation, the learning agent, and the robot control stack was managed using ROS 2 [36]. This choice simplified the eventual transfer of the learned policy to the physical robot by providing an identical software interface in both the simulated and real domains.

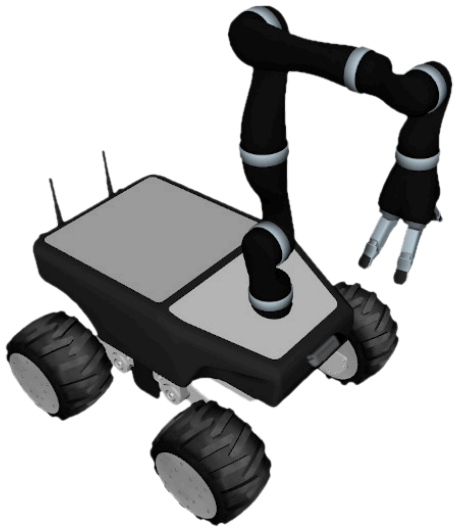


Figure 3.2 – The Summit XL-GEN mobile manipulator, with a 7-DoF Kinova Gen2 robotic arm and a stereo camera, which was used in both the simulation and the physical experiments.

3.3 End-to-End Learning from 3D Octree Observations

The learning methodology for this initial study was designed to directly map sensory inputs to robot actions in an end-to-end fashion. This approach contrasts with traditional robotics pipelines that segment a task into discrete stages such as perception, pose estimation, grasp planning, and motion execution. Such pipelines can be brittle, as errors from an earlier stage can compound and lead to failure in later stages. An end-to-end policy, in contrast, learns a single function that maps raw observations directly to low-level control commands, which can lead to more robust and reactive behaviors, albeit often at the cost of reduced interpretability and explainability [85].

The approach, illustrated in Figure 3.3, was built on the key innovation of using a compact 3D data structure for representing the visual scene while employing a model-free RL algorithm for policy optimization. The entire system was designed to be invariant to the specific robot kinematics and robust to the visual diversity introduced by the procedural simulation environment. The grasping task was formulated as an MDP, and the agent's objective was to find an optimal policy π^* that maximizes the expected discounted return from Equation (2.2).

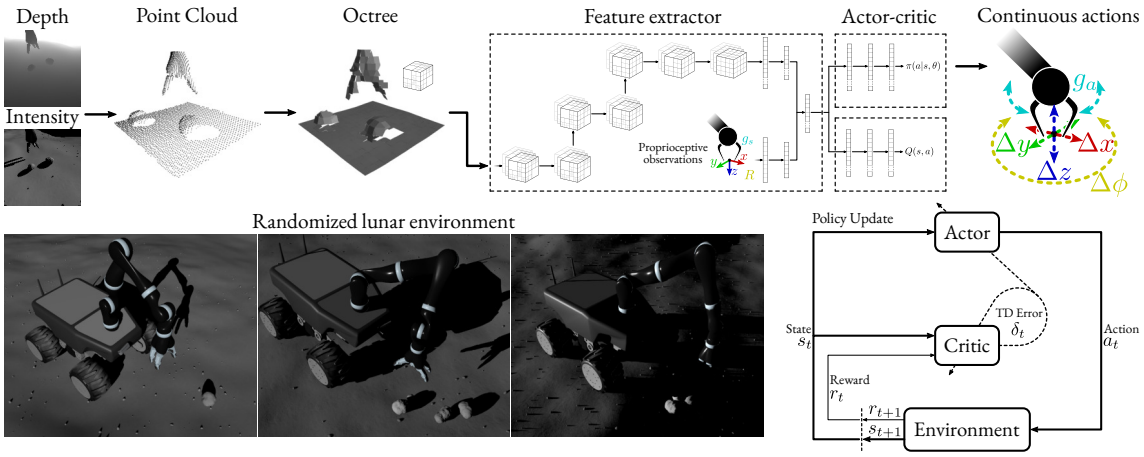


Figure 3.3 – An overview of the end-to-end learning approach. Visual data from a stereo camera is converted into a 3D octree representation. This, along with proprioceptive data, is fed into a shared feature extractor. The resulting features are then used by separate actor and critic networks to produce continuous task-space actions for the robot EE.

3.3.1 Octree-Based Scene Representation

A central hypothesis of this work was that traditional 2D image-based observations are suboptimal for learning 3D manipulation tasks. While 2D convolutional networks generalize well over the horizontal and vertical position of features in an image, they do not inherently understand the spatial relationships of a three-dimensional world, a limitation noted in other

robotics research [86]. To address this, the visual scene was represented as a 3D octree in a choice motivated by the need for a data structure that is both computationally efficient and provides a rich geometric representation suitable for 3D convolutional networks [87].

The process of constructing the observation, illustrated in Figure 3.4, began by converting the depth map and monochromatic image from the simulated stereo camera into a point cloud. This point cloud was transformed into the robot base coordinate frame and cropped to a fixed volume of $40 \times 40 \times 40$ cm. This fixed volume was then used to construct the octree, a hierarchical data structure that efficiently represents the 3D space by recursively subdividing it into eight octants [82]. Unlike a dense voxel grid, an octree only recursively subdivides and stores data for occupied space, making it highly efficient for the sparse scenes typical of a robot workspace.

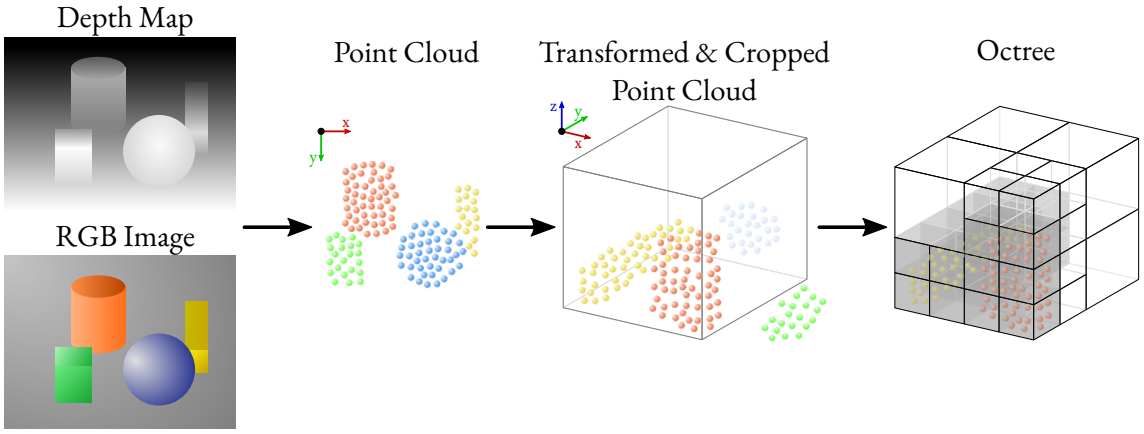


Figure 3.4 – The process of creating an octree-based scene representation. A raw point cloud is generated from sensor data. This point cloud is then transformed into the robot coordinate frame and cropped to a fixed volume. Finally, it is voxelized into an efficient octree data structure, which serves as the primary visual input for the learning agent.

To provide the agent with a rich description of the scene, three distinct features were computed and stored in the leaf nodes of the octree for each occupied cell, as depicted in Figure 3.5. These features were the average unit normal vector $\overline{n}$ which provided crucial information about the local surface geometry, the average distance of the points from the cell center $\overline{d}$, and the average intensity $\overline{i}$ for textural information. This multi-modal feature set gave the agent a compact yet descriptive summary of the scene's geometry and appearance. The observation was completed with proprioceptive data, including the gripper's pose using a continuous 6D rotation representation [88] and its open or closed state.

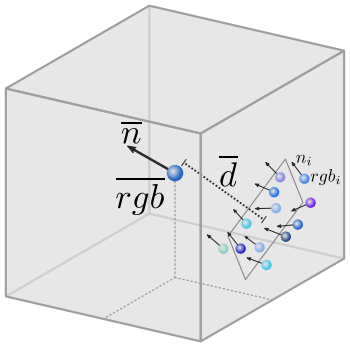


Figure 3.5 – A representation of the features stored within a single leaf octant of the octree. All points from the source point cloud that fall within the volume of a cell are used to compute the average unit normal vector $\overline{n}$, the average distance to the cell center $\overline{d}$, and the average intensity $\overline{i}$.

3.3.2 Model-Free Reinforcement Learning

To learn the grasping policy, a model-free, off-policy actor-critic algorithm known as Truncated Quantile Critics (TQC) was employed [89]. TQC is a variant of SAC [32] that is well-suited for continuous control problems. It operates under the maximum entropy RL framework, which augments the standard reward objective with an entropy term to encourage exploration.

A novel neural network architecture, shown in Figure 3.6, was designed to process the 3D octree observations. It adapted an octree-based convolutional network [80] to serve as a shared feature extractor for both the actor and critic. This feature extractor processed the octree through a series of 3D convolutions and pooling layers to produce a compact latent space representation of the scene. Sharing the feature extractor reduces the total number of learnable parameters and allows both the policy and the value function to benefit from a common, rich representation of the state.

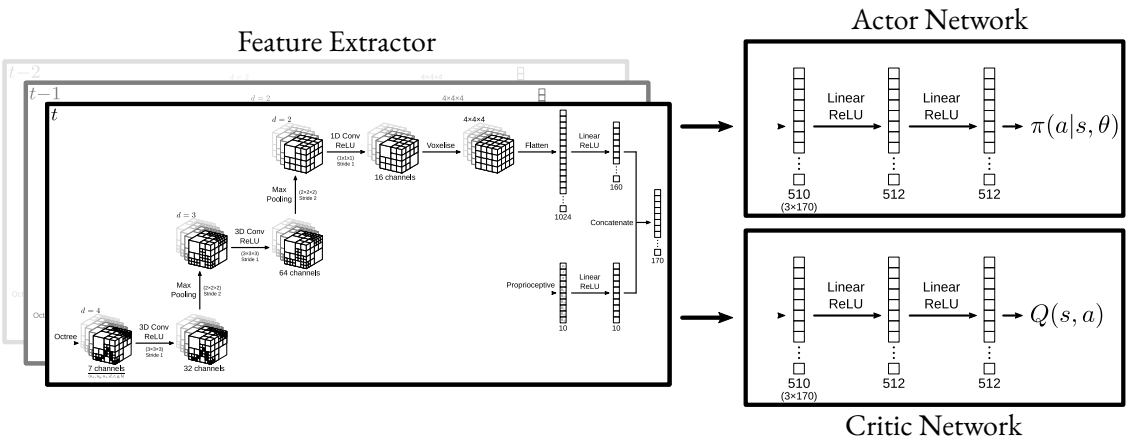


Figure 3.6 – The network architecture used for learning from octree observations. A shared octree-based convolutional network extracts features from the 3D representation. These features, combined with proprioceptive data, are then fed into separate actor and critic networks.

The policy learned by the agent was designed to be robot-agnostic. The action space $\mathcal{A}$ was defined in the Cartesian space of the EE:

$$a_t = [\Delta x, \Delta y, \Delta z, \Delta \varphi_z, g_a] \tag{3.1}$$

where the first four terms represent the relative translational displacement and yaw rotation of the gripper, and g_a is a continuous value controlling the gripper's state. These high-level commands were then sent to the MoveIt 2 motion planning framework, which used a combination of TRAC-IK and RRT-Connect solvers to calculate the necessary joint movements to execute the action while ensuring collision-free motion [53], [90].

To guide the learning process, a shaped reward function was used. The task was decomposed into four sequential stages, namely reaching, touching, grasping, and lifting. The agent received a sparse positive reward upon the completion of each stage, with the reward value increasing exponentially to incentivize progress through the entire sequence. This provided a denser learning signal that guided the agent towards the final goal.

3.4 Sim-to-Real Validation in Lunar Analogue Facility

The ultimate test for any policy trained entirely in simulation is its performance in the physical world. For space robotics, this validation must take the form of a zero-shot transfer. The policy must work out of the box. The final phase of this foundational research was therefore a rigorous zero-shot transfer experiment to validate the effectiveness of the developed approach.

This experiment was conducted in the LunaLab, a unique Moon-analogue facility at the University of Luxembourg [37]. The LunaLab is specifically designed to replicate the visual characteristics of the lunar surface. It features a basin filled with basalt gravel that mimics the appearance of lunar regolith, and a configurable light projector that can reproduce the harsh illumination found on the Moon. This facility provided a realistic and challenging testbed for the physical validation.

The Summit XL-GEN mobile manipulator used in the simulation was employed for the physical experiments, as shown in Figure 3.7. The real robot was equipped with an Intel RealSense D435 stereo camera that was mounted at the base of the rover. To systematically evaluate the core hypotheses, the experiment was designed around two key variables, namely the type of sensory observation, either 2D image or 3D octree, and the level of environmental diversity during training, reduced or full DR. The real-world task involved scattering a set of eight different physical rocks, shown in Figure 3.8, within the robot workspace and tasking the agent with grasping one of them over 25 independent trials for each condition.

Figure 3.7 – The physical setup for the sim-to-real validation experiments, conducted in the LunaLab facility at the University of Luxembourg that is filled with 20 tons of basalt gravel [37].

Figure 3.8 – The eight physical rocks used during the sim-to-real evaluation.

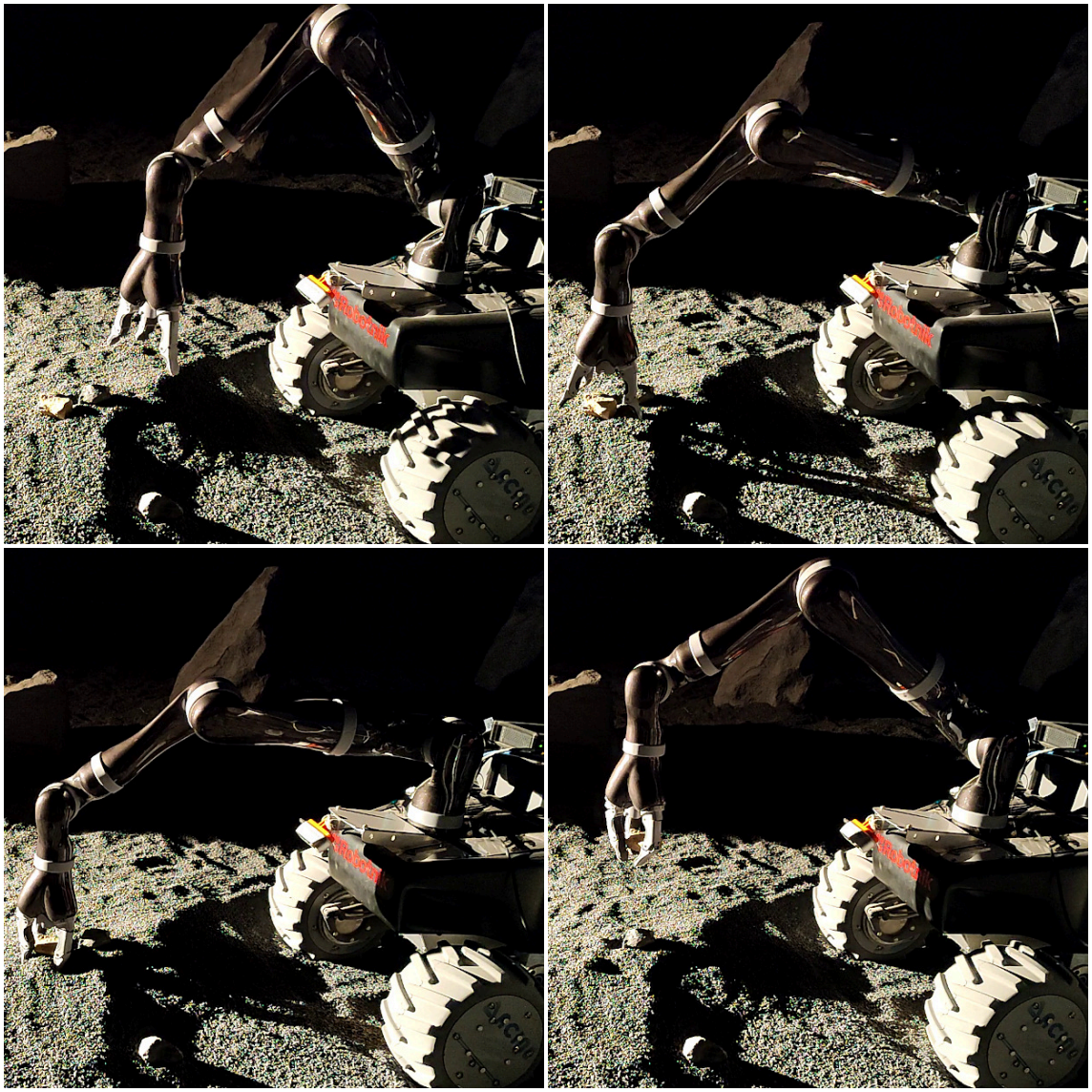


Figure 3.9 – A successful real-world grasp sequence.

A successful grasp sequence is depicted in Figure 3.9. The agent, guided solely by its learned policy, successfully perceives the scene, selects a target rock, approaches it with the gripper, secures a stable grasp, and lifts it off the ground. Despite the limited sim-to-real success rate, the quantitative results from Table 3.1 reveal two critical insights. First, agents trained in the fully randomized simulation environment with PCG assets demonstrated significantly better performance on the real robot than those trained in the static environment with reduced DR. This result was particularly apparent because the agents trained with reduced randomization achieved much higher success rates within their low-diversity simulation environment used during training. It demonstrates that they had not learned a generalizable skill, but had effectively overfit to the small and predictable set of scenarios. This provided strong evidence that a combination of PCG and DR is a critical component for bridging the sim-to-real gap.

Table 3.1 – Quantitative results of the sim-to-real transfer experiment.

Observation Type	Level of Randomization	Success Rate (N=25)
Image	Reduced	12%
Image	Full	20%
Octree	Reduced	8%
Octree	Full	32%

Second, the agent trained with 3D octree observations achieved the highest success rate of all, outperforming its 2D image-based counterpart. This validated the hypothesis that a 3D scene representation provides a more robust foundation for learning complex manipulation skills. The successful demonstration of a learned policy grasping previously unseen physical rocks in a realistic analogue facility marked a critical milestone. It proved the viability of the entire simulation-centric learning pipeline.

3.5 Limitations and Key Takeaways

This foundational study on learning to grasp on the Moon yielded several key insights that profoundly shaped the subsequent research direction of this thesis. It served as a successful proof-of-concept. It validated the core premise that a simulation-centric approach could be used to train policies for complex, contact-rich manipulation tasks and successfully transfer them to the real world.

The most significant takeaway was the demonstrated importance of diversity in training. The experiments showed that agents trained with extensive PCG and DR were far more robust and generalized better to the novel conditions of the physical world. This finding established environmental diversity not as an optional enhancement but as a fundamental requirement for achieving meaningful sim-to-real transfer. Secondly, the study provided strong evidence for the superiority of 3D vision for manipulation. Finally, the successful use of task space control and a modular software architecture suggests the viability of creating hardware-agnostic skills.

However, this initial work also revealed several critical limitations that motivated the next phases of research. While the model-free RL algorithm was successful, its learning process was often sample-inefficient and exhibited instability, especially in the highly varied procedural environments. This highlighted the need to explore more advanced and data-efficient learning paradigms, such as MBRL. A learned world model could potentially learn to filter out irrelevant environmental variations and enable the agent to learn more efficiently.

Furthermore, the control strategy, while effective for grasping, was based on rigid kinematic control. The policy had no mechanism for managing contact forces. It could only command a target pose, and the underlying controller would attempt to reach it regardless of resistance. This pointed towards the need to investigate compliant control methodologies like OSC. A truly adaptive agent must be able to modulate its physical interaction, not just its position in space.

Lastly, and most importantly, the entire experiment was built around a single simulation environment for one specific skill. The ad-hoc nature of this initial setup was not scalable for comparing different algorithms, testing a wide range of robots, or exploring long-horizon tasks. This infrastructural limitation was the most significant barrier to further progress. It became clear that before more advanced learning methodologies could be developed, a proper foundation had to be built. These limitations directly informed the design of SRB and the development of the adaptive control framework detailed in the chapters that follow.

4

Forging Virtual Frontiers

The proof-of-concept detailed in the previous chapter served as a critical validation. It confirmed that a simulation-centric approach grounded in procedural diversity could indeed bridge the sim-to-real gap for a complex space robotics task. However, that foundational study also exposed a profound set of limitations. The bespoke, single-task environment was an effective testbed for one experiment but represented a significant infrastructural bottleneck to broader scientific inquiry. To systematically investigate generalization, evaluate advanced learning algorithms across diverse tasks, and explore the capabilities of a wide range of robot morphologies, a dedicated and standardized platform was required. This chapter introduces the core contribution of this thesis, the Space Robotics Bench (SRB).

SRB is not merely a collection of simulation environments but a complete framework designed to accelerate the development of adaptive autonomy for space applications. It is the architectural answer to the infrastructural gap identified in the background review and the experimental limitations revealed in the foundational grasping study. This chapter provides a thorough overview of SRB, detailing its design philosophy, its core architectural components, the suite of benchmark tasks it contains, and its seamless integration with the broader robotics and machine learning ecosystems. It presents SRB as the foundational artifact upon which the core scientific investigations of this thesis are built, providing the necessary tools to rigorously test and validate the learning methodologies for achieving robust and generalizable robotic behavior in space. The SRB source code is openly available at github.com/AndrejOrsula/space_robotics_bench.

4.1 Design Philosophy

The design and development of SRB were guided by a set of core principles formulated to address the specific needs of the robot learning and space robotics research communities. They ensure that SRB is not just a powerful tool for this thesis, but also a valuable and lasting resource for researchers and developers working on the future of autonomous systems in space.

