Procedural Digital Cousins for Scalable Robot Learning in Space

Andrej Orsula[1], Ricard M. Castan[1], Miguel Olivares-Mendez[1], Carol Martinez[1]

Abstract— **Robots play a transformative role in the future of space exploration, from fleets of autonomous rovers navigating unstructured planetary surfaces to robotic manipulators constructing orbital megastructures. However, the adoption of data-intensive robot learning methods in this domain is severely hindered by extreme data scarcity and the practical infeasibility of physical training upon deployment. While high-fidelity digital twins are highly effective for terrestrial workflows, constructing an exact virtual replica is often infeasible for the unpredictable off-world environments. To address this limitation, we introduce the Space Robotics Bench, an open-source framework that leverages procedural content generation and domain randomization to synthesize a vast distribution of physically plausible scenarios. To ensure reliable sim-to-real transfer, the foundational parameters of these generative environments are anchored to physical analogue facilities via the incorporation of digital cousins across both planetary and orbital operations. We establish baselines using reinforcement learning algorithms and demonstrate the efficacy of the framework by successfully achieving zero-shot sim-to-real transfer of a learned navigation policy. Our results show that agents trained across diverse procedural simulations achieve robustness, which highlights the generative approach as a highly scalable pipeline for providing the statistical assurance required to deploy autonomous systems beyond Earth. *The source code is available at github.com/AndrejOrsula/space_robotics_bench.***

I. Introduction

From fleets of autonomous rovers navigating unstructured planetary surfaces to complex robotic manipulators capturing orbital debris and constructing massive structures in deep space [1], [2], the success of long-duration endeavors depends heavily on developing robotic systems capable of unassisted operation. As deep space missions face severe communication latency that often spans several minutes to hours, direct human teleoperation is generally infeasible. Consequently, robots must exhibit a high degree of onboard autonomy. Robot learning, particularly reinforcement learning (RL) [3], offers a powerful paradigm for acquiring such adaptive closed-loop behaviors. However, the adoption of data-intensive learning methods in space robotics is severely hindered by extreme data scarcity and the practical impossibility of conducting in situ physical trial-and-error training.

A fundamental challenge for any simulation-based approach attempting to solve this issue is bridging the sim-to-real gap. The conventional strategy in structured terrestrial robotics is to construct a singular digital twin using multimodal sensor sweeps. While highly effective for well-defined factory floors or mapped indoor environments, this methodology acts merely as a digital replica. It is inherently brittle and often impossible to construct, for unpredictable extraterrestrial conditions where prior mapping is either heavily degraded or unavailable. A policy trained in a singular, static environment is highly prone to overfitting to unverified, localized assumptions about the world. This narrow optimization results in catastrophic failure when the agent is inevitably deployed and faces real-world geometric variations such as unexpected rock distributions, dynamically varying regolith cohesion, or harsh orbital lighting conditions.

We advocate an alternative approach that uses generative digital cousins, which are highly procedural evolutions of traditional digital twins. If a digital twin is an exact replica of a specific environment, a digital cousin is a synthetic 3D environment that purposefully avoids replicating real-