Realism and Diversity

The framework prioritizes the accurate simulation of space-relevant physics, including modeling of regolith as granular media via particle physics based on extended position-based dynamics [91]. However, this realism is not pursued for a single, static scenario. The core philosophy of this thesis dictates that true robustness for learning lies in capturing the immense diversity and unpredictability of space. This approach reframes the sim-to-real gap not as a fixed bias to be bridged, but as a distribution of potential realities to be encountered and overcome. The goal is to train agents that are robust not because they have mastered one perfect simulation, but because they have learned to generalize across thousands of imperfect but varied ones, making the real world appear as just another variation [29]. This is achieved through the deep integration of PCG and extensive DR. For complex physical phenomena that are not explicitly modeled, such as atmospheric drag, the framework approximates their impact by applying random disturbances, forcing an agent to learn a policy that is inherently robust to a wide range of unmodeled dynamics.

Scale and Performance

Modern robot learning algorithms are notoriously data-hungry. To support this demand, SRB is built to be highly parallelizable. Leveraging the GPU-accelerated physics and rendering of Isaac Sim, the framework can run thousands of unique simulation instances concurrently, achieving a throughput of over 100k simulation steps per second for some tasks. This scale dramatically accelerates data collection for online RL. The performance is further enhanced with backend optimizations, including Rust extension modules for high-performance CPU-bound logic via PyO3 [92] and offloading computationally expensive task logic to the GPU using TorchScript [93] to maximize data throughput.

Modularity and Extensibility

Recognizing that every space mission is unique, SRB is designed with a flexible and modular architecture. It avoids the monolithic, hard-coded task definitions of the initial grasping environment. All assets, sceneries, robots, actuation models, and tasks are instead implemented as configurable, interchangeable modules registered within an internal registry. This allows researchers to easily introduce new elements by providing a standard model format and a data-validated configuration, and in turn adapting the framework to their specific needs without significant changes to the core codebase.

Openness and Accessibility

To maximize its impact and foster a collaborative research community, SRB has been developed as a fully open-source project, directly addressing the lack of standardized benchmarks. The framework is accompanied by comprehensive documentation, user-friendly installation procedures, and a unified command-line interface. Crucially, it provides standardized interfaces to established ecosystems, including the Gymnasium API [35] for the robot learning community and the ROS 2 [36] middleware for the broader robotics community.

4.2 Benchmark Suite

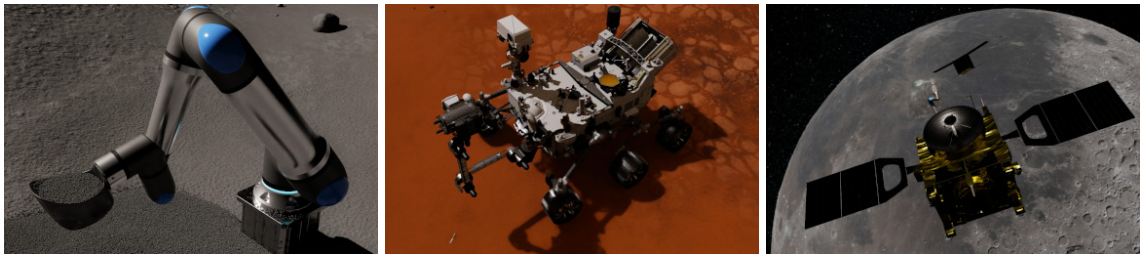
SRB is not a single environment but a comprehensive suite of assets, robots, and tasks that can be composed into a vast array of mission-relevant scenarios. These components are the fundamental building blocks of the benchmark, providing the practical implementation of the modular design philosophy. This section details these components, beginning with its positioning in the simulation landscape before diving into the multi-domain physics, the extensive robotic fleet, and the standardized tasks that constitute the suite.

4.2.1 Situating SRB in the Simulation Landscape

The development of SRB was motivated by a critical gap in the existing ecosystem of simulation tools. As outlined in Table 2.2, current space simulators are typically designed for mission-specific V&V and are often proprietary, while existing robot learning benchmarks exhibit a profound terrestrial bias and fail to capture the unique physics of space. SRB is purpose-built to bridge this gap, offering an open-source platform that uniquely synthesizes massive parallelism, procedural diversity, and space-relevant challenges. It complements mission-specific simulators by providing a platform for data-intensive development of generalizable policies, and it expands the horizons of the robot learning community by introducing a new suite of challenges beyond the well-explored domain of tabletop manipulation.

4.2.2 Domains

With its general-purpose simulation backend, a key feature of SRB is its ability to simulate a variety of extraterrestrial domains, each with distinct physical and visual characteristics. The framework models the gravitational pull of different celestial bodies and environmental factors like the intensity, color temperature, and angular diameter of solar illumination. Each domain presents unique challenges that directly influence robot dynamics, control strategies, and perception, as showcased in Figure 4.1.



(a) Surface of the Moon (b) Martian surface (c) Orbit — Low Lunar Orbit

Figure 4.1 – A selection of simulated SRB domains, ranging from planetary environments to orbital scenarios. Each domain is characterized by its unique physical, visual, and environmental properties that directly impact the design and control of autonomous systems.

The framework supports five primary domains. The **orbit** domain simulates the microgravity environment of space. The **asteroid** domain represents small celestial bodies with variable low gravity. Planetary domains include the **Moon** and **Mars**. Finally, a standard terrestrial **Earth** domain serves as a crucial baseline and is vital for sim-to-real transfer. The key properties of these domains are summarized in Table 4.1.

Table 4.1 – A comparison of the physical properties and associated robotic challenges for the primary domains simulated in SRB.

Domain	Gravity ($\frac{m}{s^2}$)	Solar Irradiance ($\frac{W}{m^2}$)	Key Robotic Challenges
Orbit	0.0	1361	Momentum management, reaction dynamics, ...
Asteroid	0.14 ± 0.14	190 ± 25	Low-traction mobility, unpredictable dynamics, ...
Moon	1.62 ± 0.01	1361	Cratered terrain, harsh lighting, abrasive regolith, ...
Mars	3.72 ± 0.01	590 ± 113	Challenging terrain, dusty atmosphere, ...
Earth	9.81 ± 0.03	775 ± 225	Sim-to-real transfer for baseline performance validation

Beyond these domain-level characteristics, SRB supports additional application-specific interactions, including articulated rigid-body dynamics, propellant consumption of spacecraft, and particle physics for regolith.

4.2.3 Robotic Fleet

SRB includes a diverse and extensible fleet of robotic platforms, organized into three main categories: **mobile robots**, **manipulators**, and **mobile manipulators**. The collection is summarized in Table 4.2 and includes commercially available robots and custom designs inspired by actual space hardware, enabling the systematic investigation of how a physical embodiment influences learning. Although not space-rated, the inclusion of commercial robots is a deliberate choice that offers a practical and more accessible starting point for students and academics due to their widespread availability and well-documented specifications.

Table 4.2 – A quantitative overview of the robotic fleet available in SRB.

Category	Sub-Category	Examples	Number of Platforms
Mobile Robots	Wheeled	Perseverance, Pragyan	6
	Legged	Spot, ANYmal	7
	Aerial	Ingenuity, Crazyflie	2
	Spacecraft	ISS, PCG cubesat	11 + 1 PCG
Manipulation Systems	Serial Arms	Canadarm, Franka, Kinova, UR	14
	Active EEs	Gripper, dexterous hand	8
	Passive EEs	Screwdriver, PCG scoop	8 + 1 PCG
Mobile Manipulators	Humanoids	Unitree H1, Unitree G1	4
	Combined	Any mobile base + manipulator	Composable

A key feature is its compositional design, which allows researchers to dynamically create novel mobile manipulators by combining any mobile base, manipulator arm, payload, and EE, as shown in Figure 4.2. While not all of the 5000+ possible configurations are physically sensible, this flexibility is aimed at empowering researchers to explore a vast design space.

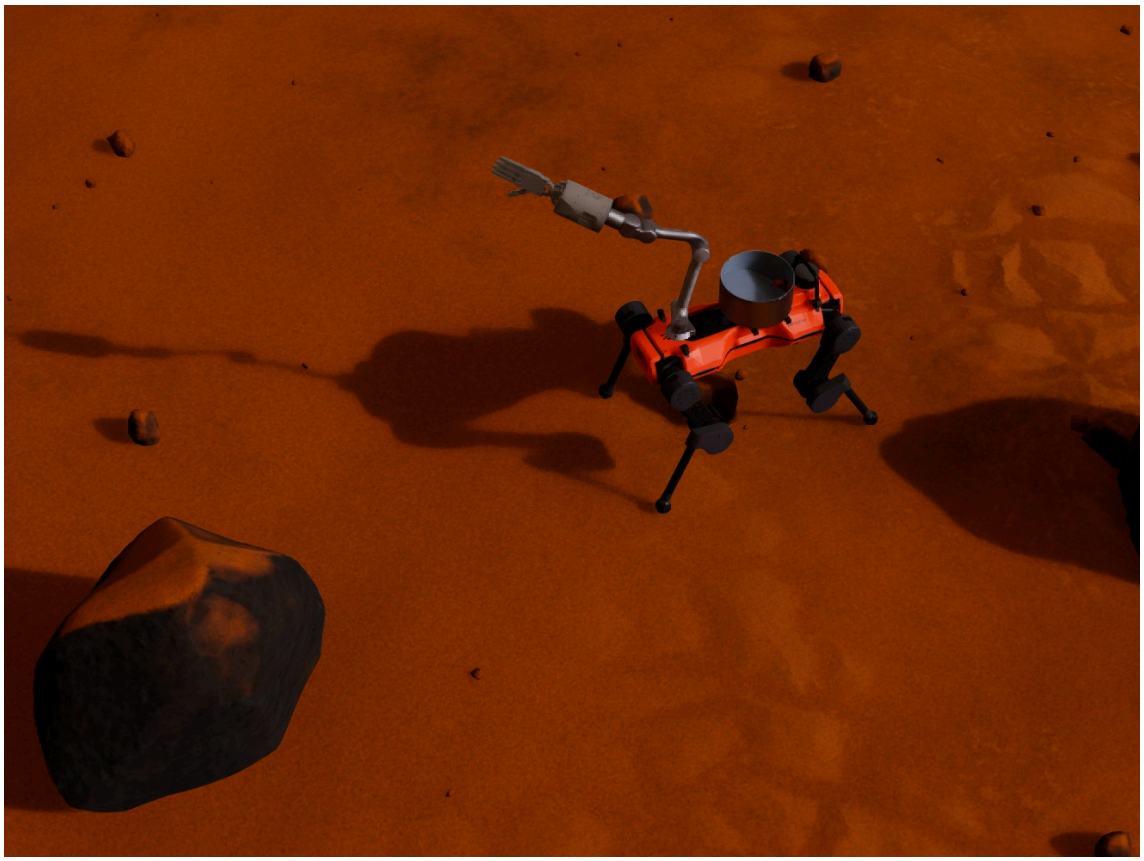


Figure 4.2 – An example of the modular composition of robots in SRB. The framework allows for the combination of any compatible mobile base with any manipulator and EE to create novel mobile manipulation systems.

Actuation Models

SRB provides a library of modular actuation models that map normalized agent actions to low-level robot commands. The framework emphasizes high-level abstractions to improve generalization across embodiments, such as target velocity control for wheeled robots and IK or OSC for manipulators. Furthermore, the composition of action spaces for multi-component robots and multi-robot systems is fully automated. For instance, an agent controlling a mobile manipulator interfaces with a unified action space that seamlessly combines the individual models for the mobile base, manipulator, and actuated EE. Supported control modalities include:

- **Wheeled**
  - Target linear & angular velocity mapped via kinematics
  - Target joint velocities
- **Legged & Humanoid**
  - Target joint positions
- **Aerial**
  - Target linear & angular accelerations
- **Spacecraft**
  - Activation of static/gimbaled thrusters with limited fuel
  - Target linear & angular acceleration
- **Manipulator**
  - Differential IK or OSC
  - Target joint positions
- **End-Effector**
  - Target joint positions
  - Target joint velocities

Sensory Modalities

SRB also provides agents with access to a rich variety of sensory information, organized into distinct modalities to support diverse learning paradigms.

- **State:** Privileged simulation information (ground-truth poses, velocities, contact forces)
  - *Purpose: Establish performance upper bounds, privileged (asymmetric) learning*
- **Proprioception:** Internal measurements (kinematic state, IMU readings, fuel)
  - *Purpose: Train policies with realistic onboard sensor data*
- **Visual:** Rendered images (RGB, depth, normals, segmentation masks)
  - *Purpose: Learn end-to-end policies directly from pixel inputs*
- **Commands:** High-level control signals (target velocity, relative waypoint pose)
  - *Purpose: Develop goal-conditioned policies for navigation tasks*

To clearly distinguish between learning paradigms and optimize performance, tasks that utilize visual sensors are purposefully registered as separate Gymnasium environments with a `_visual` suffix. This makes the intended input modality explicit and ensures non-visual tasks run more efficiently. Furthermore, observations are divided based on whether their dimensionality is fixed or varies with robot morphology. This design enables research into cross-morphology policy learning by isolating morphology-invariant observations from those that depend on the specific robot configuration.

4.2.4 Standardized Tasks

Building upon the diverse domains and robotic fleet, SRB provides a suite of standardized benchmark tasks. Each task is designed to test specific capabilities, as showcased in Figure 4.3 and detailed in Table 4.3. These tasks are designed to be representative of the challenges required for future space missions, ranging from basic mobility to long-horizon, complex sequences. The design prioritizes the evaluation of complete, mission-relevant skills over narrow, isolated sub-problems. For example, the `peg_in_hole` task requires the full pick-and-place sequence, not just the final insertion. This holistic approach produces more practical and generalizable skills that can be composed into complex, long-horizon behaviors. The full suite of tasks is organized into three primary categories: **mobile robotics**, **manipulation**, and **mobile manipulation**.

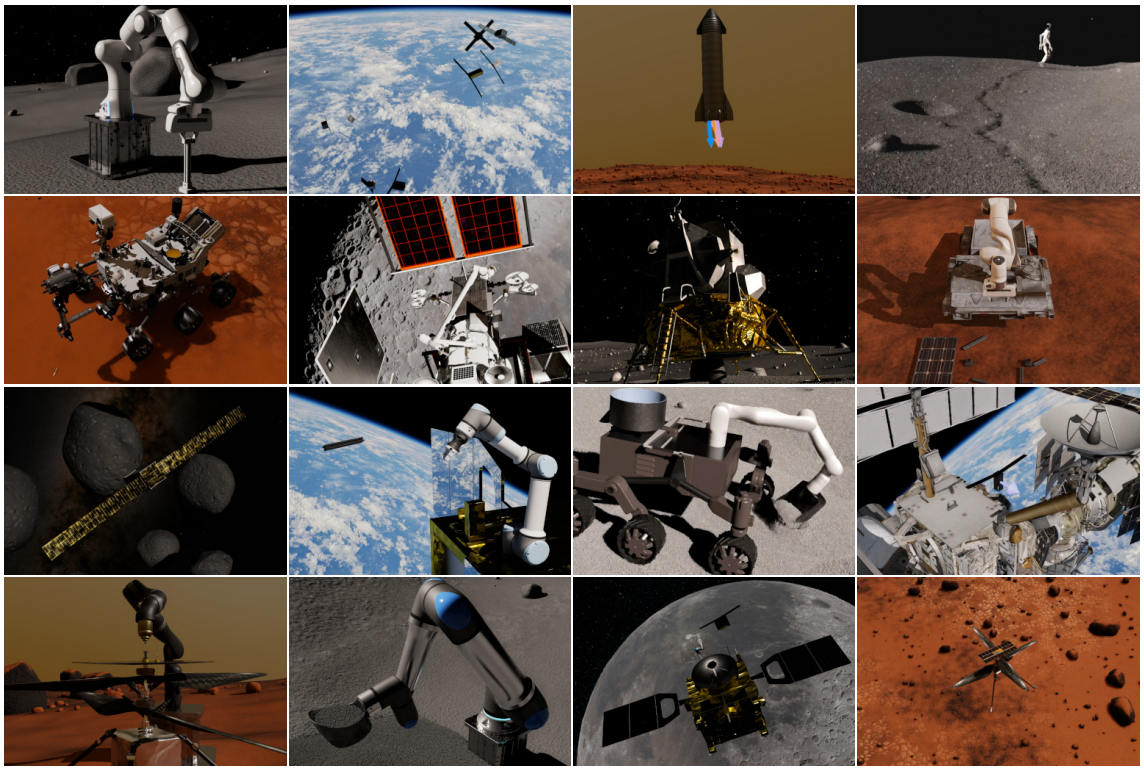


Figure 4.3 – A collage showcasing the diversity of SRB tasks.

Table 4.3 – Overview of the standard SRB benchmark tasks with their primary focus.

Task ID	Objective	Robot Morphology
Mobile Robotics		
landing	Descend and safely land with limited fuel	Spacecraft
rendezvous	Approach and match the state of a tumbling object	Spacecraft
orbital_evasion	Maneuver to avoid dynamic obstacles	Spacecraft
velocity_tracking	Follow dynamic velocity commands	Wheeled
↳ locomotion_*	↳ Variant for legged systems	Legged/Humanoid
waypoint_navigation	Track a dynamic waypoint	Wheeled
↳ locomotion_*	↳ Variant for legged systems	Legged/Humanoid
↳ aerial_*	↳ Variant for aerial vehicles	Aerial
↳ orbital_*	↳ Variant for spacecraft	Spacecraft
Manipulation (Fixed Base)		
debris_capture	Capture and stabilize a tumbling debris	Manipulator
sample_collection	Grasp a domain-specific sample	Manipulator
↳ multi_*	↳ Variant with multiple samples	Manipulator
excavation	Excavate and lift granular media	Manipulator
peg_in_hole	Pick up and precisely insert a peg into its hole	Manipulator
↳ multi_*	↳ Variant with multiple assemblies	Manipulator
screwdriving	Fasten a bolt into a threaded hole	Manipulator
solar_panel_assembly	Assemble a structure via a sequence of insertions	Manipulator
Mobile Manipulation		
mobile_debris_capture	Approach and capture a tumbling debris	Spacecraft + Manipulator
mobile_excavation	Excavate and store granular media	Ground + Manipulator

4.3 Core Architecture

The capabilities of the SRB are supported by a carefully designed software architecture that integrates a high-performance simulation backend, a powerful procedural generation engine, and a comprehensive system for DR. This synergistic relationship creates the scalable and unpredictable training distribution that is essential for forging truly adaptive autonomy.

4.3.1 Simulation Backend

The foundation of SRB is built upon Isaac Sim, a modern robotics simulation platform powered by NVIDIA Omniverse [94]. Its GPU-accelerated physics and ray-traced rendering enable the parallel execution of many independent simulation instances with complex interactions on a single workstation. As showcased in Figure 4.4, this architecture enables the simultaneous rendering and physics simulation of many parallel environments, which is a feature fundamental to making the procedural paradigm practical while satisfying the data-hungry requirements of modern robot learning.

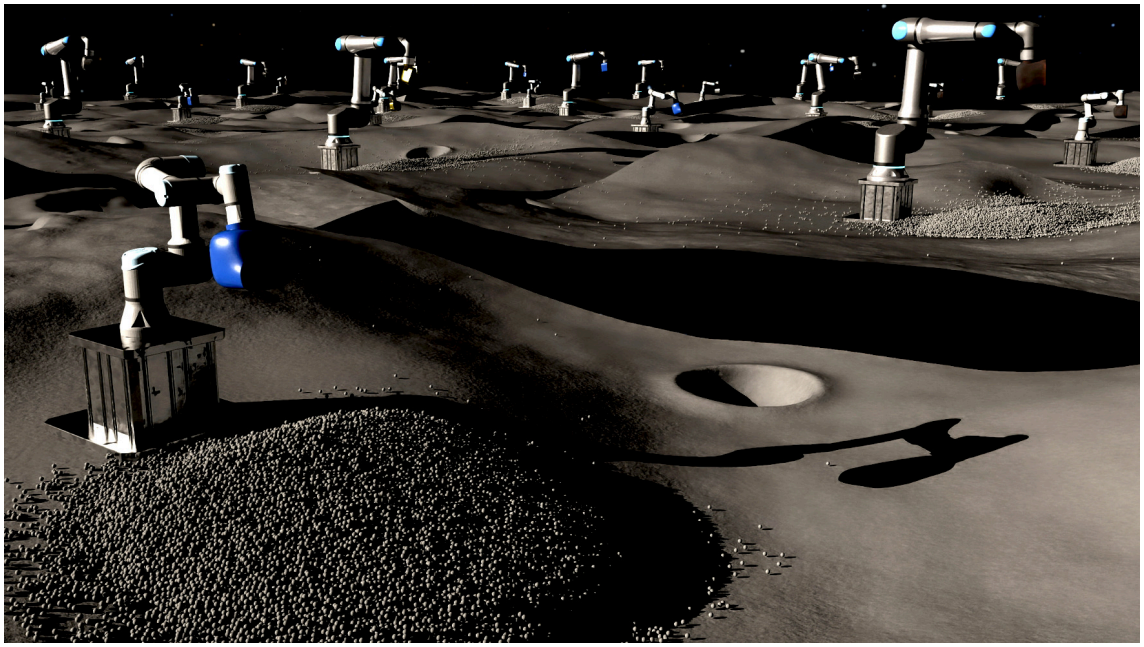


Figure 4.4 – The parallel architecture of SRB in action during the training of the `excavation` task. Each of the 64 environment instances features a unique PCG terrain and a distinct particle physics system for simulating the granular regolith as discrete particles.

To quantify the scalability of SRB, the simulation throughput was evaluated across the benchmark tasks using a workstation equipped with an AMD Ryzen 9 7950X CPU and an NVIDIA RTX 4090 GPU. The results, presented in Figure 4.5 demonstrate the aggregate throughput with respect to the number of parallel environment instances.

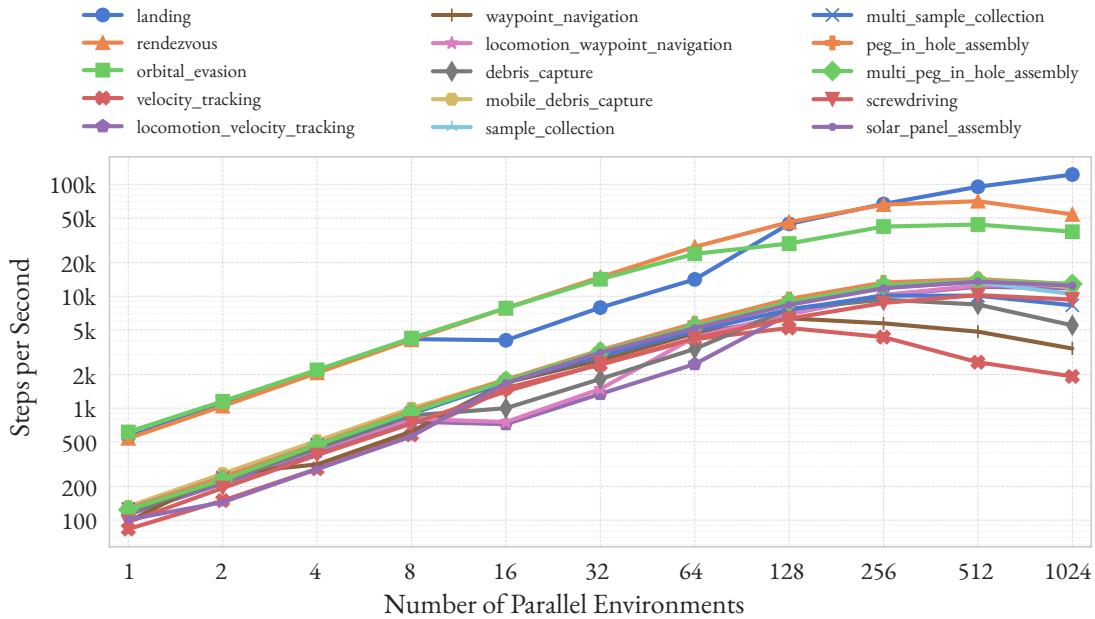


Figure 4.5 – The aggregate throughput of SRB tasks.

4.3.2 Procedural Engine

To populate the simulation with a near-infinite variety of content, SRB integrates a custom procedural engine named SimForge. This engine automates the process of generating 3D assets on demand using parametric pipelines created in Blender [95]. SimForge acts as a flexible, simulator-agnostic asset factory that decouples the complex task of content creation from the simulation environment itself. Its modular architecture, shown in Figure 4.6, is based on three core concepts: declarative **assets** that act as blueprints, **generators** that produce content from these blueprints, and **integrations** that connect the engine to frameworks like SRB.

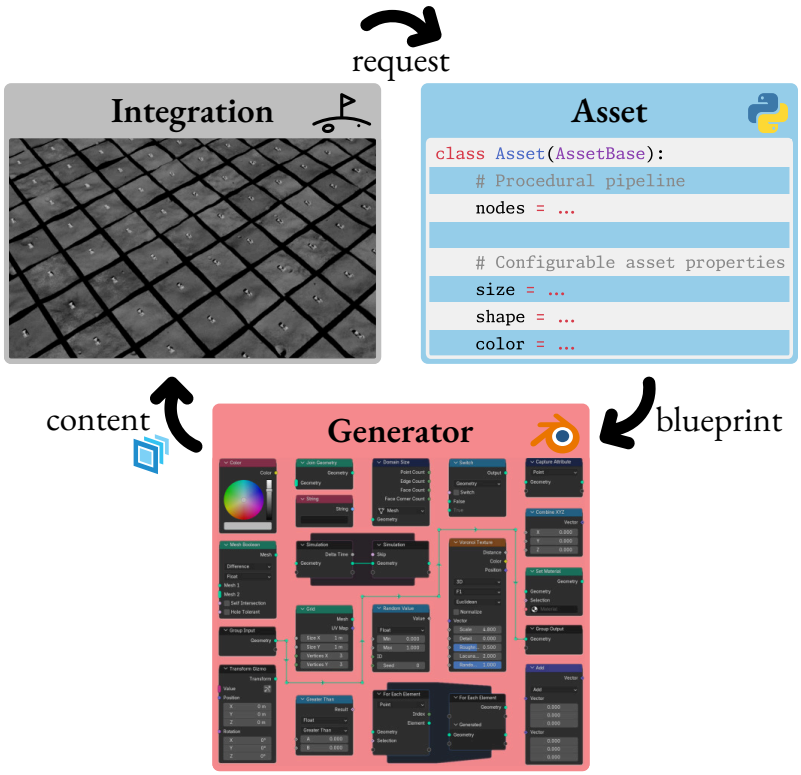


Figure 4.6 – The modular workflow of the SimForge architecture, where an integration can request assets, which are then produced by a generator and returned to the integration.

The integration between SimForge and SRB is realized through a seamless, on-demand asset pipeline. Engineered for high-throughput operation, this system transforms the static concept of a simulation environment into a dynamic, generative process, enabling the runtime creation and loading of unique procedural content for each parallel instance. As illustrated in Figure 4.7, the workflow unfolds in five distinct stages when a new simulation is launched:

1. **Initialization:** The user defines a scenario, specifying the number of parallel environments, the desired robots, and the procedural assets (e.g., terrains, tools, targets). SRB consumes this configuration and prepares the main simulation stage.

2. **Request:** For each parallel environment, SRB generates a unique, deterministic seed. It then identifies all procedural assets required for the scenario and dispatches a batch request to the SimForge API, specifying the asset blueprints and their corresponding seeds. This ensures every requested asset will be a distinct and reproducible variant.
3. **Generation:** SimForge receives the batch request and invokes its generation backend in a headless mode for each asset variant. This step is managed in the background, ensuring the generation process is deterministically isolated from the simulation runtime.
4. **Export:** As each asset is generated, the automated workflow of SimForge bakes its procedural materials into a set of standard PBR textures. The final model, comprising both its mesh and textures, is exported as a Universal Scene Description (USD) file to a shared cache directory, which is a format optimized for rapid and parallel loading within the Omniverse ecosystem.
5. **Spawning:** SRB monitors the cache for the newly created USD files. Once available, it loads each asset directly into memory and spawns it into its designated parallel environment.

This entire pipeline is designed for efficiency, making the on-demand generation of hundreds of unique procedural worlds a practical reality that can be completed in a matter of seconds.

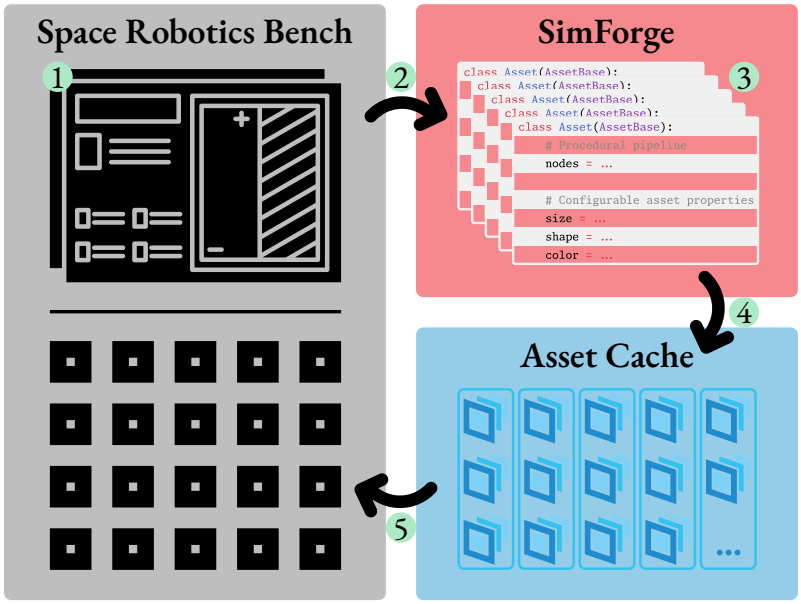


Figure 4.7 – The on-demand asset generation pipeline that programmatically links SRB with SimForge: (1) Initialize, (2) Request, (3) Generate, (4) Export, and (5) Spawn.

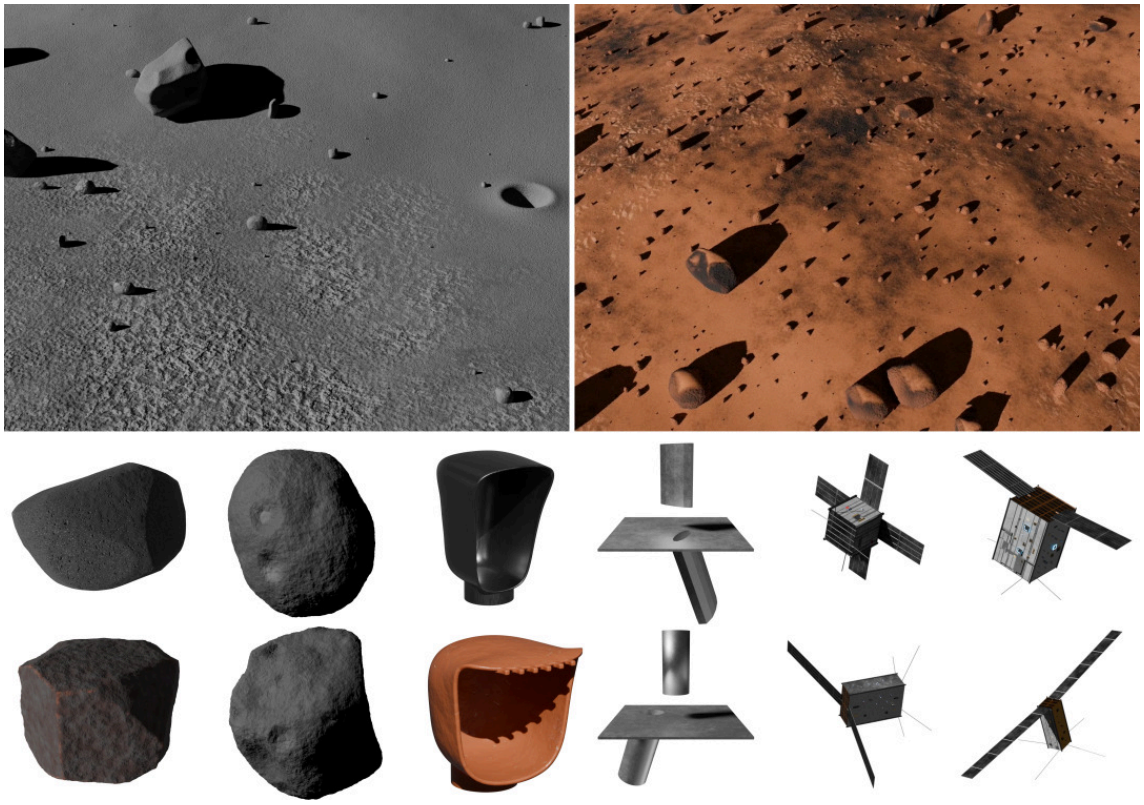


Figure 4.8 – Examples of procedural assets generated by SimForge, including varied terrains, rocks, and structured 3D models like excavation tools, peg-in-hole modules, and spacecraft.

The primary generator uses Blender to construct parametric pipelines for a wide range of assets, from extraterrestrial landscapes to spacecraft components, as shown in Figure 4.8. Each blueprint is a declarative Python class, as shown in the asteroid example of Listing 4.1. The generation process is fully automated so that when a new environment instance is initialized, a unique seed is passed to SimForge, which uses it to deterministically generate a unique asset.

```python
class Asteroid(GeometryBase):
    # Procedural pipeline
    nodes: Nodes = Nodes("geo_nodes.py")

    # Configurable asset properties
    detail: int = 5
    scale: Sequence[float] = (1.0, 1.0, 1.0)
    ...
```

Listing 4.1 – A declarative SimForge asset definition for a procedural asteroid geometry. This Python class serves as a high-level blueprint that specifies the parameters for the underlying Blender generation pipeline, making asset creation programmatic and reproducible.

4.3.3 Extensive Domain Randomization

While SimForge provides structural diversity through PCG, DR applies a final, critical layer of parametric variation. At the beginning of each training episode, a wide range of simulation parameters are sampled from pre-defined distributions across three main categories: **physical**, **visual**, and **dynamic**. As summarized in Table 4.4, this comprehensive randomization strategy is essential for training agents that can handle the unpredictable nature of real-world physics and perception.

Table 4.4 – A summary of the parameters subject to DR in SRB.

Category	Randomized Parameters	Purpose
Physics	• Gravity magnitude • Inertial matrix • Contact parameters • Material properties • Actuator modeling	Learning policies robust to unmodeled physical variations and hardware degradation.
Visuals	• Lighting direction • Lighting intensity • Lighting color • Skydome appearance • Camera pose • Post-processing effects	Achieving invariance to superficial visual changes and different lighting conditions.
Dynamics	• Initial entity state • External disturbances • Sensor noise level	Training policies that are stable and can recover from unexpected disturbances.

This three-tiered architecture of scalable parallelism, procedural diversity, and comprehensive randomization forms the technical core of SRB to provide the rich and varied stream of experience necessary to train adaptive and generalizable policies.

4.4 Open Platform for the Community

A core objective of the SRB is to serve as a catalyst for research and collaboration. It is designed as an open and extensible platform for seamless integration with existing tools and workflows.

4.4.1 Integrations and Interfaces

SRB directly integrates a number of modern RL libraries like Stable Baselines3 [96] and skrl [97]. To support more traditional development workflows and bridge the gap to physical hardware, the framework features a comprehensive, dynamically configured ROS 2 [36] interface that exposes nearly every aspect of the simulation to the ROS ecosystem. This allows researchers to leverage standard robotics tools like the RViz2 visualizer, as shown in Figure 4.9.

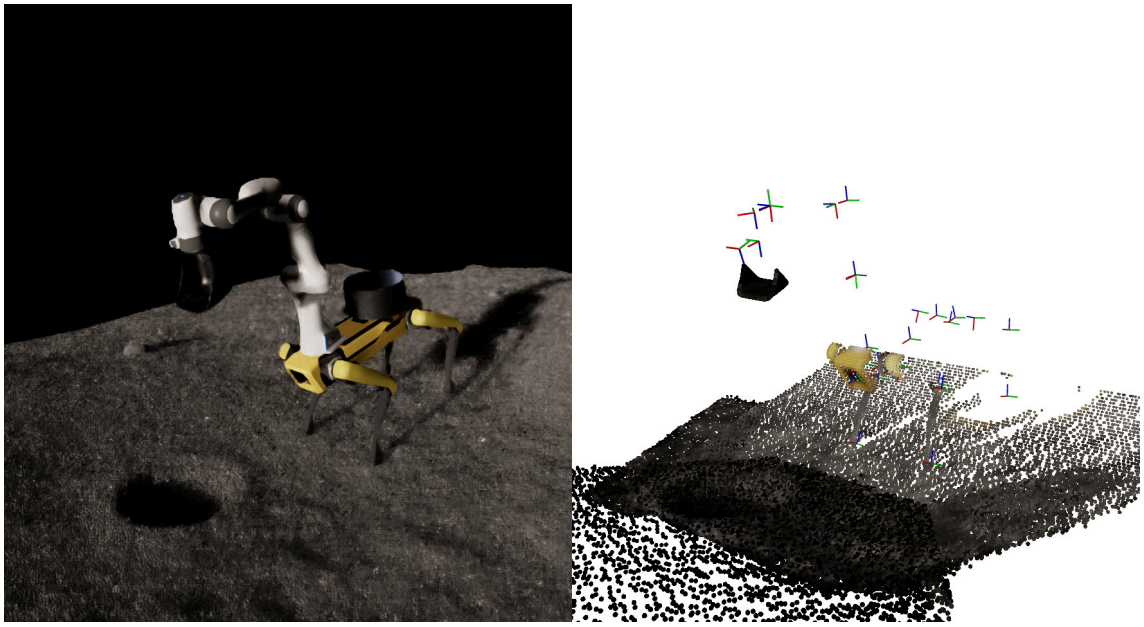


Figure 4.9 – An example of the ROS 2 interface showing the RViz2 visualization of the state and visual observations for a mobile manipulator inside SRB simulation.

Inherited from the underlying backend, SRB also supports the collection of visual data with ground truth annotation. As shown in Figure 4.10, this includes synchronized RGB, depth, and segmentation streams from an onboard camera. This is essential for developing learning-based perception systems for tasks like orbital inspection.

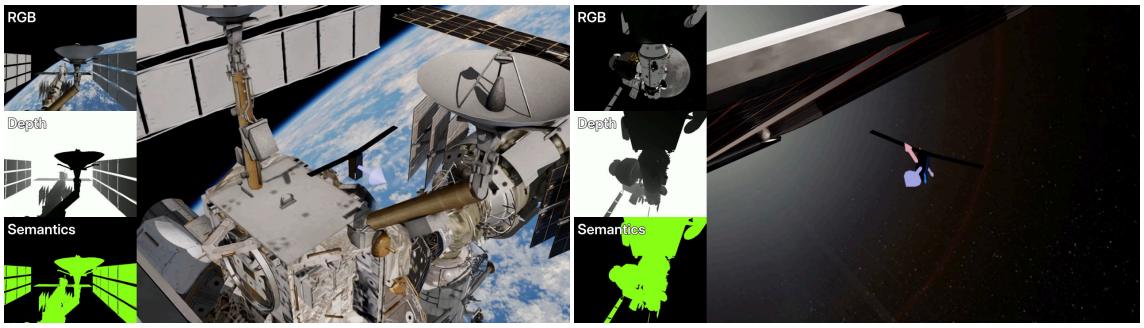


Figure 4.10 – Orbital inspection scenario of SRB providing synchronized RGB, depth, and segmentation streams from an onboard camera maneuvered around the ISS and Gateway.

Furthermore, the framework provides comprehensive support for human-in-the-loop teleoperation using various input devices like those shown in Figure 4.11, enabling expert data collection for IL and direct comparison of autonomous and human performance.

(a) Gamepad (b) SpaceMouse (c) Haptic interface

Figure 4.11 – A selection of teleoperation interfaces supported by SRB.

4.4.2 Sim-to-Real Workflow

A key innovation enabling rapid hardware deployment is the framework's automated sim-to-real workflow. This system leverages runtime reflection to inspect a simulated Gymnasium environment and generate its lightweight, real-world counterpart. This hardware-specific environment routes the actions, observations, rewards, and termination signals through a set of modular hardware interfaces that can be implemented via any middleware, such as ROS 2. The generation process automatically handles critical details like action space scaling and rate limiting to ensure consistency between the simulated and real domains. This modular design provides a versatile and robust solution for bridging the sim-to-real gap, forming the critical pathway for the validation experiments in Chapter 6.

4.5 Vision for Standardized Evaluation of Robots in Space

The ultimate purpose of SRB extends beyond serving as a tool for individual research projects. It aims to catalyze a paradigm shift in how autonomous systems are developed and validated for space. By offering a shared suite of challenging tasks, a diverse robotic fleet, and consistent performance metrics, the benchmark enables researchers to directly compare the efficacy of novel algorithms in a reproducible scientific manner. This transforms development from disconnected efforts into a collaborative, community-driven pursuit.

Crucially, such a vision goes beyond simply measuring success rates on static tasks. The deep integration of PCG and DR enables a more meaningful form of evaluation in the form of a direct measurement of generalization capabilities across a wide distribution of unseen scenarios. This allows the community to move beyond asking whether an agent can solve a task and towards answering the more critical question about the robustness of its solution to novelty and uncertainty. Quantifying this generalization gap is fundamental to building trust in learning-based systems for safety-critical applications.

5

Achieving Adaptive Autonomy

With SRB established as the foundational framework for this investigation, the focus shifts from the environment where an agent learns to the principles of the learning process itself. The availability of a diverse, scalable, and realistic testbed provides the necessary foundation to move beyond baseline applications and enables a systematic investigation into the fundamental question of how an agent forges the skills required for adaptive autonomy. This chapter presents the core methodological contributions of this thesis, detailing the inquiry into the learning paradigms, control representations, and perceptual strategies that together enable the development of robust, generalizable, and physically compliant robotic behaviors. This chapter directly addresses the methodological limitations of sample inefficiency and rigid kinematic control identified in the foundational grasping study, presenting the advanced learning framework developed to overcome them.

Achieving this level of autonomy requires a holistic approach that extends beyond the selection of a single RL algorithm. Therefore, this chapter systematically deconstructs the agent environment interaction loop to construct a complete methodological blueprint. The first component is the learning paradigm, where an empirical comparison reveals the critical role of world modeling. The next component addresses the influence of perception, establishing a pragmatic approach for focused research. The central pillar of this framework, however, is the redefinition of how a learned agent physically interacts with its world. This work moves past the brittleness of standard kinematic control to introduce a framework for learning compliant manipulation with MBRL through OSC. Finally, the blueprint accounts for the influence of embodiment, showing how this adaptive methodology allows agents to develop strategies that are sensitive to their own physical morphology. This integrated methodology constitutes the workflow for learning adaptive control that is designed to produce a policy capable of succeeding in the unpredictable domain of space.

5.1 Learning Paradigm

The foundation of any autonomous agent is the learning paradigm that governs its acquisition of knowledge and skill. This choice dictates how the agent processes information, explores its environment, and generalizes from its experience. This section opens with a systematic, empirical comparison of different RL paradigms to establish the most effective approach for the challenging scenarios presented by SRB.

5.1.1 Algorithmic Comparison and Baselines

To ground the discussion of learning methodologies in empirical evidence, this research conducted a thorough comparison of algorithms representing the three main paradigms of deep RL: on-policy model-free (PPO), off-policy model-free (SAC, TD3), and model-based (DreamerV3).