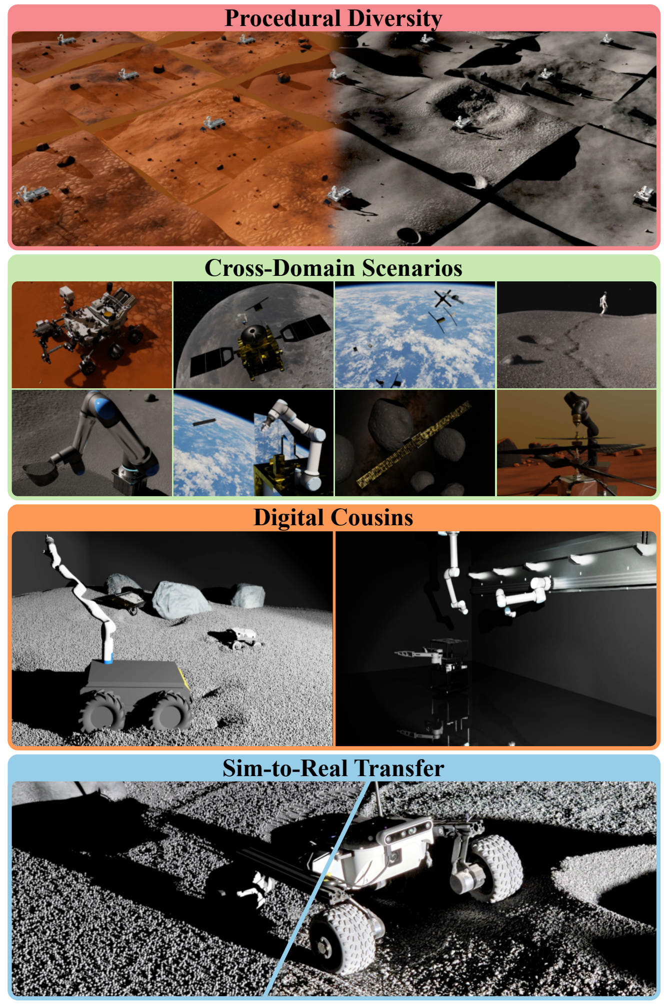


Fig. 1: The generative workflow of the Space Robotics Bench towards robust robot learning beyond Earth.

[1]University of Luxembourg

world peculiarities. Instead, it functionally and statistically shares the physical characteristics while exhibiting a vastly different structural topology via novel layouts and randomized parameters. By exposing an RL agent to such a vast and continuously shifting distribution of these procedural cousins synthesized via procedural content generation and comprehensive domain randomization, the agent is forced to acquire a deeply generalizable understanding of underlying physical interactions, terramechanics, and spatial constraints. This paradigm circumvents the pitfalls of spatial memorization.

To realize this conceptual shift at scale, we introduce the Space Robotics Bench (SRB), a comprehensive, open-source simulation framework built natively on NVIDIA Isaac Lab [4]. This work formalizes the generative digital cousin workflow by detailing a procedural generation engine that automates the parametric synthesis and programmatic export of infinite space-relevant environment variations for scalable training. To ensure these generative models remain physically plausible, we present a real-to-sim pipeline that anchors their foundational parameters to terrestrial analogue facilities. This approach dynamically bounds the physical distributions for both planetary surface operations, modeled after a lunar analogue facility, and orbital robotics, modeled after a microgravity laboratory. Finally, we establish baselines using standard reinforcement learning algorithms and demonstrate the efficacy of the framework through zero-shot sim-to-real transfer of learned policies entirely optimized within these procedural distributions. Our empirical validations confirm that this generative approach produces robust autonomous behaviors across challenging conditions. Together, these contributions establish a highly scalable pipeline for providing the statistical assurance required to deploy reliable autonomous systems in the unpredictable environments of space.

II. Related Work

This work is positioned at the intersection of generative simulation, cross-domain robot learning, and scalable sim-to-real pipelines that are tailored for the challenging extraterrestrial environments.

A. Generative Digital Cousins

Recent advances heavily leverage digital twins for real-to-sim and sim-to-real pipelines. Traditional methods rely on capturing and reconstructing exact 3D replicas from physical data. Techniques such as photogrammetry or Neural Radiance Fields are frequently employed to achieve photorealistic rendering and precise geometric bounds [5], [6]. However, these methods produce static representations that completely fail to encapsulate the immense variance of unknown environments. For unstructured extraterrestrial deployment, a single digital twin presents a critical point of failure. Consequently, the research community is increasingly shifting toward generative digital twins capable of synthesizing broad distributions of training data [7]. Our work pushes this concept to its logical extreme via generative digital cousins. By abandoning the goal of absolute geometric parity, we leverage procedural simulation to heavily randomize the underlying physical dynamics, including non-linear granular media sinkage and six-degree-of-freedom microgravity inertia, alongside completely novel visual geometries. These cousins act as a robust, expansive proxy distribution covering nearly all potential states an agent might encounter upon physical deployment.