Focused Comparison on a Representative Task

The initial comparison focused on the procedurally generated peg-in-hole assembly modules shown in Figure 5.1, a prototype challenge in contact-rich manipulation detailed in **Publication II**. To isolate the core challenges of precision and generalization that are independent of manipulator dynamics, the task was constrained to a simulation with direct control of the peg trajectory in Cartesian space. The problem was formulated as a POMDP due to the agent having no direct access to latent physical properties like the precise geometry of the procedurally generated pegs, material friction, or contact forces.

Figure 5.1 – Procedurally generated assembly modules used in the peg-in-hole task to create a diverse training and evaluation distribution.

As illustrated in Figure 5.2, the observation space was constructed from the relative transformations between the peg and the hole, and the agent learned to output target linear and angular velocity commands.

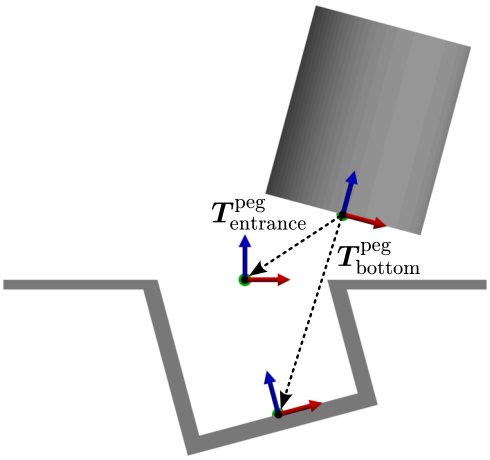


Figure 5.2 – The observation space of the peg-in-hole task agents is defined by the $T^{\text{peg}}_{\text{entrance}}$ and $T^{\text{peg}}_{\text{bottom}}$ transformations that capture the kinematic state of the assembly.

This setup, combined with 1024 parallel environments featuring unique PCG modules, provided a challenging testbed and a clear initial hierarchy of algorithmic performance, as shown in Figure 5.3. The MBRL agent, DreamerV3, demonstrated its superior sample efficiency by converging to a high-performance policy significantly faster than its model-free counterparts.

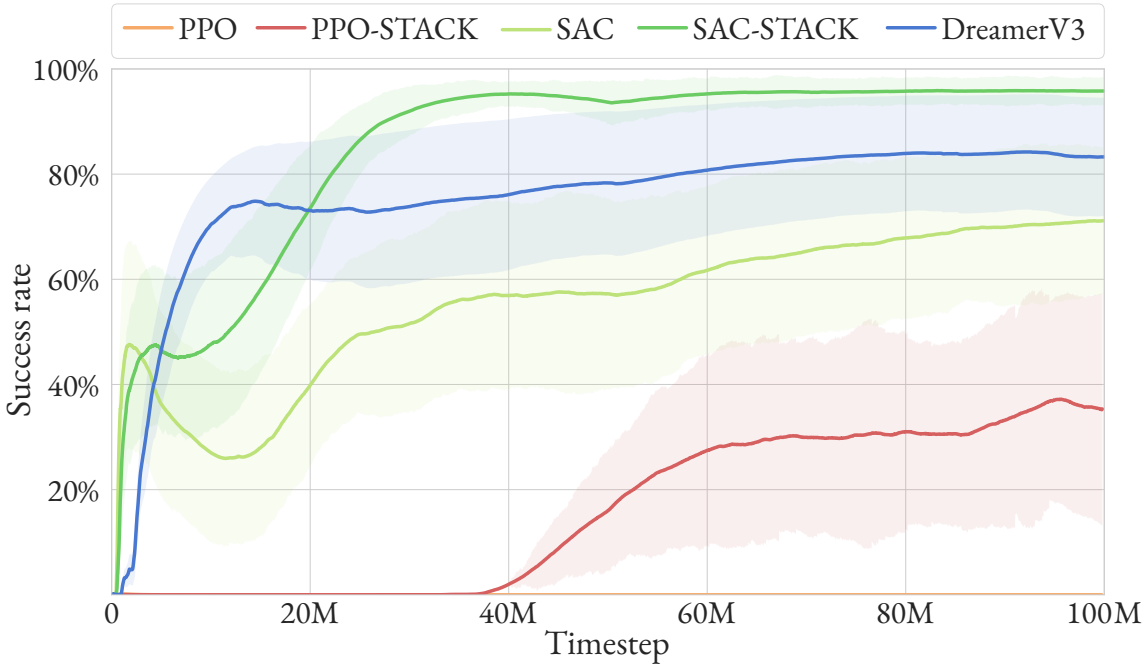


Figure 5.3 – Learning curves for the peg-in-hole assembly task from **Publication II**, comparing model-free (PPO, SAC) and model-based (DreamerV3) algorithms.

The performance of the model-free methods was directly tied to their ability to handle the task's partial observability. While the standard PPO algorithm struggled, a variant incorporating a history of past observations (PPO-STACK) learned a moderately successful policy. Similarly, the performance and stability of SAC were improved by its SAC-STACK counterpart, underscoring the critical role of temporal context for model-free agents. As shown in Figure 5.4, the DreamerV3 agent not only learned fastest but also achieved the quickest task completion times. This is a direct result of its inherent ability to handle temporal dependencies through its recurrent world model, which provides a more powerful mechanism for memory than explicit observation stacking.

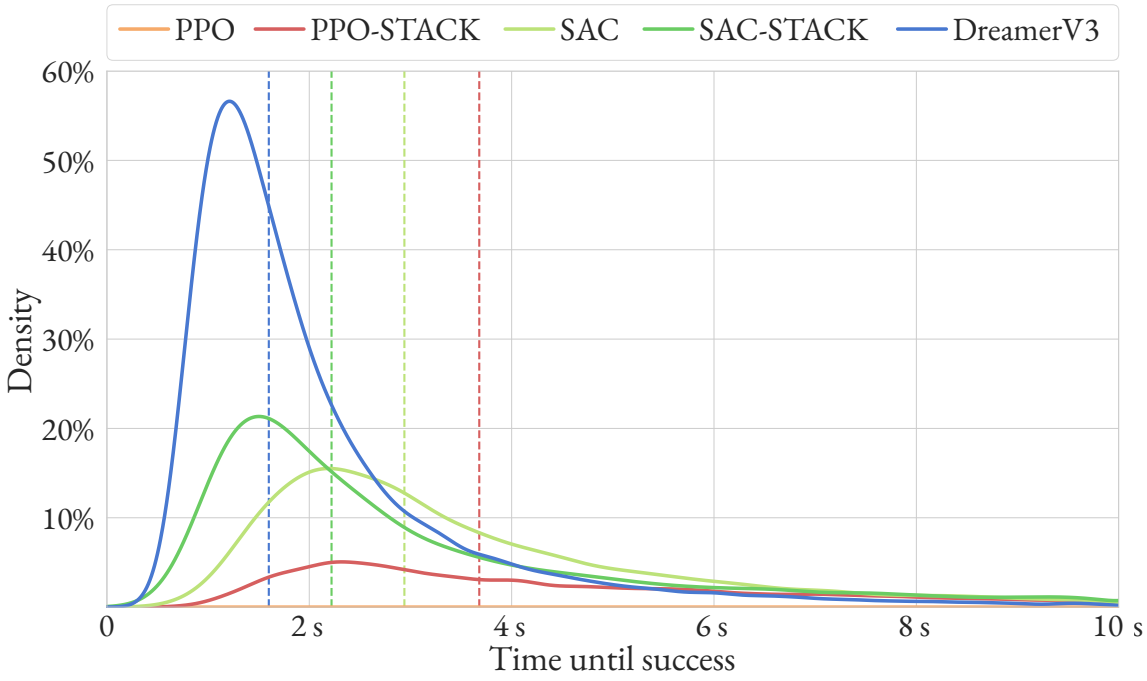


Figure 5.4 – Time until successful completion for the peg-in-hole assembly task. DreamerV3 demonstrates the fastest completion times across all scenarios, followed by the SAC variants.

The most critical distinction, however, emerged during evaluation on a test set of unseen procedural modules. Here, the DreamerV3 agent demonstrated robust generalization by maintaining its high success rate, while the SAC variants exhibited a significant performance degradation, highlighting their greater tendency to overfit to the training distribution.

Baselines for the Space Robotics Bench

The performance hierarchy established in the focused study was decisively confirmed by a large-scale evaluation across the full suite of tasks in SRB, as presented in **Publication VII**. The results, summarized in Figure 5.5, consistently demonstrated the superior performance

of the MBRL paradigm. Across nearly all tasks, from `landing` to `sample_collection`, the DreamerV3 agent achieved the highest episodic returns and success rates.

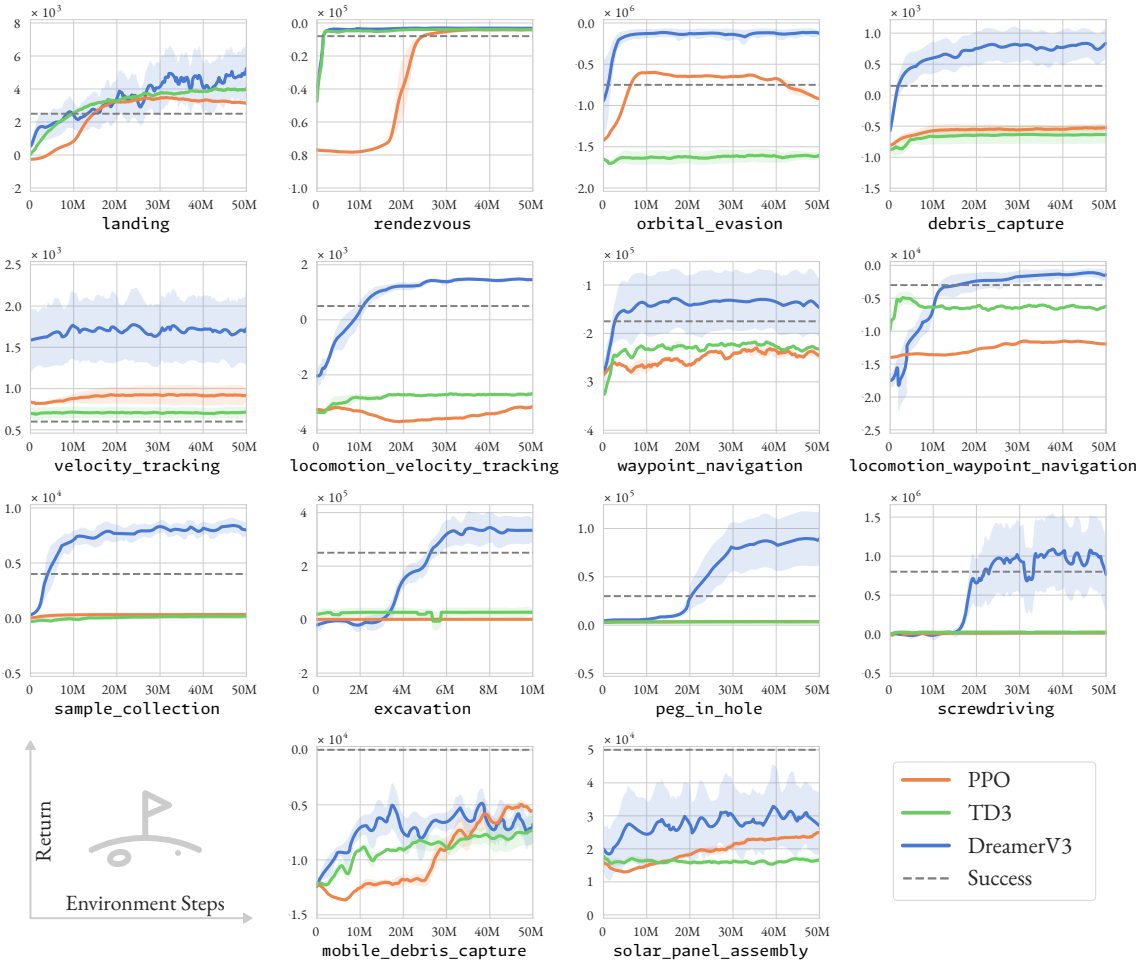


Figure 5.5 – RL baselines across SRB tasks from **Publication VII**. The results compare the performance of PPO, TD3, and DreamerV3 agents. DreamerV3 consistently achieves the highest performance across the majority of tasks.

However, this performance comes at a greater computational cost in terms of wall-clock training time. With our setup and 32 updates per environment step, DreamerV3 required on average 5.3× longer to train than PPO for the same number of environment steps. This trade-off positions MBRL as a powerful but resource-intensive paradigm, particularly suitable for complex problems where physical simulation or real-world interactions are the primary bottleneck. This large-scale evaluation also clearly delineates the frontier of current capabilities, as the most complex long-horizon tasks like `solar_panel_assembly` remained unsolved by any of the tested algorithms.

5.1.2 Importance of World Modeling

The consistent performance advantage of the DreamerV3 agent is a direct consequence of its core component, namely the learned world model [33]. As depicted in Figure 5.6, this learned model provides two profound advantages. First, it dramatically improves sample efficiency by allowing the agent to train on vast amounts of imagined experience, a process orders of magnitude faster than interacting with the full physics simulation. Second, the world model learns an abstract representation of the environment. It learns to filter out the superficial sensory details introduced by DR and PCG to focus on the latent dynamics essential for prediction. This learned abstraction is a powerful, implicit mechanism for generalization, forcing the policy to become robust to the immense diversity of the training distribution.

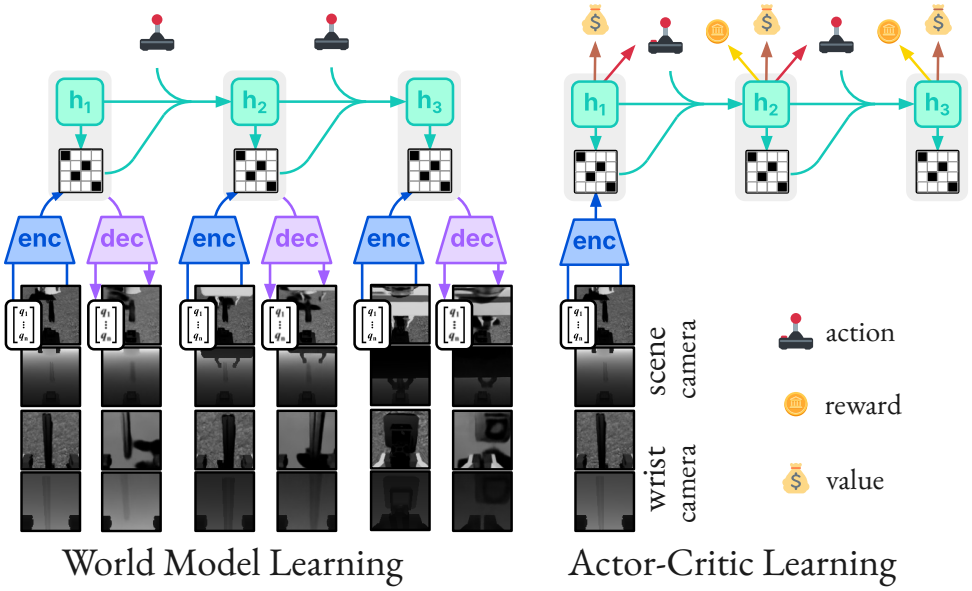


Figure 5.6 – The concept of learning in imagination, where the agent uses its world model to predict sequences of future latent states and rewards. The actor-critic networks are trained entirely on these imagined trajectories, which dramatically improves sample efficiency.

5.1.3 Benefits of Recurrent Architecture

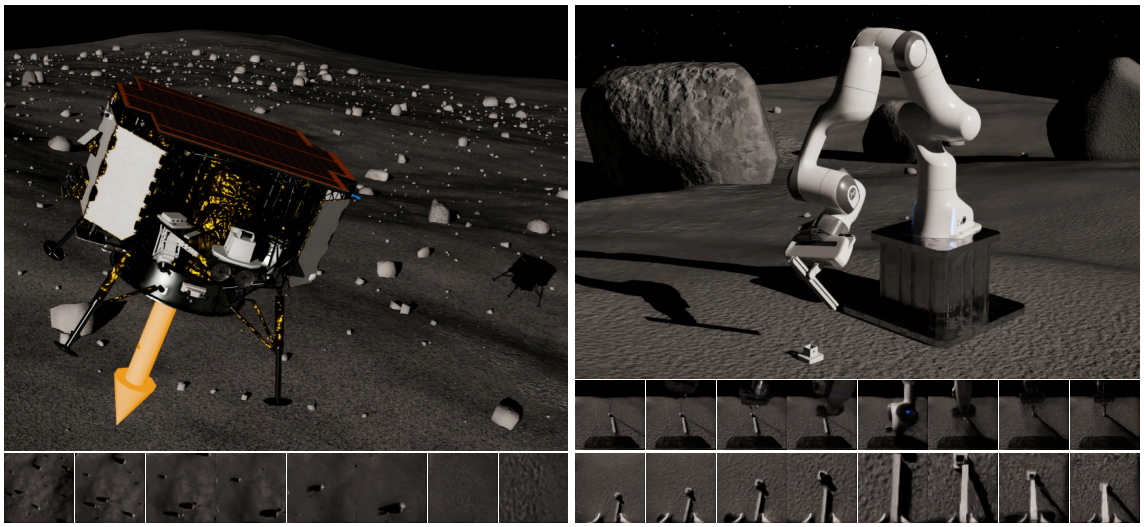
The world model within the DreamerV3 agent integrates a recurrent neural network (RNN). This structure is essential for addressing the partial observability inherent in all the benchmark tasks, as the agent never has access to the complete ground truth state of the environment. The recurrent state of the world model functions as the agent's memory, integrating the current observation with its previous state to build a more complete and temporally coherent understanding of the environment's hidden dynamics. This ability to reason about the world over time is critical for success and explains why adding observation stacking also improved the performance of the model-free RL agents in the peg-in-hole task from **Publication II**.

5.2 Role of Perception

The ability of an autonomous agent to act effectively is fundamentally constrained by its ability to perceive its environment. This section investigates the challenges of learning directly from raw visual data, a critical step towards creating truly end-to-end autonomous systems.

5.2.1 Learning from Pixels

Learning directly from high-dimensional visual observations is a significant challenge, requiring the agent to simultaneously learn a meaningful representation of the world from raw pixel data in addition to a control policy. To explore this within SRB, experiments were conducted on the `landing` and `peg_in_hole` tasks using only rendered RGB images as input, with camera perspectives shown in Figure 5.7.



(a) `landing_visual` (b) `peg_in_hole_visual`

Figure 5.7 – Camera perspectives for the end-to-end learning. The `landing` task uses a bottom-mounted camera, while the `peg_in_hole` task uses wrist- and base-mounted cameras.

The results, summarized in Table 5.1, demonstrated that learning from pixels is indeed feasible, but performance was lower than that of agents trained with access to privileged state information. The learning process was slower, less stable, and resulted in a lower final success rate. These findings highlight the immense challenge that visual complexity adds to the learning problem. They also underscore the value of state-based observations as a tool for research by decoupling the perception and control problems.

Table 5.1 – Success rates of end-to-end policies trained with different sensory modalities.

Task	State	Proprioception	Visual
`landing`	63.2%	11.4%	47.6%
`peg_in_hole`	97.8%	0.0%	73.2%

5.3 Learning Adaptive Compliance

For the complex manipulation tasks central to this thesis, actions result in physical, contact-rich interactions. This section details the investigation into control representations that enable skillful and safe interaction, moving from the inherent limitations of rigid control to a more sophisticated framework of learned, adaptive compliance.

5.3.1 Brittleness of Rigid Kinematic Control

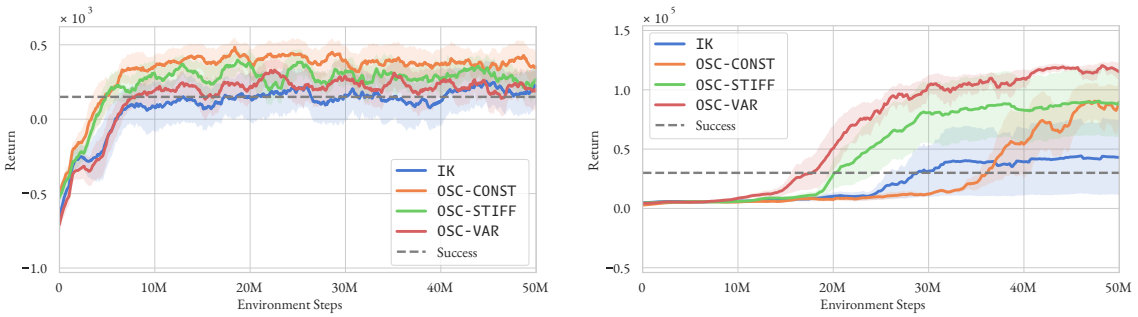
The standard actuation model for many robotic manipulators is differential IK. While effective for motion in free space, this position-based approach is fundamentally non-adaptive when faced with physical contact, attempting to follow a commanded trajectory regardless of external forces. This rigidity is a critical failure point in unstructured environments. If an agent attempts to insert a peg with a slight misalignment, the rigid controller can generate an uncontrolled escalation of contact forces, leading to jamming, hardware damage, or mission failure. This inherent brittleness, first identified as a key risk in the foundational grasping study, necessitates a control paradigm that can gracefully and intelligently manage physical contact.

5.3.2 Operational Space Control as a Compliant Framework

To overcome the brittleness, this research adopted operational space control (OSC) [34]. This framework formulates the control problem directly in the task space and models the EE as a programmable mass-spring-damper system, providing a principled method for implementing software-defined compliance. The first step of the investigation was to confirm the hypothesis that even passive compliance offers an advantage. To this end, initial experiments compared the rigid `IK` baseline against an `OSC-CONST` strategy, which used OSC with fixed, hand-tuned gains. As shown in the learning curves of Figure 5.8, the results were immediate and conclusive. Across contact-rich tasks, the `OSC-CONST` agent demonstrated significantly more stable learning and achieved higher final returns than its rigid counterpart. This confirmed that even a constant, passive level of compliance is a powerful first step toward robust interaction.