B. Hybrid Physics-Based and Generative Models

Bridging the sim-to-real gap traditionally relies on rigorous system identification, which is difficult to perform accurately when the final deployment target is another celestial body. Domain randomization (DR) [8] forces policy robustness by perturbing physical and visual parameters to implicitly regularize neural networks against exploiting simulator-specific flaws. We advance this foundational strategy by coupling it tightly with procedural content generation (PCG). While standard DR generally alters only the parameters of pre-existing static assets, our hybrid generative model algorithmically synthesizes an extensive distribution of topologically unique entities [9]. In this framework, we treat the sim-to-real gap not as an error metric to be minimized against a single digital twin, but as an expression of epistemic uncertainty represented by a broad probabilistic distribution of digital cousins that the agent must seamlessly master.

C. Scalable Simulation Frameworks

The field of robot learning has advanced significantly through the adoption of standardized, GPU-accelerated simulators [10]. Frameworks built on NVIDIA Isaac Sim and Isaac Lab enable thousands of parallel environments to run simultaneously on a single workstation by keeping both physics and rendering entirely within VRAM [4]. While existing benchmarks predominantly focus on highly structured, Earth-centric scenarios, such as quadruped locomotion and predictable tabletop manipulation, our framework exploits scalability to sample from generative distributions. This architecture allows us to simulate the complex, non-linear contact dynamics of extraterrestrial domains across tens of thousands of unique morphological variations, providing an unprecedented, highly scalable testbed for space robotics.

III. Procedural Engine For Systematic Diversity

The foundation of our approach requires a dedicated engine designed to generate virtually limitless semantic and morphological variety. We introduce an open-source procedural engine and extensible PCG framework specifically engineered for the on-demand generation of space-relevant robot learning scenarios. It functions as a highly flexible asset factory by utilizing a software design pattern that deliberately decouples the computational complexities of procedural content creation from the strict real-time constraints of the reinforcement learning simulation itself.

A. Modular Architecture

As illustrated in Fig. 2, our procedural engine implements a modular software architecture built upon three primary abstractions to ensure rapid scalability:

Assets are declarative blueprints containing the parametric constraints, boundaries, and logical rulesets that define the general class of an object, such as the statistical likelihood of craters on a lunar surface or the bounding boxes of a modular robotic tool.

Generators are the backend computational components that automate the creation of these digital assets from their declarative blueprints, functioning as interpreters that interface directly with external digital content creation APIs.

Integrations are lightweight connectors that seamlessly connect generators to external frameworks such as SRB. They leverage domain-specific hooks to concurrently request, import, and dynamically configure the generated assets into parallel simulation pipelines.

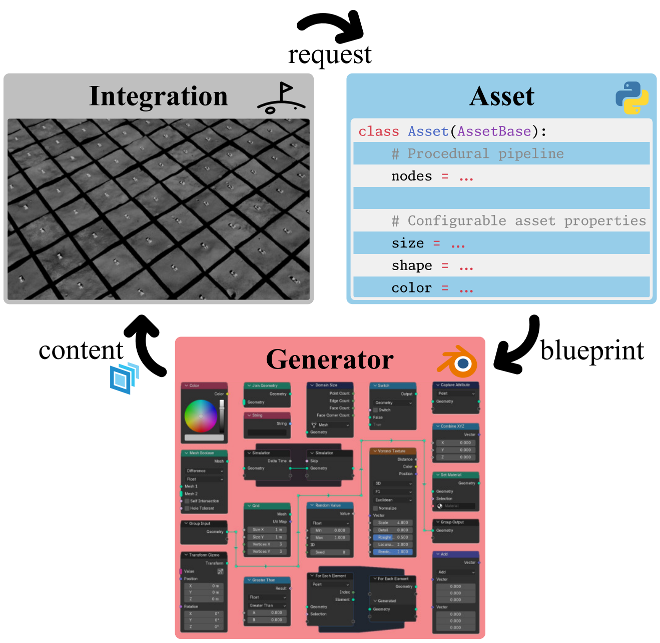


Fig. 2: The modular architecture of our procedural engine incorporates assets that define the parametric constraints of object classes, generators that automate the creation of assets from these blueprints, and integrations that seamlessly link the generated content to external robot learning frameworks.

B. Parametric Generation with Blender

The primary generation backend in our procedural engine is heavily integrated with the headless execution capabilities of Blender [11]. The generator leverages non-destructive, node-based computational systems, specifically Geometry Nodes and Shader Nodes, to construct deeply complex and parametric asset blueprints. By exposing high-level architectural parameters like crater density, noise scales, fractal dimension, and mesh resolution directly to a Python API, the engine seamlessly maps programmatic requests to massive structural and geometric mutations. Consequently, rather than producing one static twin, the engine generates an almost infinite family of unique digital assets by deterministically sampling these structural parameters from seeded pseudo-random distributions, as illustrated in Fig. 3.

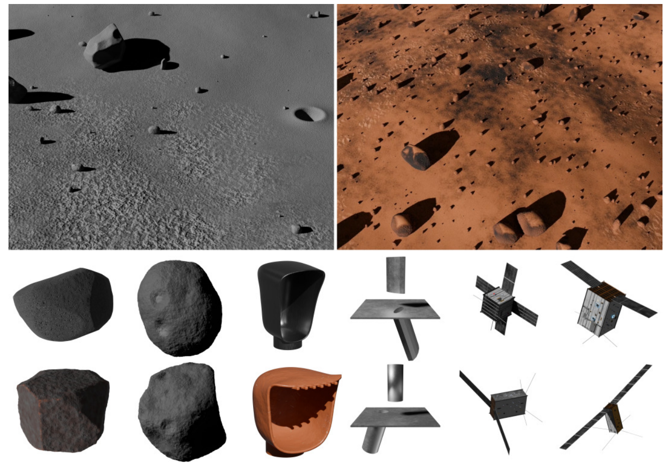


Fig. 3: Diverse space-relevant assets synthesized by the procedural engine. The procedural pipelines dynamically generate multi-scale planetary terrains, structurally varied celestial bodies like asteroids, and parametrically altered mission-critical hardware.

To serve as an example for planetary operations, the generation of a lunar surface cousin operates as a multi-stage process that systematically builds complexity through layered scale-invariance. A high-resolution base mesh is initially displaced by low-frequency Perlin noise to establish rolling macro-scale topography. Subsequent computational layers apply higher-frequency noises: cellular Voronoi patterns carve out sharp, highly variable crater rims while turbulent Musgrave noise simulates long-term micrometeorite erosion. A secondary PCG compositional pipeline subsequently generates hundreds of topologically unique rocks, which are then instanced across the terrain using Poisson disk sampling to ensure a natural, non-uniform spatial distribution without artificial overlap.

Geometric complexity is intrinsically linked to visual realism via a dynamic procedural material system. By leveraging the world-space orientation of geometric surface normals, the shader realistically simulates the accumulation of fine regolith dust on upward-facing planes while leaving steeper, gravity-sheared slopes exposed as bare rock.

C. Automated Export Workflow

Dynamically evaluating massive procedural node trees at runtime is computationally prohibitive for high-throughput RL pipelines, which require thousands of environment resets per minute. To maintain simulator performance, the engine programmatically finalizes and optimizes the digital cousins immediately prior to simulation deployment. A fully automated and headless export pipeline ensures that generated

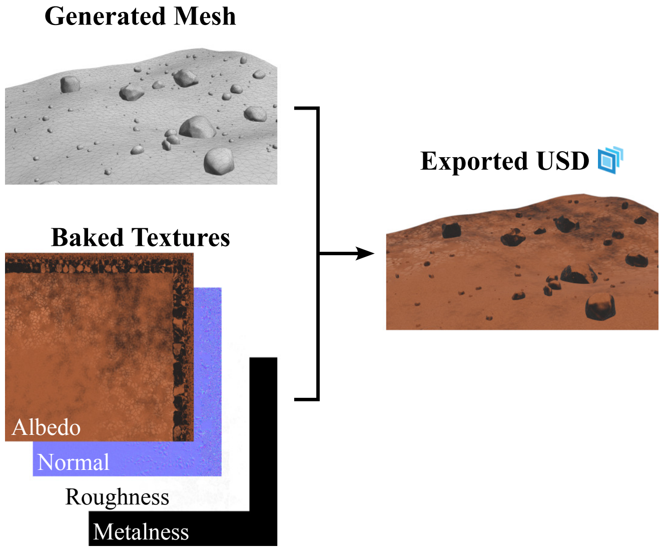


Fig. 4: The automated export process of the procedural engine. Because evaluating massive node trees at runtime is computationally prohibitive for RL, the engine bakes the procedural materials into standard PBR textures and fuses them into highly optimized, portable USD assets ready for immediate tensorized physics simulation.

content is instantly usable without manual intervention.