5.3.3 Learning to Dynamically Modulate Stiffness and Damping

While fixed compliance provides some benefits, the ideal level of compliance is not constant but dynamic and task-dependent. For example, a manipulator should be stiff and precise during free-space motion but become soft and yielding upon making contact to guide a part into place. This observation led to the central methodological combination of this thesis for empowering an MBRL agent to learn its own adaptive compliance strategy.



(a) `debris_capture` (b) `peg_in_hole`

Figure 5.8 – Learning curves comparing different control strategies on the `debris_capture` and `peg_in_hole` tasks. The adaptive OSC variants (`OSC-STIFF` and `OSC-VAR`) demonstrate more stable convergence and higher final returns than the rigid `IK` approach.

This was achieved by augmenting the agent's action space. In addition to commanding the desired EE motion, the agent also learned to output the desired stiffness and damping gains for the OSC controller at each timestep. Two adaptive strategies were investigated: `OSC-STIFF` , where the agent modulated only the six stiffness (K_p) gains, and `OSC-VAR` , where it learned to control all twelve stiffness (K_p) and damping (K_d) gains, as illustrated in Figure 5.9.

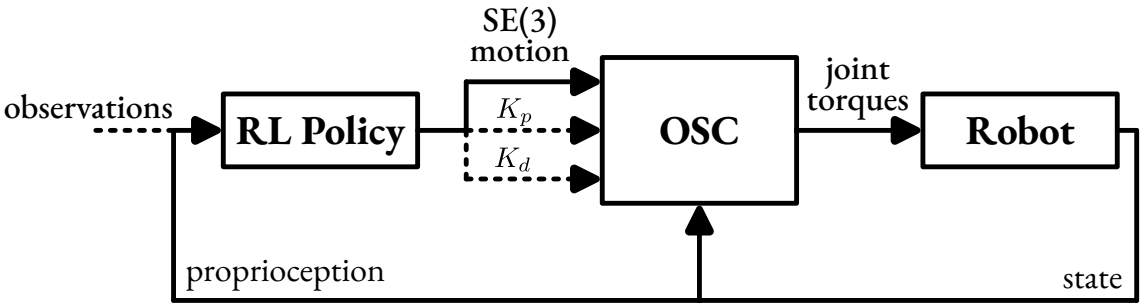


Figure 5.9 – Conceptual illustration of adaptive compliance through OSC. The RL agent learns to dynamically modulate the EE's stiffness and damping parameters in addition to its SE(3) motion commands, enabling it to adapt its physical interaction strategy in real-time.

The performance metrics, summarized in Table 5.2, further reinforce these findings. The data reveals a clear pattern of improvement when agents are empowered with adaptive compliance.

Table 5.2 – Relative performance of DreamerV3 using rigid IK and adaptive OSC controllers. The results are normalized with respect to the IK baseline for numerical clarity.

Task	Controller	Return	Motion Jerk
`debris_capture`	`IK`	**1.00**	1.00
	`OSC-VAR`	0.97	**0.42**
`peg_in_hole`	`IK`	1.00	1.00
	`OSC-VAR`	**1.10**	**0.50**

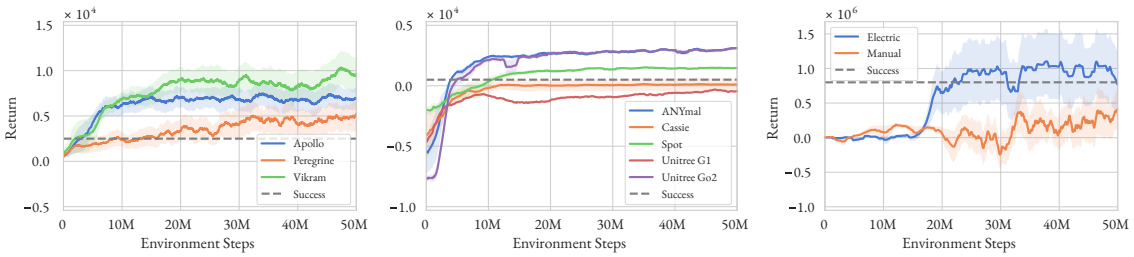
The results of this investigation were conclusive. As shown in Figure 5.8, the agents with learned compliance (`OSC-STIFF` and `OSC-VAR`) consistently outperformed both rigid and fixed-compliance strategies. This superiority was validated across distinct manipulation challenges:

- For the delicate `debris_capture` task, where minimizing disturbances is paramount, the adaptive agent learned a safer, more conservative strategy, achieving a 58% reduction in motion jerk compared to the rigid `IK` controller.
- For the high-precision `peg_in_hole` task, the ability to compliantly negotiate contact states resulted in a 10% higher final success rate, measured via normalized return.

By learning to dynamically modulate its own compliance, the agent discovered sophisticated, emergent strategies analogous to human motor intelligence, such as using high stiffness for rapid approaches, then softening upon contact to gently guide parts into place. This cohesive result demonstrates that learned adaptive compliance is a critical capability for achieving robust, safe, and effective physical interaction in the unpredictable environments of space.

5.4 Influence of Embodiment

The behavior of an autonomous agent is shaped not only by its learning algorithm and control representation but also by its physical form, or embodiment. The modular design of SRB provides a unique opportunity to investigate this interplay. Experiments comparing different robot morphologies on several key tasks revealed that physical design creates strong priors that either facilitate or hinder the learning process. In the `locomotion_velocity_tracking` task, quadrupedal robots consistently learned stable gaits more quickly than bipedal robots, suggesting their inherent static stability provides a simpler learning landscape. In the `landing` task, performance varied significantly between different spacecraft designs. These results, with learning curves shown in Figure 5.10, underscore that optimal performance is achieved when the hardware embodiment and the learned control policy are well-matched. This highlights that embodiment is a key factor, and a truly general agent must learn policies that can generalize not just across environments, but also across variations in its own physical form. This principle of embodiment-awareness finds its most compelling validation in the tool-aware manipulation case study of the next chapter.



(a) `landing` (b) `locomotion_vel_track` (c) `screwdriving`

Figure 5.10 – Learning curves comparing the performance of different robot morphologies.

5.5 Blueprint for Adaptive Control

The investigations detailed throughout this chapter distill into a cohesive blueprint for achieving adaptive control in complex robotic systems. This blueprint is not a single algorithm but a methodological framework that combines several key principles.

The first principle is the adoption of a model-based learning paradigm. The evidence strongly suggests that for complex tasks with partial observability, an agent that learns a predictive world model like DreamerV3 offers superior data efficiency, generalization, and planning capabilities.

Building upon this predictive model, the second principle is to embrace learned compliant control. The brittleness of pure position-based controllers makes them unsuitable for reliable interaction. This blueprint, therefore, prescribes using a compliant control framework like OSC and, crucially, empowering the agent to learn not just where to move but how to physically behave by dynamically modulating its own compliance parameters.

Underpinning this entire framework is the final principle of training with maximum experiential diversity. The ability of an agent to develop generalizable and adaptive strategies is directly proportional to the richness of its training distribution. SRB, with its procedural engine SimForge, is the practical embodiment of this principle and provides the necessary experiential foundation for the learning methodologies to succeed.

6

Empirical Validation

The preceding chapters have established the theoretical foundations, developed the simulation framework, and formulated a blueprint for achieving adaptive autonomy. This chapter now moves from principles to practice to provide an empirical validation of the proposed system. Its purpose is to demonstrate, through a series of in-depth case studies, that the integrated framework of SRB and the adaptive control methodology can successfully solve a range of challenging, mission-relevant tasks.

This validation is not a simple measure of task success under a single set of conditions. Instead, each scenario is designed to specifically probe the key capabilities central to this thesis, namely generalization, adaptation, and robustness in the face of uncertainty. By leveraging the full power of PCG and DR within SRB, these experiments are designed to answer the critical question of whether the proposed methodology produces agents that can reliably perform complex tasks not just in a familiar setting, but across a wide distribution of previously unseen environments and configurations.

The chapter will present detailed results for three distinct scenarios, each chosen to highlight a different aspect of adaptive autonomy. The first case study provides a complete sim-to-real validation of adaptive traversal on unstructured terrain. The second study delves into the complexities of contact-rich interaction through tool-aware resource excavation. The final study offers a unique validation of the core diversity principle in a competitive, high-speed, and adversarial domain. The quantitative results and qualitative behaviors observed in these experiments provide the evidence supporting the core claims of this thesis, showcasing the practical effectiveness of the developed framework in preparing autonomous systems for the unpredictable challenges of space.

6.1 Adaptive Traversal on Unstructured Terrain

Reliable autonomous navigation across the unstructured terrains of distant planetary surfaces is a critical enabler for future space exploration. This first and most comprehensive case study addresses this foundational challenge by presenting a complete sim-to-real validation of the entire thesis framework. It details the successful development and deployment of a robust control policy for dynamic waypoint tracking on challenging granular surfaces, as presented in **Publication V**.

The core hypothesis of this validation is that policies trained with extensive procedural diversity can be transferred zero-shot to a physical rover and achieve robust, high-performance navigation to successfully bridge the sim-to-real gap for a task dominated by complex contact dynamics. The sim-to-real experiment, conceptually outlined in Figure 6.1, leverages the full capabilities of SRB. It uses massively parallel simulation to train RL agents across a vast distribution of procedurally generated environments with randomized physics. Furthermore, it explores the benefits of high-fidelity particle simulation for modeling granular media. The resulting policies are then deployed directly without any real-world fine-tuning onto a physical wheeled rover operating in the LunaLab, which is a lunar-analogue facility filled with basalt gravel that was also used in Chapter 3.

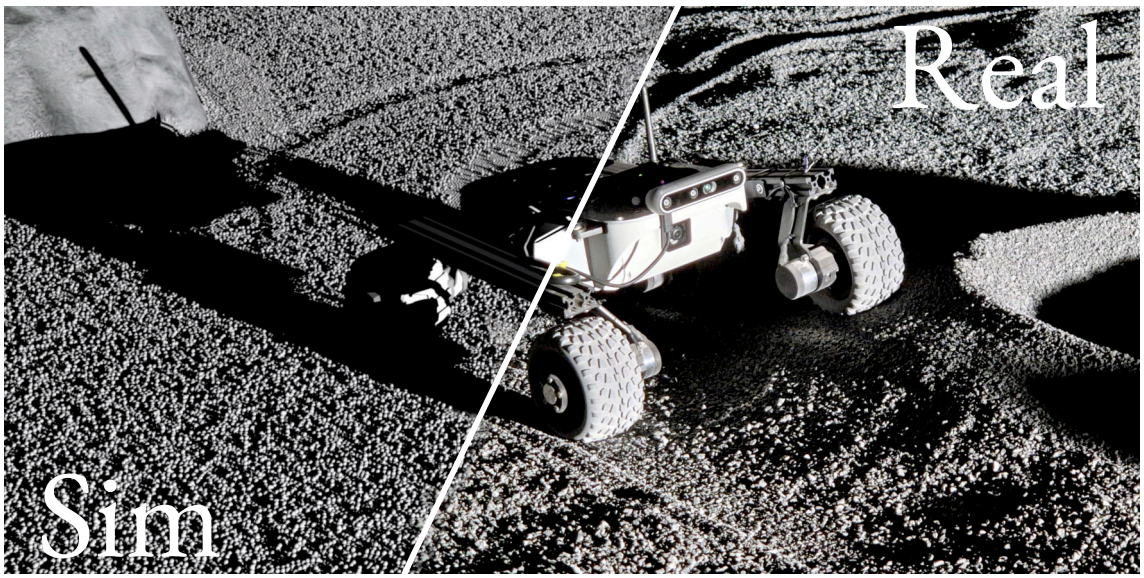


Figure 6.1 – The conceptual parallel for the adaptive traversal case study from **Publication V**, where agents are trained in SRB to track dynamic waypoints across diverse, procedurally generated scenarios. The generalization learned from this experience enables the acquired policies to be transferred to a physical rover operating on granular media in a lunar-analogue facility.

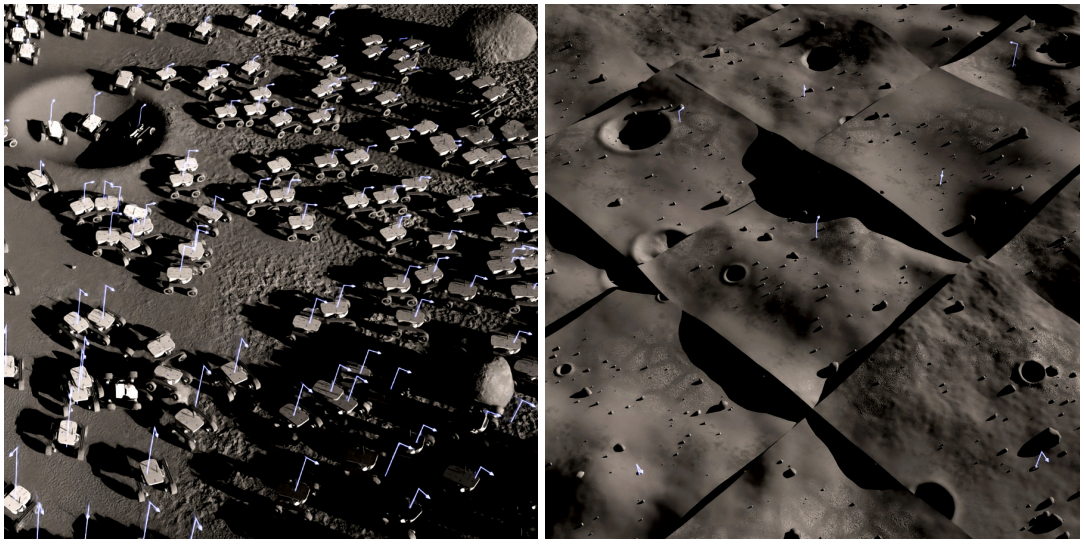
6.1.1 Experimental Setup and Methodology

The validation was built upon an integrated framework combining a powerful simulation environment for training, a realistic physical testbed for evaluation, and a mission-relevant control task.

Simulation Framework and Training Regimes

All policy development occurred within SRB. The primary training methodology leveraged the massive parallelization capabilities of the framework, running 512 environment instances on rigid surfaces simultaneously to generate the vast amount of experience required for RL. To test the core hypothesis of this thesis, two distinct training regimes were explored, as illustrated in Figure 6.2:

- **Stacked** regime served as a baseline representing a more traditional approach. All parallel environment instances were trained on a single and shared PCG terrain. This setup risks policy overfitting to the specific features of that one scenario.
- **Procedural** regime, embodying the core philosophy of this thesis, provided each of the 512 parallel environment instances with its own unique PCG terrain. This was designed to force the agent to learn a generalizable navigation strategy, rather than memorizing paths on a single map.



(a) Stacked regime (b) Procedural regime

Figure 6.2 – Parallel simulation regimes of SRB. In the stacked regime, all agents share a single static terrain, risking overfitting. In the procedural regime, each parallel environment instance is exposed to a unique terrain to foster generalization.

Beyond this structural diversity, extensive DR was employed to enhance policy robustness. At the start of each training episode, key parameters were varied, including the gravity vector and small offsets of the rover's base frame to account for manufacturing variations. Crucially, randomized noise and variable delays were injected into both actions and observations to build resilience to the unpredictable latencies and sensor inaccuracies inherent in a physical system. The experimental framework also supports a high-fidelity physics mode, shown in Figure 6.3, which models the granular media as millions of discrete particles. Due to its computational expense, this mode was reserved for a specialized fine-tuning experiment.

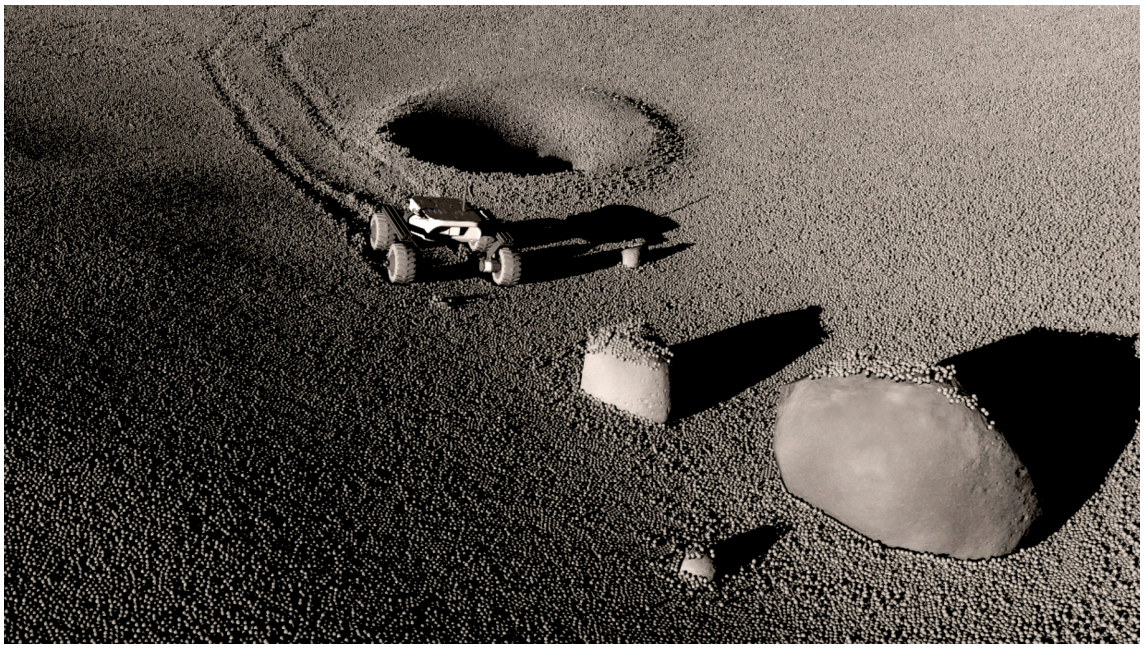


Figure 6.3 – High-fidelity simulation environment with millions of discrete particles used for fine-tuning. This setup provides a more realistic model of wheel-regolith interaction dynamics.

Physical Testbed and Task Formulation

All real-world validation was conducted in the LunaLab, a lunar-analogue facility at the University of Luxembourg containing 20 tons of basalt gravel that emulates the properties of regolith [37]. The robotic platform was the Leo Rover, a four-wheeled skid-steer mobile robot, shown in Figure 6.4. For ground-truth localization, an OptiTrack motion capture system was used. This was a critical methodological choice, as it supplied the high-frequency pose data needed for both real-time control and post-experiment analysis, allowing for the evaluation of the policy's performance independent of any potential state estimation errors.

Figure 6.4 – The real-world validation setup in the LunaLab facility with a Leo Rover.

The agent's objective was to master dynamic waypoint tracking, formulated as a POMDP. The policy operated at 25 Hz, receiving the relative 2D position and yaw to the target and outputting linear and angular velocity commands. The reward function was carefully shaped to encourage a sequence of behaviors in the form of a continuous penalty on distance to guide the general approach, rewards for precise alignment that became dominant near the target, and penalties on large action changes to encourage smooth and stable tracking.

6.1.2 Algorithmic Comparison and Selection

The investigation first identified the most suitable RL algorithm for zero-shot transfer in this dynamic control task. Four distinct algorithmic paradigms were evaluated: on-policy PPO, a recurrent variant of PPO with long short-term memory (LSTM) [98] network, off-policy TD3, and model-based DreamerV3. Each agent was trained for a comprehensive duration in 512 unique parallel environments in the procedural regime to ensure a fair test of generalization. The learning curves from simulation, shown in Figure 6.5, provided the first piece of evidence. DreamerV3 demonstrated vastly superior sample efficiency, converging to a higher and more stable final episodic return in only 20 million steps, compared to the 100 million steps required by the model-free agents.

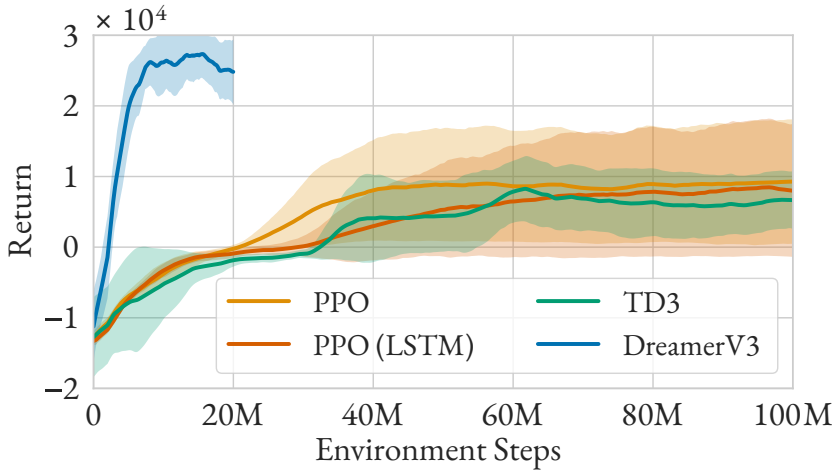


Figure 6.5 – Learning curves of the evaluated RL algorithms from **Publication V**. The results are averaged over five random seeds, with shaded regions representing the standard deviation. DreamerV3 demonstrates superior sample efficiency and achieves a higher final return.

This simulated performance translated directly to the physical world. The quantitative sim-to-real results, presented in Table 6.1, decisively confirm the superiority of the DreamerV3 agent. It achieved a substantially lower average tracking error (ATE) across all tested velocities, with an error rate that was often less than a quarter of its model-free counterparts. While PPO offered the lowest inference latency, a critical consideration for resource-constrained flight hardware, the vastly superior real-world tracking performance of DreamerV3 made it the clear choice for this mission-critical application. The training and evaluation were conducted on a workstation with an AMD Ryzen 9 7950X CPU and an NVIDIA RTX 4090 GPU. While this high-performance hardware acceleration is not yet space-qualified, this limitation is being continuously addressed by rapid advancements in onboard computing [99].