As visualized in Fig. 4, the engine projects and bakes the complex procedural materials of a generated 3D cousin into a highly optimized set of standard Physically-Based Rendering (PBR) textures. This step bakes the complex optical appearance into a memory-efficient format, which preserves the VRAM necessary to simulate thousands of these cousins in parallel. In the case of SRB, the final structurally optimized mesh and baked PBR textures are exported directly into the Universal Scene Description (USD) format for frictionless integration with the underlying NVIDIA Omniverse platform. However, the modular architecture of the engine allows for flexible export to any desired format or simulation framework.

IV. Scalable Simulation & Randomization

The procedural engine provides the critical architectural capability to generate morphological variance. Its maximum utility is realized through batched and programmatic integration with SRB. Formally, instead of training an agent to minimize a loss function within a single and highly deterministic digital twin, we continually optimize the neural policy over a dynamically shifting procedural diversity.

A. High-Performance Parallelism

Leveraging this wide environmental variety effectively requires a simulation architecture capable of instantiating, managing, and rapidly resetting thousands of unique topological worlds in parallel. SRB achieves massive concurrency via a GPU-accelerated backend built upon NVIDIA Isaac Lab [4]. By keeping the entire physics step, collision detection pipeline, and reward calculation constrained entirely within the GPU memory space, we eliminate significant data transfer bottlenecks. Consequently, computationally lighter cousins can consistently execute over 100k simulation steps per second with 1024 parallel environments. Even highly complex, contact-rich manipulation scenarios with extensive articulations plateau near an impressive 15k steps per second on a single consumer workstation equipped with an NVIDIA RTX 4090 GPU, an Intel Core i9-13900K processor, and 64GB of DDR5 RAM. This extreme throughput is the catalyst that transforms the data-heavy digital cousin paradigm into a highly practical, iterative pipeline.

B. Extensive Domain Randomization

Acting as an essential complement to topological variance, the framework applies an exhaustive, multi-modal layer of domain randomization. Across physics, randomized parameters include sweeping variations to gravitational constants, robotic link inertial matrices, center-of-mass offsets, joint stiffness limits, motor latencies, and diverse contact dynamics in the form of varying friction coefficients, restitution, and parameters of granular soil. On the visual front, optical modalities are perturbed through randomized global illumination, harsh directional solar angles replicating deep-space shadowing, and synthetic sensor noise profiles applied to camera lenses. This ensures that every visual observation and physical interaction within the procedurally generated world is completely unique, severely penalizing neural memorization and systematically reducing the sim-to-real gap.

V. Digital Cousins of Terrestrial Analogues

Despite their overwhelming advantages for generalization, unconstrained procedural distributions risk synthesizing environments with physically implausible dynamics that do not reflect reality. To ensure policies remain viable for deployment, we employ digital cousins of terrestrial analogues. We anchor the foundational parameters of our generative cousins to properties of two physical analogue facilities while permitting their structural topologies to vary infinitely.

A. Lunar Analogue

A physical lunar analogue facility, Lunalab [12], serves as our terrestrial analogue testbed for evaluating planetary mobile manipulation. The facility is an environmentally controlled room filled with fragmented basalt gravel, which was specifically chosen to emulate the complex geotechnical sinkage and shear challenges of lunar regolith. The hardware suite features multiple rovers equipped with articulated robotic manipulator arms, as shown in Fig. 5.

Attempting to meticulously map the precise location of every microscopic rock to construct a singular exact replica yields a completely obsolete digital twin, the exact moment a physical rover executes a single drive command through the soil. Instead, we generate distributions of digital cousins. The simulation spawns dynamically varying distributions of craters, varying incline gradients, and dense rock fields. Furthermore, we can simulate a large number of these cousins in parallel, each with different randomized physics

(a) Physical Lunar Analogue Facility (b) Single Digital Cousin Variant (c) Scalable Cluster of Generative Cousins

Fig. 5: Digital cousin of the lunar analogue: (a) The physical testbed features rovers navigating highly deformable basalt gravel; (b) A single variant of a digital cousin with randomized terrain geometry; (c) A parallel group of digital cousins with varying environment topologies and physics parameters.

parameters for soil cohesion and friction. By training across this vast distribution, the agent learns to implicitly generalize the underlying terramechanics of granular media, which is critical for robust sim-to-real transfer in this domain.