Table 6.1 – A comparison of the sim-to-real transfer ATE performance, training duration, and inference latency for policies trained with different RL algorithms.

	PPO	**PPO (LSTM)**	**TD3**	**DreamerV3**
5 cm/s	13.2 cm | 7.8°	11.4 cm | 4.8°	12.6 cm | 8.5°	**2.3 cm | 1.7°**
15 cm/s	13.7 cm | 8.6°	11.2 cm | 8.1°	11.6 cm | 6.2°	**3.3 cm | 1.9°**
25 cm/s	14.8 cm | 8.7°	12.9 cm | 9.9°	13.1 cm | 9.1°	**3.6 cm | 2.3°**
Training	13.5 h (100M)	25.0 h (100M)	15.0 h (100M)	17.5 h (20M)
GPU Inference	**0.42 ± 0.1 ms**	0.71 ± 0.2 ms	0.43 ± 0.1 ms	1.27 ± 0.1 ms
CPU Inference	**0.24 ± 0.1 ms**	0.71 ± 0.2 ms	0.43 ± 0.1 ms	2.38 ± 0.2 ms

The quantitative superiority is visually apparent in the qualitative trajectory plots in Figure 6.6. The path traced by the DreamerV3 policy is smooth, precise, and closely aligned with the target trajectory. In contrast, the other agents exhibit large deviations and erratic behavior, clearly demonstrating a failure to generalize to the complex dynamics of the physical world.

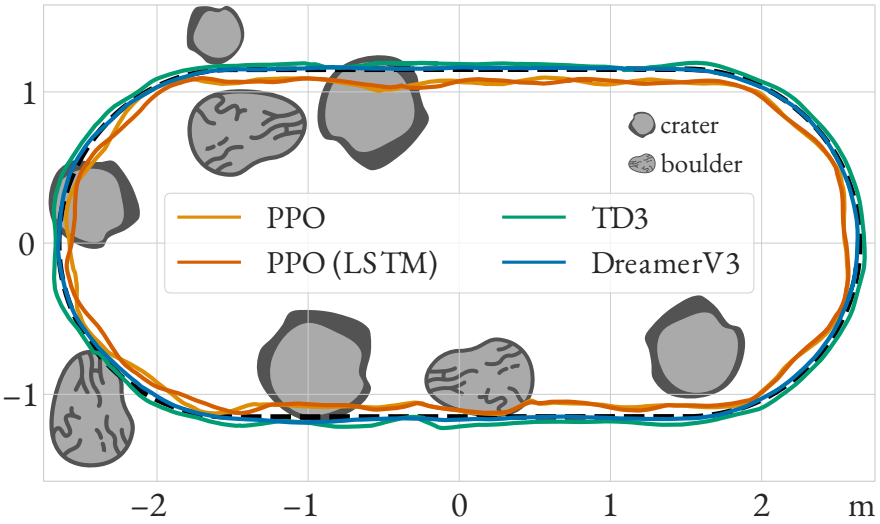


Figure 6.6 – Real-world trajectories of the physical rover following a capsule-shaped path, controlled by policies trained with different RL algorithms.

Given its overwhelming advantages in both sample efficiency and real-world performance, DreamerV3 was selected as the exclusive algorithm for all subsequent experiments in this study. The high quality of the learned controller was further demonstrated by deploying it on a series of more complex paths, as shown in Figure 6.7, and by observing the highly repeatable tracks it imprinted in the granular media, seen in Figure 6.8.

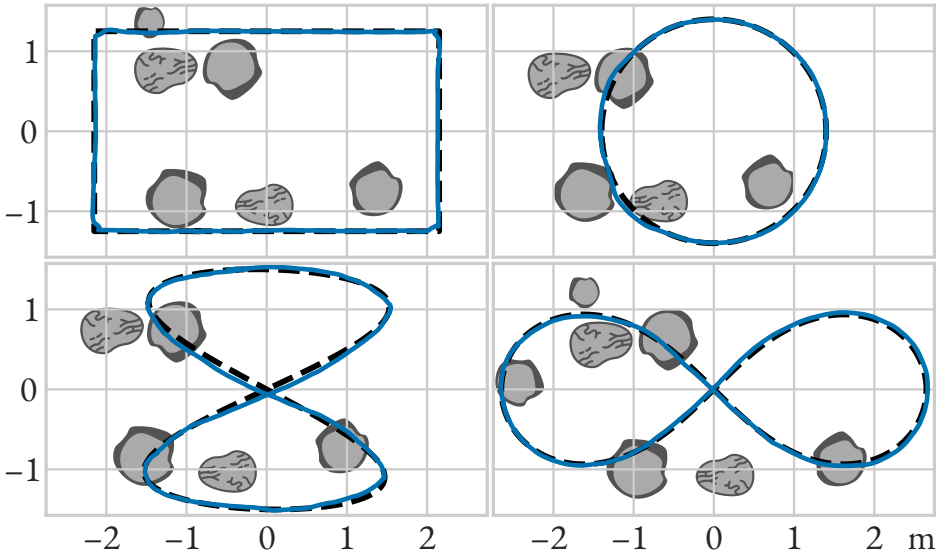


Figure 6.7 – Additional real-world trajectories executed by the DreamerV3 agent, including rectangular, circular, Lissajous, and lemniscate paths.

Figure 6.8 – Repeatable lemniscate path imprinted by the rover's wheels in the basalt gravel of the LunaLab during a real-world deployment.

6.1.3 The Critical Role of Procedural Diversity

The experiment then systematically evaluated the core hypothesis that policy robustness is a direct result of simulation diversity. Four distinct training regimes were compared. A baseline `static` agent was trained with all 512 parallel environments sharing a single terrain. A `DR` agent was trained on the same static terrain but with full DR. The `DR&PCG` agent was trained under the full procedural paradigm, with each instance featuring a unique terrain. A final `DR&PCG+PF` agent augmented this with a fine-tuning stage using high-fidelity particle physics.

The results, presented in Table 6.2, provide definitive evidence for the procedural paradigm. While the `static` agent learned the task in simulation, its performance on the physical rover was poor, exhibiting high tracking error and instability. The addition of `DR` alone provided a substantial improvement, particularly in orientation error, confirming that randomizing physics and noise is a critical first step. However, the best overall performance was achieved by the `DR&PCG` agent. By forcing the policy to generalize across a vast distribution of unique terrains, it learned the most robust strategy, achieving the lowest tracking error at higher, more challenging velocities. The final fine-tuning with particle physics for the `DR&PCG+PF` agent offered only a minor improvement at low speed for a significant additional training cost, indicating that broad structural diversity from PCG is a more critical and cost-effective factor for generalization than high-fidelity contact modeling for this task. This result provides empirical evidence for the central hypothesis of this thesis that policies trained with extensive procedural diversity do not merely perform better in simulation, but are fundamentally more robust and capable of successful zero-shot transfer to the physical world.

Table 6.2 – Sim-to-real transfer performance across different training regimes.

Velocity	Static	DR	DR&PCG	DR&PCG+PF
5 cm/s	3.4 cm | 4.2°	2.5 cm | 1.6°	2.3 cm | 1.7°	**2.2 cm | 1.5°**
15 cm/s	4.2 cm | 6.8°	3.3 cm | 2.3°	**3.3 cm | 1.9°**	3.3 cm | 2.0°
25 cm/s	4.4 cm | 7.1°	4.1 cm | 2.9°	**3.6 cm | 2.3°**	4.3 cm | 2.6°
Training	17.0 h (20M)	17.0 h (20M)	17.5 h (20M)	+82.0 h (+1M)

6.1.4 Ensuring Stability for Hardware Deployment

The final set of experiments within this case study addressed a practical engineering concern critical for any real-world mission in terms of the stability and smoothness of the learned controller. While RL optimization can produce highly performant policies, the resulting controllers often generate high-frequency and oscillating actions. These jerky commands, while potentially optimal for maximizing a reward signal in the discrete-time environment of a simulation, can lead to unstable behavior, cause excessive mechanical stress, and increase power consumption on physical hardware, thereby compromising the long-term reliability of the robotic system.

To investigate this issue and identify a practical solution, the performance of the raw, unfiltered DreamerV3 policy was compared against versions augmented with three different low-pass action filters. The selected filters represent common and computationally efficient approaches to signal smoothing:

- **Moving average** filter with a history window of 5 steps.
- **Savitzky-Golay** (third order) filter with a history window of 9 steps.
- **Butterworth** (third order) filter with a cutoff frequency of 2.5 Hz.

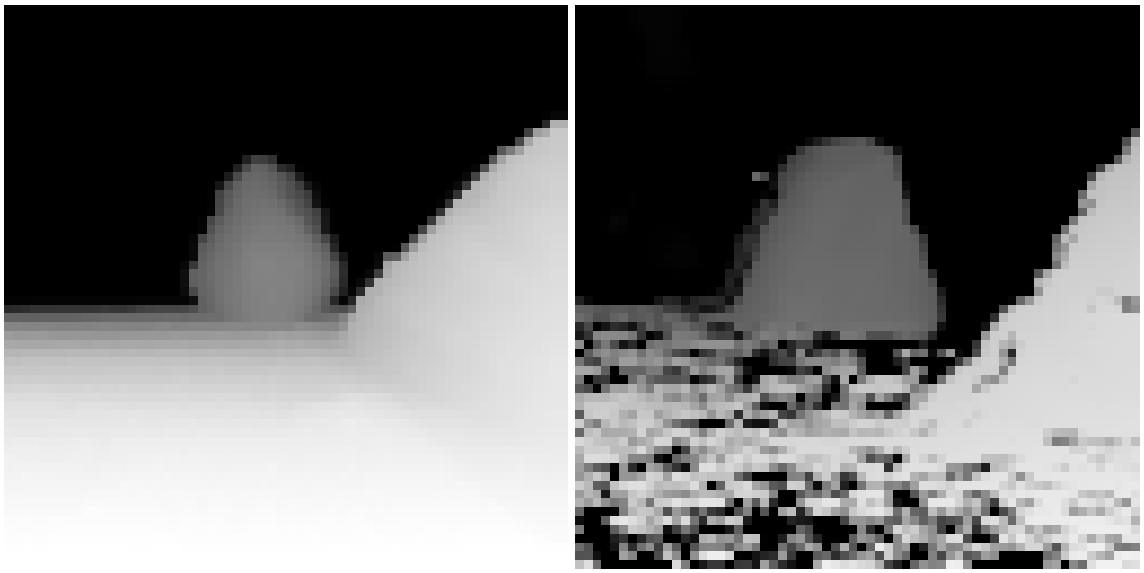
The results, presented in Table 6.3, reveal a critical trade-off between tracking precision and motion stability. The unfiltered policy, while achieving the best tracking accuracy at higher speeds, did so at a significant cost while producing motion three times jerkier than the filtered alternatives. All smoothing filters dramatically reduced motion jerk, but the Savitzky-Golay filter's large history window introduced a significant phase lag, leading to catastrophic instability at high speed. The simple moving average filter provided the most satisfactory compromise, substantially reducing motion jerk by 67% with only a minor and acceptable accuracy penalty at high speed. This experiment underscores a crucial lesson that real-world deployment of optimizing for raw performance metrics alone is insufficient. A well-tuned action filter represents a practical, computationally efficient, and effective method for achieving the control stability required for safe and reliable long-term robotic operation.

Table 6.3 – Sim-to-real performance and relative motion jerk of different action smoothing filters. While the unfiltered policy is the most accurate at high speeds, it produces significantly jerkier motion. The Savitzky-Golay filter fails at high speed due to phase lag.

Velocity	Unfiltered	Moving Avg.	Savitzky-Golay	Butterworth
5 cm/s	2.3 cm | 1.7°	**2.2 cm | 1.6°**	2.6 cm | 1.7°	2.8 cm | 1.7°
15 cm/s	**3.3 cm | 1.9°**	3.7 cm | 2.4°	5.0 cm | 2.3°	4.3 cm | 2.1°
25 cm/s	**3.6 cm | 2.3°**	4.2 cm | 2.1°	64.9 cm | 16.4°	4.9 cm | 2.4°
Motion Jerk	1.00	0.33	0.30	0.39

6.1.5 The Perceptual Sim-to-Real Gap

As a final investigation, this study explored the feasibility of learning an end-to-end policy directly from visual data. An agent was trained with access to a 64×64 px depth map from an onboard camera. While this policy could be transferred to the physical rover, its tracking accuracy was significantly degraded, with an ATE of 9.2 cm and 6.5° at 15 cm/s. The reason for this performance collapse became immediately apparent when comparing the simulated and real-world sensor data, as shown in Figure 6.9. The standard simulation provided a clean and ideal depth map, but the physical basalt gravel in the LunaLab created a noisy sensor signal with substantial dropouts due to its reflective properties and lack of significant texture when using the Intel RealSense D455 camera.



(a) Simulated depth map (b) Real-world depth map

Figure 6.9 – Comparison of simulated and real-world depth views. The simulated map is clean, while the real-world image suffers from significant noise and signal dropout due to the properties of the basalt gravel.

This result highlights a crucial limitation and a key area for future work. Even with a robust and well-generalized control policy, unmodeled sensor physics can create a severe perceptual sim-to-real gap that becomes the primary point of failure. It also serves as a strong justification for the methodological choice in this thesis to primarily focus on state-based observations for developing the core control and interaction methodologies. By decoupling the immense challenge of perception from the equally difficult challenge of control, it becomes possible to make systematic progress on the latter. Closing this perceptual gap through higher-fidelity sensor simulation and the development of perception algorithms robust to real-world noise remains a critical open problem for achieving true end-to-end autonomy.

6.2 Tool-Aware Regolith Excavation

Following the successful validation of adaptive traversal on physical hardware, this second case study delves into the complexities of contact-rich manipulation. It addresses a challenge central to the vision of ISRU with an essential task of autonomous regolith excavation. This task is foundational for future off-world construction and resource extraction, but it presents a formidable control problem. The interaction is dominated by the complex, unpredictable physics of granular media, where forces are difficult to model analytically. Furthermore, true autonomy demands versatility and a robot must be able to operate effectively with a variety of tools, whether due to mission requirements or unforeseen hardware degradation from the abrasive lunar environment.

This case study, based on the work in **Publication VI**, serves as a direct and rigorous validation of the adaptive control methodology proposed in Chapter 5. The core hypothesis is that an agent can learn a generalizable excavation skill by combining a predictive world model with learned adaptive compliance. The experiment is specifically designed to test whether training an agent on a procedural distribution of tool geometries forces it to develop a robust, tool-aware policy that can generalize to unmodeled tool changes.

6.2.1 Experimental Setup and Methodology

The experimental simulation setup for this validation was the `excavation` task within SRB. To accurately capture the complex physics of the problem, the regolith was modeled using a high-fidelity particle simulation based on extended position-based dynamics [91], as shown in Figure 6.10. This provided the realistic, non-linear force feedback necessary for learning a physically grounded skill.

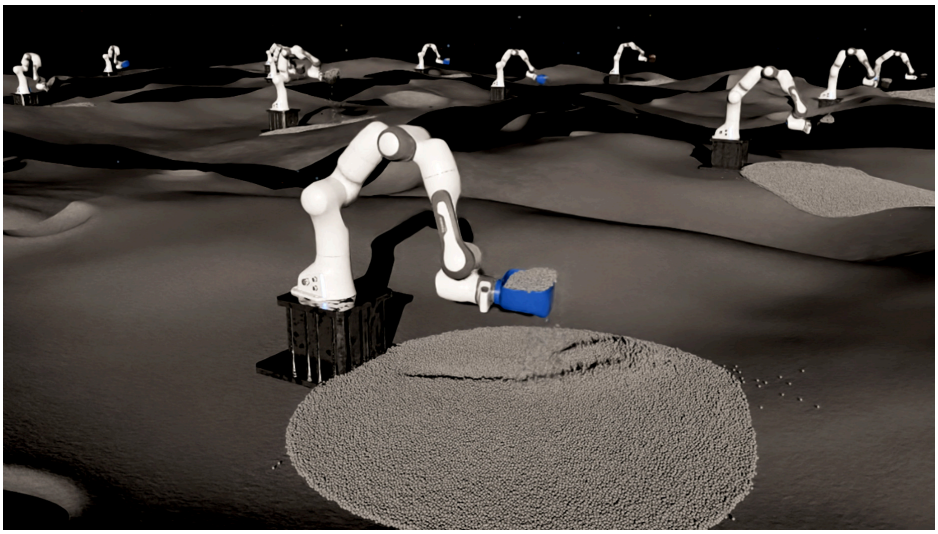
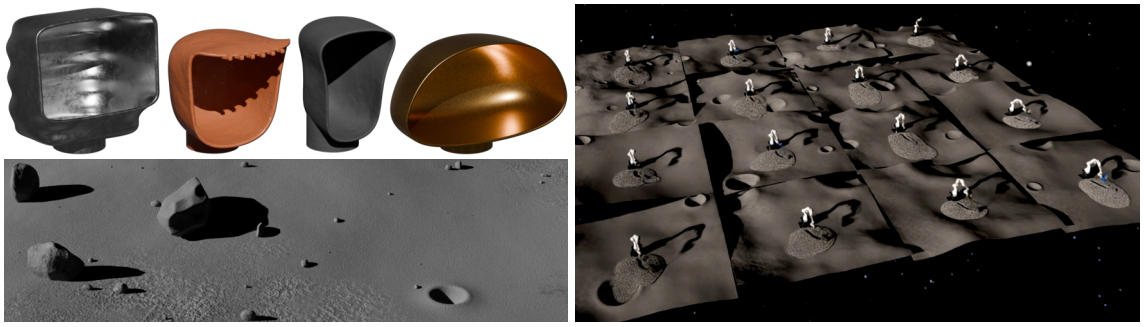


Figure 6.10 – A generalizable excavation skill is learned in SRB by training an agent on a vast distribution of randomized scenarios with procedurally generated tools.

The principle of procedural diversity was applied in two key ways, as shown in Figure 6.11. First, each of the 16 parallel training environments featured a unique lunar terrain generated by the SimForge engine. Second, and central to this study, the agent was equipped with a unique PCG excavation tool geometry.



(a) Procedurally generated assets (b) Parallel training environments

Figure 6.11 – The procedural paradigm for the `excavation` task, where PCG pipelines generate diverse terrains and tools that are then used to populate parallel environment instances.

The procedural pipeline for the tools, featuring over forty parameters, was designed to produce a wide spectrum of morphologies, varying scoop width, depth, curvature, and the number and shape of teeth. This forced the agent to learn an abstract understanding of excavation, rather than a strategy tied to a single tool. A variety of the generated tools is shown in Figure 6.12.

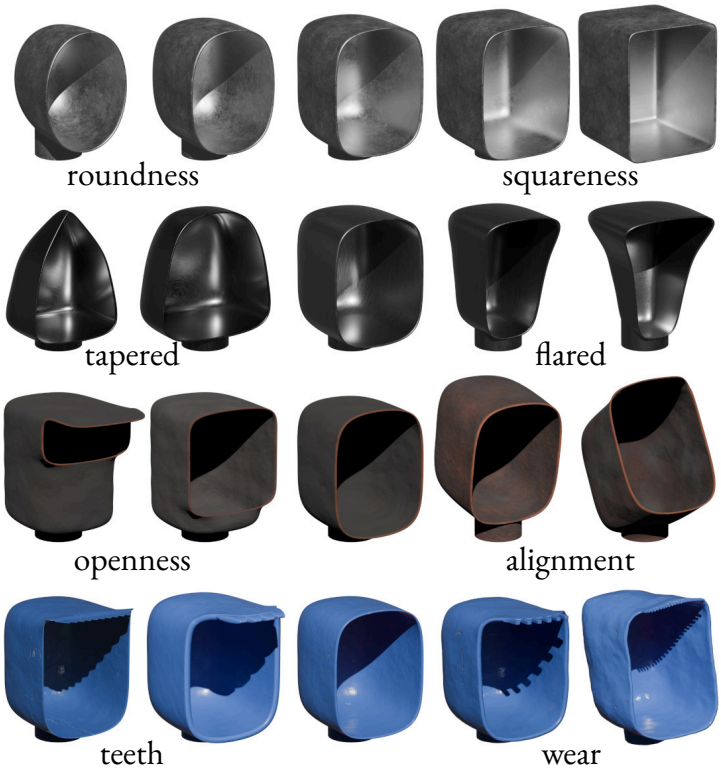


Figure 6.12 – A grid of PCG excavation tool geometries from **Publication VI**.

The agent controls a stationary Franka Emika manipulator and is trained using the DreamerV3 algorithm. The task is formulated as a POMDP, where the agent had no direct knowledge of its equipped tool's geometry or the specific physical properties of the regolith. Its action space was designed to directly test the adaptive compliance framework, combining SE(3) motion commands with learned stiffness (K_p) and damping (K_d) gains for OSC. The reward function was structured to incentivize lifting and stabilizing a large volume of regolith while penalizing undesirable behaviors such as dust generation and jerky motions.

6.2.2 Results and Analysis

The validation involved a systematic comparison of agents trained under different conditions to isolate the effects of compliance, procedural diversity, and perception. A baseline rigid `IK` agent was compared against several compliant agents. A `SPECIALIST` agent was trained with the full adaptive compliance model but only ever used a single, static tool geometry. In contrast, the generalist `ADAPTIVE` agent was trained on the full procedural distribution of tools. A final `VISUAL` agent was also trained, augmenting the `ADAPTIVE` agent's proprioceptive inputs with depth maps from two camera views, as shown in Figure 6.13.

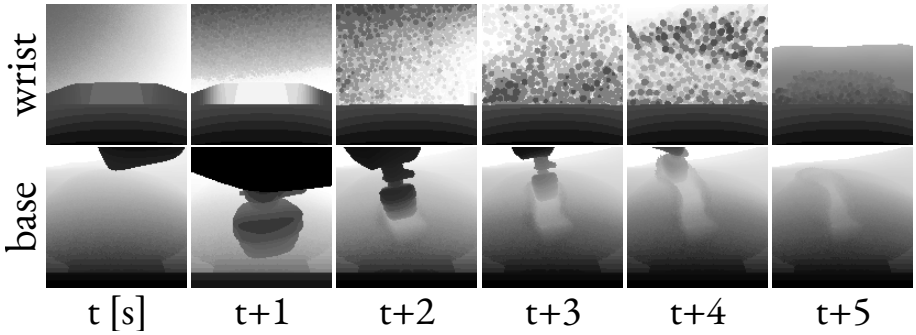


Figure 6.13 – Visual feedback provided to the `VISUAL` agent from two camera views over the course of a single excavation episode. The depth maps provide rich and real-time information about the robot workspace.

The results provide definitive evidence for the core hypotheses of this thesis. As shown in the learning curves in Figure 6.14, the `SPECIALIST` agent learned its task more quickly, as it only needed to master a single condition. However, this apparent efficiency was deceptive. When evaluated on a held-out test set of eight novel tool geometries, its performance collapsed, as shown in Table 6.4. It was unable to generalize its strategy to tools it had never seen, excavating almost no regolith. In stark contrast, the `ADAPTIVE` agent, trained with procedural tool diversity, maintained a high level of performance across all unseen tools. This outcome confirms that training with procedural diversity in embodiment is a critical factor in forging a truly robust and tool-aware policy.

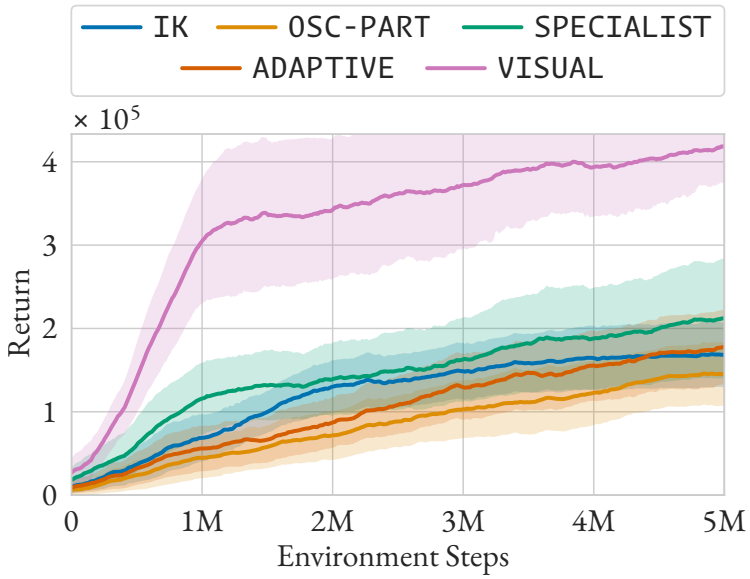


Figure 6.14 – Learning curves for the excavation task, where the `VISUAL` agent, trained with vision on diverse PCG tools, achieves the highest final performance.

Furthermore, the study validated the benefits of both learned compliance and visual perception. The agents with learned adaptive compliance learned significantly smoother and safer excavation strategies compared to a rigid IK baseline using PCG tools, with the `VISUAL` agent achieving an 80% reduction in motion jerk. The `VISUAL` agent also achieved the best overall performance, excavating nearly double the volume of regolith compared to its proprioception-only counterpart. This confirms that augmenting the agent with direct visual perception of the terrain and its own tool provides a significant advantage for planning and reactive control in such a dynamic task. This case study successfully demonstrates that the integrated framework can produce sophisticated, tool-aware behaviors, validating a critical capability for the future of autonomous construction and resource utilization in space.

Table 6.4 – Zero-shot generalization performance on eight novel tool geometries. The `SPECIALIST` agent fails to generalize. The `ADAPTIVE` agent generalizes well, and the `VISUAL` agent achieves the best performance across all metrics.

Agent	Excavated Volume (L)	Dust Generated	Motion Jerk
`IK` (baseline)	0.13	1.00	1.00
`SPECIALIST`	0.02	0.91	0.39
`ADAPTIVE`	0.14	0.94	0.44
`VISUAL`	**0.27**	**0.89**	**0.20**

6.2.3 Implications for Hardware Co-Design

This methodology presents a novel opportunity to inform hardware development. Because the learned generalist agent can operate a wide variety of tools, it can serve as a consistent evaluation metric for different mechanical designs. The performance of the `VISUAL` agent across the held-out set of novel tools, shown in Figure 6.15, reveals a significant variance in performance attributable to tool geometry alone. Some designs consistently enabled the agent to excavate more material with lower dust generation. For instance, tools with deeper cavities and more rounded designs generally performed better. This data-driven approach could enable a synergistic co-design process where hardware and control policies are developed in parallel to create tools that are inherently more compatible with autonomous systems.

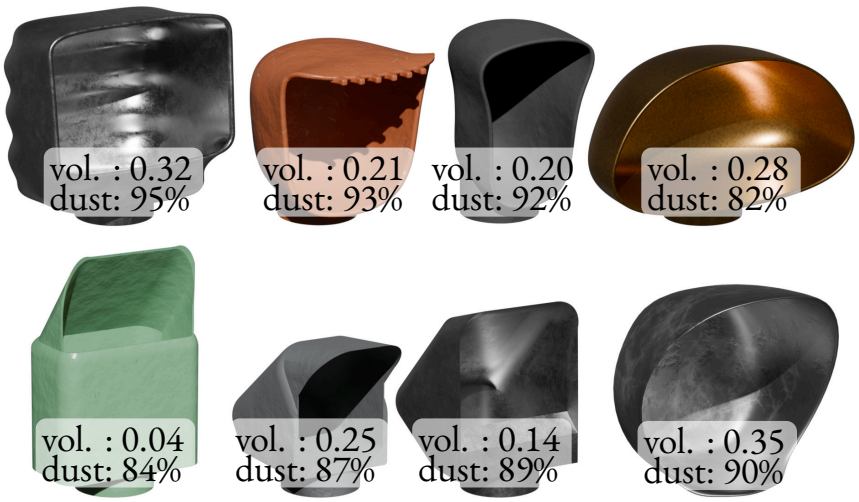


Figure 6.15 – The performance of the `VISUAL` agent across the held-out set of novel tool geometries. The results show significant performance variation based on tool design, highlighting the potential for data-driven hardware optimization.

6.3 Adversarial Air Hockey Diversity

To test the generality of the core principles beyond traditional space robotics tasks, this final case study examines their application in a highly dynamic, competitive, and contact-rich setting of the Robot Air Hockey Challenge 2023, detailed in **Publication X**. While seemingly unrelated to planetary exploration or orbital servicing, this task serves as a powerful abstract validation of the diversity-over-fidelity paradigm. In this context, diversity is not procedural or parametric, but strategic and adversarial. The intelligent, ever-changing behavior of an opponent becomes the source of unpredictable variation that the agent must learn to generalize against.

The challenge involved training a learning-based agent to control a 7-DoF KUKA iiwa14 manipulator to play air hockey against the agents of other teams, as shown in Figure 6.16. The task is characterized by fast-paced dynamics, the necessity for precise physical interaction with the puck, and the need to react to an intelligent opponent in real-time. Consistent with the blueprint developed in this thesis, the DreamerV3 algorithm was employed, training a policy from sparse rewards corresponding only to goals scored or faults conceded. The control interface also adhered to the principle of task-space control, where the agent learned to command an absolute target position for the mallet in the Cartesian plane, which was then mapped to joint commands via differential IK.

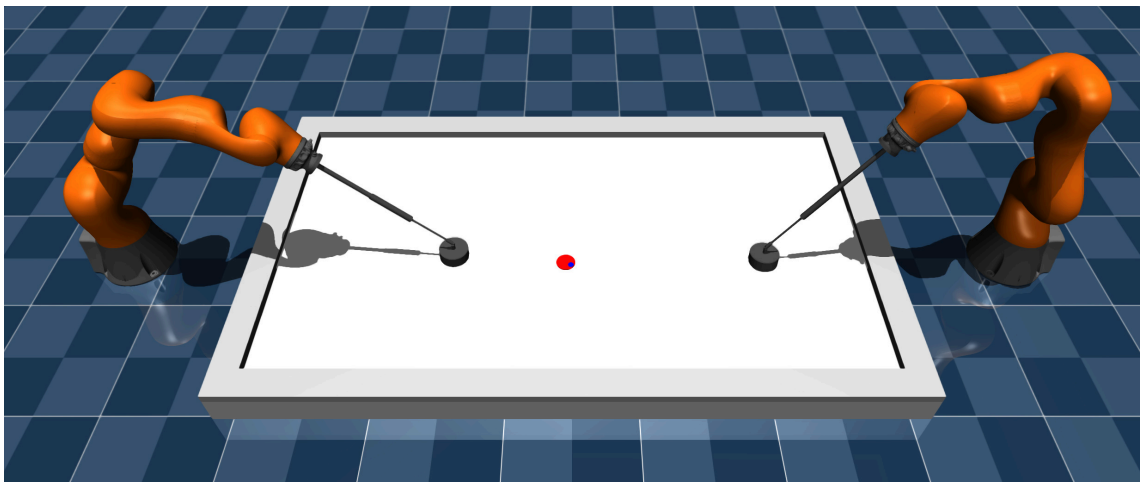


Figure 6.16 – MuJoCo [65] simulation environment of the Robot Air Hockey Challenge.

The core finding of this study was a direct and compelling parallel to the procedural diversity experiments. An agent trained solely against a single, static baseline opponent became highly specialized and overfit. This `static` agent appeared to master the game, achieving a near-perfect win rate against its predictable training partner. However, this mastery was an illusion. When faced with novel strategies from other competitors, the agent's performance was extremely brittle, and it failed catastrophically.

The solution was to introduce diversity through self-play, a form of adversarial training. The `self-play` agent was trained against an ever-evolving pool of its own past policies. This process was further enriched by pre-populating the opponent pool with specialist agents that had been trained with distinct reward functions to encourage aggressive, defensive, and balanced playstyles. This forced the agent to continuously discover and patch weaknesses in its own strategy, leading to a much more robust and generalizable policy.

The deceptive nature of the static training is made explicit by the quantitative results from head-to-head matches, summarized in Table 6.5. The `static` agent utterly dominated the `baseline` it was trained against with an average score of 14.5 to 0.1. However, when matched against the `self-play` agent, its strategy collapsed, and it was defeated just as decisively with a score of 0.7 to 14.3. This result is a powerful demonstration of overfitting. The `static` agent had not learned to play air hockey, but merely exploited the specific weaknesses of a single opponent.

Table 6.5 – Match results, where the `static` agent, trained only against the `baseline` opponent, appears dominant until it faces the novel strategies of the `self-play` agent.

Agent 1	Agent 2	Average Match Score
`static`	`baseline`	14.5 : 0.1
`self-play`	`baseline`	6.1 : 0.2
`static`	`self-play`	**0.7 : 14.3**

This process is a direct strategic analogue to training against procedurally generated terrains. In both cases, the agent is prevented from overfitting to a narrow, predictable set of conditions and is instead forced to learn the fundamental, generalizable principles of its task. The `self-play` agent ultimately secured second place in the tournament and was successfully transferred to the real-world robotic platform, losing only to a solution that leveraged optimal control and task priors. This success demonstrates the universality of the diversity-over-fidelity paradigm as a core principle for achieving robust and generalizable intelligence.

6.4 Discussion

The case studies in this chapter provide comprehensive validation for the core claims of this thesis, transitioning the work from methodological design to practical demonstration. The results form a cohesive picture of how the proposed framework achieves adaptive autonomy.

A foundational principle, confirmed across all validations, is the critical role of procedural diversity. From the successful zero-shot sim-to-real transfer of the traversal policy to the generalization across unseen tools in the excavation task and the robust performance against adversarial strategies in the air hockey challenge, the evidence is that an extensive experiential diversity is a fundamental prerequisite for learning generalizable policies.

While diversity enhances generalization, robust physical interaction demands a more sophisticated capability of a learned adaptive compliance. The success of the tool-aware excavation hinged on the agent's ability to gracefully manage unpredictable contact forces by dynamically modulating its own stiffness and damping, which is a skill a rigid controller lacks.

The success of these validations is ultimately a product of the synergistic combination. SRB provided the rich playground of experience, the MBRL paradigm supplied the efficient learning algorithm, and the learned compliant control methodology endowed the agent with physical intelligence. This integrated system, validated on real-world hardware, constitutes the practical realization of the blueprint for adaptive autonomy proposed by this thesis.

Finally, these validations also highlight the frontiers of current capabilities. The pronounced perceptual sim-to-real gap, identified in the traversal study, remains a significant hurdle. This finding suggests that even with a robustly transferred control policy, unmodeled sensor physics can be a primary point of failure, isolating robust perception as the next critical frontier for achieving true end-to-end autonomy. In conclusion, this chapter has grounded the contributions of this thesis in rigorous empirical evidence, providing tangible proof that the proposed framework is an effective and validated pathway toward building the autonomous systems required for the final frontier.

7

Conclusion

This thesis began with the grand challenge of creating robotic intelligence that can generalize to the profound uncertainty of extraterrestrial environments. It was guided by the hypothesis that true adaptive autonomy emerges not from the pursuit of perfect fidelity, but from the mastery of immense diversity. The research presented here systematically investigated this principle, delivering the Space Robotics Bench (SRB), an open-source framework for generating diverse experience at scale, and a learning blueprint that combines predictive world models with adaptive compliant control. The resulting body of evidence, culminating in successful real-world hardware deployments, confirms this core hypothesis and offers a new paradigm for developing robust autonomous systems for space.

This final chapter serves as a synthesis of that evidence. It begins by revisiting the research questions that structured the inquiry and providing the definitive answers this work has produced. It demonstrates how each of the initial research objectives was systematically fulfilled. The discussion then abstracts from the specific results to consider their broader implications for the scientific communities of machine learning and robotics, and for the engineering practice of space mission design. The thesis concludes with a critical reflection on its limitations and a forward-looking perspective on the open scientific questions that now lie on the path to a future of autonomy beyond Earth.

7.1 Answers to Research Questions

This research was structured as an inquiry into three fundamental questions. The work has produced distinct, evidence-backed answers to each.

Research Question 1

How can simulation environments be leveraged to effectively train robotic policies that generalize across the diverse and unpredictable conditions of space?

The answer provided by this thesis is that simulation must be leveraged not as an attempt to create a single, perfect digital twin, but as a factory for generating a vast and varied distribution of potential realities. The experiments consistently demonstrated that a policy's robustness is a direct product of the diversity of its training data. The combination of PCG for structural novelty and DR for parametric variation proved to be a powerful method for creating a training distribution rich enough to force a learning agent to discover invariant, generalizable strategies. The successful zero-shot sim-to-real transfer of the rover navigation policy in Chapter 6 serves as the primary evidence, where the agent trained with procedural diversity vastly outperformed its statically-trained counterpart in the physical world.

Research Question 2

What learning methodologies and control representations unlock the adaptive and compliant behaviors necessary for complex operations in space?

This work concludes that adaptive autonomy is unlocked by a methodology that endows an agent with both a predictive mind and a compliant body. The consistent superiority of the MBRL paradigm across all benchmark tasks points to the critical importance of an internal world model for efficient learning and effective decision-making under uncertainty. However, abstract prediction alone is insufficient for physical interaction. The integration of learned adaptive compliance via OSC provided the missing link, endowing the agent with physical intelligence. This allowed it to learn not just where to move but how to physically interact by dynamically modulating its own stiffness and damping in response to the world.

Research Question 3

What are the key failure modes and limitations of state-of-the-art robot learning algorithms when faced with the unstructured complexities of space-relevant scenarios?

Through systematic benchmarking with SRB, this research identified three primary limitations that define the current frontier of the field. First, there is a clear planning horizon limit. State-of-the-art algorithms consistently fail on tasks requiring long, precise action sequences, such

as the `solar_panel_assembly` task, suggesting that current methods for temporal credit assignment and deep exploration are insufficient. Second, a significant perceptual sim-to-real gap remains. As shown in Section 6.1, performance degrades substantially when learning from pixels due to unmodeled sensor noise, indicating that robust representation learning is a major, unsolved challenge. Third, the most pervasive failure mode is the brittleness of static training. This lack of generalization, confirmed in every case study, reinforces that policies that overfit to their training conditions are not merely suboptimal, but they are fundamentally unsuitable for the unpredictable nature of real-world robotics.

7.2 Summary of Fulfilled Objectives

The pursuit of answers to these questions was operationalized through four research objectives, all of which were successfully fulfilled.

Research Objective 1

Design and implement an open-source simulation framework for robot learning in space.

This objective was fulfilled by the creation and open-source release of SRB. As detailed in Chapter 4, this platform provides the critical infrastructure for this research, integrating a high-performance backend with a powerful procedural engine and accessible robotics and learning interfaces.

Research Objective 2

Establish a standardized suite of benchmark tasks within the developed simulation framework to rigorously evaluate the generalization and adaptation capabilities of robot learning algorithms in space-relevant scenarios.

This objective was realized by populating SRB with a comprehensive suite of mission-relevant benchmark tasks. The tasks, presented in Chapter 4, provide the standardized, challenging testbed that enabled the rigorous empirical evaluations forming the core evidence of this thesis.

Research Objective 3

Investigate and develop a learning methodology for robust adaptive control.

This objective was fulfilled by the development of the learning blueprint in Chapter 5. This work systematically established the superiority of a model-based paradigm and introduced a novel methodology for learning adaptive compliance through the integration of MBRL with OSC.

Research Objective 4

Validate the proposed framework and methodology via sim-to-real transfer in a terrestrial analogue facility.

This final and most critical objective was met through multiple, successful zero-shot sim-to-real experiments. The foundational grasping study in Chapter 3 and the comprehensive validation of adaptive traversal in Chapter 6 provide definitive physical proof that the framework and methodologies developed in this thesis can successfully bridge the sim-to-real gap.

7.3 Broader Implications and Impact

The conclusions of this research have implications that extend beyond the specific domain of space robotics, contributing new tools, methodologies, and insights to the broader scientific and engineering communities.

7.3.1 Machine Learning

For the machine learning community, this work serves as a large-scale case study in embodied intelligence, providing strong empirical evidence that generalization in physical systems is deeply tied to the diversity of the learning curriculum. SRB itself is a significant contribution, offering a new, challenging, open-source testbed for research in RL, representation learning, and long-horizon planning that moves beyond the well-explored domain of tabletop manipulation. The success of the learned compliance framework reinforces a key idea in embodied AI where true intelligence is not just about abstract reasoning but also about skillful and adaptive physical interaction with the world.

7.3.2 Terrestrial Robotics

The methodologies forged for the extreme unstructured environments of space are directly applicable to the most challenging domains on Earth. The blueprint of combining procedural simulation with learned compliant control can enhance the robustness of robots in fields such as agriculture, construction, disaster response, and logistics, where robots must operate in similarly unpredictable and dynamic conditions. The open-source framework can be adapted to create new benchmarks for these domains, fostering a similar data-driven approach to solving their respective contact-rich challenges.

7.3.3 Space Mission Design and Operations

For the space sector, this research offers a new paradigm for developing and validating autonomous systems. The traditional and deterministic verification process can be augmented with the methodology presented here. The ability to probabilistically validate control software against thousands of procedurally generated scenarios provides a pathway to building statistical assurance in learning-based systems. This can increase the technological readiness of autonomy more efficiently, enable more ambitious mission concepts, facilitate the data-driven co-design of robotic hardware and software, and ultimately accelerate the deployment of the persistent robotic infrastructure needed for a sustainable human presence beyond Earth.

7.4 Limitations

A rigorous scientific inquiry requires a clear acknowledgment of its boundaries. While this thesis provides a validated pathway toward adaptive autonomy, the findings are subject to several important limitations that define the frontier of current research.

7.4.1 The Inescapable Reality of the Sim-to-Real Gap

The entire methodology is simulation-centric. While successful zero-shot transfer was demonstrated, this success was achieved in controlled laboratory analogues. A residual gap to the true physics of extraterrestrial environments undoubtedly remains, with phenomena like extreme thermal cycles, vacuum effects, and complex regolith chemistry. The ultimate performance of these policies in space is therefore an extrapolation, contingent on the core assumption that the procedural distribution of simulated realities is broad enough to contain the single reality of the target environment.

7.4.2 Constraints of Space-Grade Hardware

This research focused on developing generalizable software and, as such, did not fully model the severe constraints of flight-certified hardware. The performance of a policy on a real space robot will be affected by factors like the limited computational power of radiation-hardened processors, different sensor noise profiles, and unique actuator dynamics that were only approximated through randomization. The inference latency of the preferred model-based agent, while manageable on modern hardware, presents a significant challenge for deployment on current space-grade computers.

7.4.3 The Grand Challenge of Verification and Validation

The most significant limitation, shared by the entire field of learning-based control, is the lack of formal safety guarantees. The policies are products of a stochastic optimization process and are not amenable to traditional verification techniques that rely on deterministic models. While their robustness is demonstrated empirically across thousands of test cases, this does not constitute a mathematical proof of safe behavior under all possible conditions. This verifiability gap remains the primary obstacle to the deployment of any learned policy in a mission-critical context where failure is not an option.

7.5 Future Work

This thesis provides a solid foundation and a clear direction for future scientific inquiry. The identified limitations motivate several critical and exciting avenues of research that can build directly upon the contributions of this work.

The most immediate priority is to continue closing the perceptual sim-to-real gap. Future work should focus on developing higher-fidelity sensor models within SRB and on training perception systems that are robust to real-world noise, potentially using techniques that learn to adapt to the sensor domain online.

Second, the long-horizon planning problem remains a major hurdle. The solution will likely require moving towards new algorithmic designs that dynamically create their own abstractions and automatically pursue short-horizon goals that break down complex tasks into manageable segments. This will involve integrating the low-level adaptive motor skills developed here with high-level symbolic planners. The data collection capabilities of SRB make it an ideal platform for exploring how these abstraction hierarchies can be bootstrapped with LfD.

A third, transformative avenue is the integration of multi-modal foundation models [100]. Combining the robust control policies from this work with modern vision-language models could enable a new level of semantic understanding and human-robot interaction, allowing astronauts to command robots with natural language.

Expanding the framework to handle multi-agent coordination is another promising direction. The architecture of SRB is well-suited for research into collaborative robotics, a critical capability for future large-scale construction and exploration missions where teams of robots will need to work together.

Finally, the grand challenge of safety verification must be confronted. Future work must focus on developing methods to make learned policies more interpretable, to formally bound their behavior in uncertain states, and to integrate them with classical safety supervisors that can act as a fail-safe. Solving this problem is the critical final step in transforming adaptive autonomy from a powerful research concept into a trusted and indispensable tool for the future of human presence beyond Earth.

Bibliography

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

[2] V. Verma *et al.*, “Enabling Long & Precise Drives for The Perseverance Mars Rover via Onboard Global Localization,” in *IEEE Aerospace Conference*, 2024, pp. 1–18. doi: 10.1109/AERO58975.2024.10521160.

[3] M. Rognant, C. Cumer, J.-M. Biannic, M. A. Roa, A. Verhaeghe, and V. Bissonnette, “Autonomous Assembly of Large Structures in Space: A Technology Review,” in *European Conference for Aeronautics and Aerospace Sciences*, Madrid, Spain, July 2019.

[4] Z. Xue, J. Liu, C. Wu, and Y. Tong, “Review of In-Space Assembly Technologies,” *Chinese Journal of Aeronautics*, vol. 34, no. 11, pp. 21–47, 2021, doi: 10.1016/j.cja.2020.09.043.

[5] R. Doyle *et al.*, “Recent Research and Development Activities on Space Robotics and AI,” *Advanced Robotics*, vol. 35, no. 21–22, pp. 1244–1264, Nov. 2021.

[6] T. Zhang *et al.*, “The Progress of Extraterrestrial Regolith-Sampling Robots,” *Nature Astronomy*, vol. 3, pp. 487–497, June 2019, doi: 10.1038/s41550-019-0804-1.

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

[8] GITAI, “GITAI Develops Lunar Robotic Rover R1 and Conducts Successful Demonstration at JAXA's Mock Lunar Surface Environment.” [Online]. Available: https://gitai.tech/2022/02/10/gitai-develops-lunar-robotic-rover-r1

[9] GITAI, “GITAI Successfully Demonstrates Lunar Manipulator and Rover in Simulated Regolith Chamber.” [Online]. Available: https://gitai.tech/2023/06/06/gitai-successfully-demonstrates-lunar-manipulator-and-rover-in-simulated-regolith-chamber

[10] GITAI, “Space Robotics Start-up GITAI Completes Successful Technology Demonstration Inside the ISS.” [Online]. Available: https://gitai.tech/2021/10/28/iss-tech-demo-ja

[11] GITAI, "GITAI Completes Fully Successful Technology Demonstration Outside the ISS." [Online]. Available: https://gitai.tech/2024/03/19/gitai-completes-fully-successful-technology-demonstration-outside-the-iss

[12] R. W. Moses and D. M. Bushnell, *Frontier In-Situ Resource Utilization for Enabling Sustained Human Presence on Mars*. National Aeronautics and Space Administration, 2016.

[13] A. Rankin, N. Patel, E. Graser, J.-K. F. Wang, and K. Rink, "Assessing Mars Curiosity Rover Wheel Damage," in *IEEE Aerospace Conference*, 2022, pp. 1–19.

[14] 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. 1395–1476, Jan. 2021.

[15] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: A Bradford Book, 2018.

[16] D. Silver *et al.*, "Mastering the game of Go without human knowledge," *Nature*, vol. 550, no. 7676, pp. 354–359, Oct. 2017, doi: 10.1038/nature24270.

[17] O. Vinyals *et al.*, "Grandmaster level in StarCraft II using multi-agent reinforcement learning," *Nature*, vol. 575, no. 7782, pp. 350–354, Nov. 2019, doi: 10.1038/s41586-019-1724-z.

[18] S. P. Hughes, R. H. Qureshi, S. D. Cooley, and J. J. Parker, "Verification and Validation of the General Mission Analysis Tool (GMAT)," in *AIAA/AAS Astrodynamics Specialist Conference*, 2014.

[19] P. W. Kenneally, S. Piggott, and H. Schaub, "Basilisk: A Flexible, Scalable and Modular Astrodynamics Simulation Framework," *Journal of Aerospace Information Systems*, vol. 17, no. 9, pp. 496–507, 2020.

[20] P. Henderson, R. Islam, P. Bachman, J. Pineau, D. Precup, and D. Meger, "Deep Reinforcement Learning that Matters," in *Association for the Advancement of Artificial Intelligence*, 2018.

[21] K. Bousmalis *et al.*, "Using Simulation and Domain Adaptation to Improve Efficiency of Deep Robotic Grasping," in *International Conference on Robotics and Automation*, Brisbane, Australia, 2018, pp. 4243–4250. doi: 10.1109/ICRA.2018.8460875.

[22] A. B. Mortensen and S. Bøgh, "RLRoverLAB: An Advanced Reinforcement Learning Suite for Planetary Rover Simulation and Training," in *International Conference on Space Robotics*, 2024, pp. 273–277. doi: 10.1109/iSpaRo60631.2024.10687686.

[23] M. El-Hariry, A. Richard, and M. Olivares-Mendez, "RANS: Highly-Parallelised Simulator for Reinforcement Learning based Autonomous Navigating Spacecrafts," *arXiv:2310.07393*, 2023, doi: 10.48550/arXiv.2310.07393.

[24] 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, doi: 10.1109/LRA.2020.2974707.

[25] T. Yu *et al.*, "Meta-World: A benchmark and evaluation for multi-task and meta reinforcement learning," in *Conference on Robot Learning*, 2020, pp. 1094–1100.

[26] 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.

[27] S. Tao *et al.*, "ManiSkill3: GPU Parallelized Robotics Simulation and Rendering for Generalizable Embodied AI," *arXiv:2410.00425*, 2024, doi: 10.48550/arXiv.2410.00425.

[28] K. Cobbe, C. Hesse, J. Hilton, and J. Schulman, "Leveraging Procedural Generation to Benchmark Reinforcement Learning," in *International Conference on Machine Learning*, 2020, pp. 2048–2056.

[29] 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 *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2017, pp. 23–30.

[30] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal Policy Optimization Algorithms," *arXiv:1707.06347*, 2017, doi: 10.48550/arXiv.1707.06347.

[31] S. Fujimoto, H. Hoof, and D. Meger, "Addressing Function Approximation Error in Actor-Critic Methods," in *International Conference on Machine Learning*, 2018, pp. 1587–1596.

[32] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor," in *International Conference on Machine Learning*, 2018.

[33] D. Hafner, J. Pasukonis, J. Ba, and T. Lillicrap, "Mastering Diverse Control Tasks through World Models," *Nature*, vol. 640, pp. 647–653, 2025.

[34] O. Khatib, "A Unified Approach for Motion and Force Control of Robot Manipulators: The Operational Space Formulation," *IEEE Journal on Robotics and Automation*, vol. 3, no. 1, pp. 43–53, 1987.

[35] M. Towers *et al.*, "Gymnasium: A Standard Interface for Reinforcement Learning Environments," *arXiv:2407.17032*, 2024, doi: 10.48550/arXiv.2407.17032.

[36] S. Macenski, T. Foote, B. Gerkey, C. Lalancette, and W. Woodall, "Robot Operating System 2: Design, architecture, and uses in the wild," *Science Robotics*, vol. 7, no. 66, May 2022, doi: 10.1126/scirobotics.abm6074.

[37] P. Ludivig, A. Calzada-Diaz, M. Olivares-Mendez, H. Voos, and J. Lamamy, "Building a Piece of the Moon: Construction of Two Indoor Lunar Analogue Environments," in *International Astronautical Congress*, 2020.

[38] Open X-Embodiment Collaboration, "Open X-Embodiment: Robotic Learning Datasets and RT-X Models," *arXiv:2310.08864*, 2023, doi: 10.48550/arXiv.2310.08864.

[39] V. Verma, A. Jonsson, R. Simmons, T. Estlin, and R. Levinson, "Survey of Command Execution Systems - for NASA Spacecraft and Robots," p. 8, 2005.

[40] G. A. Soffen, "The Viking Project," *Journal of Geophysical Research*, vol. 82, no. 28, pp. 3959–3970, 1977.

[41] S. M. Stevenson, "Mars Pathfinder Rover-Lewis Research Center Technology Experiments Program," in *IECEC-97 Proceedings of the Thirty-Second Intersociety Energy Conversion Engineering Conference (Cat. No. 97CH6203)*, 1997, pp. 722–727.

[42] National Aeronautics and Space Administration, "AS12-47-6932: Close-up view of a set of tongs, an Apollo Lunar Hand Tool, being used by Astronaut Charles Conrad Jr., to pick up lunar samples during the Apollo XII mission." [Online]. Available: https://www.archives.gov/presidential-libraries/events/centennials/nixon/exhibit/nixon-online-exhibit-samples.html

[43] J. P. Grotzinger *et al.*, "Mars Science Laboratory Mission and Science Investigation," *Space Science Reviews*, vol. 170, pp. 5–56, July 2012, doi: 10.1007/s11214-012-9892-2.

[44] L. Dilillo, A. Bosser, A. Javanainen, and A. Virtanen, "Space Radiation Effects in Electronics," *Rad-hard Semiconductor Memories*. pp. 1–64, 2022.

[45] National Aeronautics and Space Administration, "Detailed Panorama of Mars' Jezero Crater Delta." [Online]. Available: https://www.jpl.nasa.gov/images/pia24921-detailed-panorama-of-mars-jezero-crater-delta

[46] D. St-Onge *et al.*, "Planetary Exploration With Robot Teams: Implementing Higher Autonomy With Swarm Intelligence," *IEEE Robotics Automation Magazine*, vol. 27, no. 2, pp. 159–168, June 2020, doi: 10.1109/MRA.2019.2940413.

[47] M. J. Schuster *et al.*, "The ARCHES Space-Analogue Demonstration Mission: Towards Heterogeneous Teams of Autonomous Robots for Collaborative Scientific Sampling in Planetary Exploration," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 5315–5322, Oct. 2020, doi: 10.1109/LRA.2020.3007468.

[48] F. Zhou *et al.*, "Simulations of Mars Rover Traverses," *Journal of Field Robotics*, vol. 31, no. 1, pp. 141–160, 2014, doi: 10.1002/rob.21483.

[49] Z. Gu *et al.*, "Humanoid Locomotion and Manipulation: Current Progress and Challenges in Control, Planning, and Learning," *arXiv:2501.02116*, 2025, doi: 10.48550/arXiv.2501.02116.

[50] I. F. Jasim, P. W. Plapper, and H. Voos, "Contact-State Modelling in Force-Controlled Robotic Peg-in-Hole Assembly Processes of Flexible Objects Using Optimised Gaussian Mixtures," *Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, vol. 231, no. 8, 2017.

[51] H. Lee, S. Park, K. Jang, S. Kim, and J. Park, "Contact State Estimation for Peg-in-Hole Assembly Using Gaussian Mixture Model," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, 2022.

[52] J. Xu, Z. Hou, Z. Liu, and H. Qiao, "Compare Contact Model-based Control and Contact Model-free Learning: A Survey of Robotic Peg-in-hole Assembly Strategies," *arXiv:1904.05240*, 2019, doi: 10.48550/arXiv.1904.05240.

[53] P. Beeson and B. Ames, "TRAC-IK: An Open-Source Library for Improved Solving of Generic Inverse Kinematics," in *IEEE-RAS International Conference on Humanoid Robots*, Nov. 2015, pp. 928–935. doi: 10.1109/HUMANOIDS.2015.7363472.

[54] J. Peters and S. Schaal, "Learning operational space control," *Robotics: Science and Systems*, 2006, doi: 10.15607/RSS.2006.II.033.

[55] J. Wong, V. Makoviychuk, A. Anandkumar, and Y. Zhu, "OSCAR: Data-Driven Operational Space Control for Adaptive and Robust Robot Manipulation," in *International Conference on Robotics and Automation*, 2022, pp. 10519–10526. doi: 10.1109/ICRA46639.2022.9811967.

[56] A. S. Polydoros and L. Nalpantidis, "Survey of Model-Based Reinforcement Learning: Applications on Robotics," *Journal of Intelligent & Robotic Systems*, vol. 86, no. 2, pp. 153–173, May 2017, doi: 10.1007/s10846-017-0468-y.

[57] M. P. Deisenroth and C. E. Rasmussen, "PILCO: A model-based and data-efficient approach to policy search," in *28th International Conference on Machine Learning*, in ICML'11. Madison, WI, USA: Omnipress, June 2011, pp. 465–472.

[58] T. Osa, J. Pajarinen, G. Neumann, J. Bagnell, P. Abbeel, and J. Peters, "An Algorithmic Perspective on Imitation Learning," *Foundations and Trends in Robotics*, vol. 7, pp. 1–179, Nov. 2018, doi: 10.1561/2300000053.

[59] T. Zhang *et al.*, "Deep Imitation Learning for Complex Manipulation Tasks from Virtual Reality Teleoperation," in *IEEE International Conference on Robotics and Automation*, May 2018, pp. 5628–5635. doi: 10.1109/ICRA.2018.8461249.

[60] N. J. Cho, S. H. Lee, J. B. Kim, and I. H. Suh, "Learning, Improving, and Generalizing Motor Skills for the Peg-in-Hole Tasks Based on Imitation Learning and Self-Learning," *Applied Sciences*, vol. 10, no. 8, 2020.

[61] K. Wang, Y. Zhao, and I. Sakuma, "Learning Robotic Insertion Tasks From Human Demonstration," *IEEE Robotics and Automation Letters*, 2023.

[62] A. Orsula, "Deep Reinforcement Learning for Robotic Grasping from Octrees," Master's Thesis, 2021.

[63] B. Mehta, M. Diaz, F. Golemo, C. J. Pal, and L. Paull, "Active Domain Randomization," in *Conference on Robot Learning*, Oct. 2020, pp. 1162–1176.

[64] N. Koenig and A. Howard, "Design and Use Paradigms for Gazebo, an Open-Source Multi-Robot Simulator," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sept. 2004, pp. 2149–2154vol.3. doi: 10.1109/IROS.2004.1389727.

[65] E. Todorov, T. Erez, and Y. Tassa, "MuJoCo: A physics engine for model-based control," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct. 2012, pp. 5026–5033. doi: 10.1109/IROS.2012.6386109.

[66] J. Lee *et al.*, "DART: Dynamic Animation and Robotics Toolkit," *The Journal of Open Source Software*, vol. 3, p. 500, Feb. 2018, doi: 10.21105/joss.00500.

[67] S. Stukes *et al.*, "An Innovative Approach to Modeling VIPER Rover Software Life Cycle Cost," in *IEEE Aerospace Conference*, 2021, pp. 1–16. doi: 10.1109/AERO50100.2021.9438347.

[68] B. Coltin *et al.*, "Astrobee Robot Software: A Modern Software System for Space," in *IEEE International Conference on Robotics and Automation*, 2019, pp. 5556–5562.

[69] S. Wang, Y. Cao, X. Zheng, and T. Zhang, "Collision-Free Trajectory Planning for a 6-DoF Free-Floating Space Robot via Hierarchical Decoupling Optimization," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 4953–4960, 2022.

[70] M. A. Stephenson and H. Schaub, "BSK-RL: Modular, High-Fidelity Reinforcement Learning Environments for Spacecraft Tasking," in *International Astronautical Congress*, 2024.

[71] D. Hirano, S. Mitani, K. Watanabe, T. Nishishita, T. Yamamoto, and S. P. Yamaguchi, "Int-Ball2: On-Orbit Demonstration of Autonomous Intravehicular Flight and Docking for Image Capturing and Recharging," in *IEEE International Conference on Robotics and Automation*, 2025.

[72] C. Leake, H. Grip, V. Steyert, T. D. Hasseler, M. Cacan, and A. Jain, "HeliCAT-DARTS: A High Fidelity, Closed-Loop Rotorcraft Simulator for Planetary Exploration," *Aerospace*, vol. 11, no. 9, 2024, doi: 10.3390/aerospace11090727.

[73] T. D. Hasseler *et al.*, "EELS-DARTS: A Planetary Snake Robot Simulator for Closed-Loop Autonomy Development," *Aerospace*, vol. 11, no. 10, 2024, doi: 10.3390/aerospace11100795.

[74] A. Richard *et al.*, "OmniLRS: A Photorealistic Simulator for Lunar Robotics," in *International Conference on Robotics and Automation*, 2024, pp. 16901–16907. doi: 10.1109/ICRA57147.2024.10610026.

[75] S. Narvekar, B. Peng, M. Leonetti, J. Sinapov, M. Taylor, and P. Stone, *Curriculum Learning for Reinforcement Learning Domains: A Framework and Survey*. 2020.

[76] Y. Zhu *et al.*, "robosuite: A Modular Simulation Framework and Benchmark for Robot Learning," in *arXiv:2009.12293*, 2020. doi: 10.48550/arXiv.2009.12293.

[77] V. Kumar *et al.*, "RoboHive: A Unified Framework for Robot Learning," *Advances in Neural Information Processing Systems*, vol. 36, pp. 44323–44340, 2023.

[78] K. Zakka *et al.*, "RoboPianist: Dexterous Piano Playing with Deep Reinforcement Learning," in *Conference on Robot Learning*, in Proceedings of Machine Learning Research, vol. 229. 2023, pp. 2975–2994.

[79] C. Sferrazza, D.-M. Huang, X. Lin, Y. Lee, and P. Abbeel, "HumanoidBench: Simulated Humanoid Benchmark for Whole-Body Locomotion and Manipulation," *arXiv:2403.10506*, 2024, doi: 10.48550/arXiv.2403.10506.

[80] P.-S. Wang, Y. Liu, Y.-X. Guo, C. Sun, and X. Tong, "O-CNN: Octree-based Convolutional Neural Networks for 3D Shape Analysis," *ACM Transactions on Graphics*, vol. 36, no. 72, pp. 1–11, Aug. 2017, doi: 10.1145/3072959.3073608.

[81] P.-S. Wang, Y. Liu, and X. Tong, *Deep Octree-based CNNs with Output-Guided Skip Connections for 3D Shape and Scene Completion*. 2020.

[82] G. Riegler, A. O. Ulusoy, and A. Geiger, "OctNet: Learning Deep 3D Representations at High Resolutions," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6620–6629, July 2017.

[83] G. Brockman *et al.*, "OpenAI Gym," *arXiv:1606.01540*, 2016, doi: 10.48550/arXiv.1606.01540.

[84] D. Ferigo, S. Traversaro, G. Metta, and D. Pucci, "Gym-Ignition: Reproducible Robotic Simulations for Reinforcement Learning," in *IEEE/SICE International Symposium on System Integration*, Jan. 2020, pp. 885–890. doi: 10.1109/SII46433.2020.9025951.

[85] S. Levine, C. Finn, T. Darrell, and P. Abbeel, "End-to-End Training of Deep Visuomotor Policies," *Journal of Machine Learning Research*, vol. 17, no. 39, pp. 1–40, Jan. 2016.

[86] M. Gualtieri, A. t. Pas, and R. Platt, "Pick and Place Without Geometric Object Models," in *IEEE International Conference on Robotics and Automation*, May 2018, pp. 7433–7440. doi: 10.1109/ICRA.2018.8460553.

[87] E. Ahmed *et al.*, *Deep Learning Advances on Different 3D Data Representations: A Survey*. 2018.

[88] 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.

[89] A. Kuznetsov, P. Shvechikov, A. Grishin, and D. Vetrov, "Controlling Overestimation Bias with Truncated Mixture of Continuous Distributional Quantile Critics," in *International Conference on Machine Learning*, July 2020, pp. 5556–5566.

[90] J. Kuffner and S. M. LaValle, "RRT-Connect: An Efficient Approach to Single-Query Path Planning," in *IEEE International Conference on Robotics and Automation*, Apr. 2000, pp. 995–1001. doi: 10.1109/ROBOT.2000.844730.

[91] M. Macklin *et al.*, "Small steps in physics simulation," in *Proceedings of the 18th Annual ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, in SCA '19. Los Angeles, California: Association for Computing Machinery, 2019. doi: 10.1145/3309486.3340247.

[92] PyO3 Project and Contributors, "PyO3: Rust bindings for the Python interpreter." [Online]. Available: https://github.com/PyO3/pyo3

[93] J. Ansel *et al.*, "PyTorch 2: Faster Machine Learning Through Dynamic Python Bytecode Transformation and Graph Compilation," in *29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2*, Apr. 2024. doi: 10.1145/3620665.3640366.

[94] NVIDIA Corporation, "NVIDIA Omniverse." [Online]. Available: https://nvidia.com/omniverse

[95] Blender Online Community, “Blender - A 3D Modelling and Rendering Package.” [Online]. Available: http://blender.org/

[96] 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.

[97] A. Serrano-Muñoz, D. Chrysostomou, S. Bøgh, and N. Arana-Arexolaleiba, “skrl: Modular and Flexible Library for Reinforcement Learning,” *Journal of Machine Learning Research*, vol. 24, no. 1, 2023.

[98] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” *Neural Computation*, vol. 9, no. 8, 1997.

[99] M. A. Felix, W. S. Slater, D. C. Landauer, R. E. Pinson, and B. B. Rutherford, “Total Ionizing Dose Radiation Testing of NVIDIA Jetson Orin NX System on Module,” in *IEEE Space Computing Conference*, 2024, pp. 116–121. doi: 10.1109/SCC61854.2024.00019.

[100] A. Brohan *et al.*, “RT-2: Vision-Language-Action Models Transfer Web Knowledge to Robotic Control,” in *arXiv:2307.15818*, 2023. doi: 10.48550/arXiv.2307.15818.


SnT