B. Orbital Microgravity Analogue

For mission profiles like in-space assembly and active debris removal, robotic systems must dexterously manipulate objects under microgravity conditions. Our physical microgravity facility emulates this exceptionally challenging and frictionless environment via low-friction floating air-bearing platforms on top of a flat epoxy floor. Furthermore, the facility is equipped with two robotic arms mounted on ceiling and wall linear rails, which can be configured to simulate a wide variety of robotic morphologies relevant for orbital operations, as shown in Fig. 6.

A single digital replica fails to capture the variability of space debris. Therefore, our digital cousins randomize the physical and visual parameters of the environment. By training across this distribution, the agent is incentivized to learn robust momentum exchange and control policies that are invariant to the specific mass distribution of the target debris, the exact center-of-mass of the robotic platform, or the precise thruster latency. This ensures that the learned policies are not overfit to a single, static model of microgravity, but rather have implicitly generalized the underlying physics of orbital manipulation across a wide range of potential real-world conditions. A virtual counterpart of the free-floating platform is shown in Fig. 7.

VI. Benchmarking Suite

To standardize robust evaluation across these generative cousins, we formulated a suite of diverse task scenarios as Partially Observable Markov Decision Processes (POMDPs) utilizing the Gymnasium API.

As illustrated in Fig. 8, the benchmark suite extensively encompasses mobile robotics, fixed-base manipulation, and mobile manipulation. To strongly support real-world end-to-end visuomotor control, observation spaces are strictly architected to explicitly isolate privileged simulator states like the exact ground-truth coordinates and hidden physics parameters from deployable sensor data, such as noisy joint proprioception, raw depth maps, RGB feeds, and IMU streams. This data segregation ensures the fully deployed neural policy only expects and reacts to imperfect data streams it can actively receive on the physical hardware.

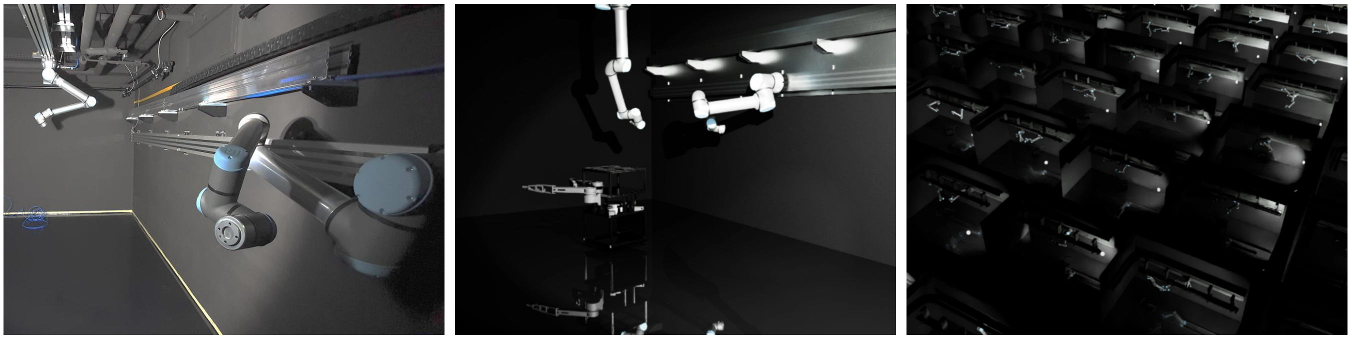
VII. Sim-to-Real Transfer Case Studies

To conclusively validate the generative digital cousin approach, we analyzed standard algorithmic baselines and executed zero-shot sim-to-real transfers directly onto physical robots operating dynamically in unstructured environments.

A. Zero-Shot Sim-to-Real on Deformable Granular Media

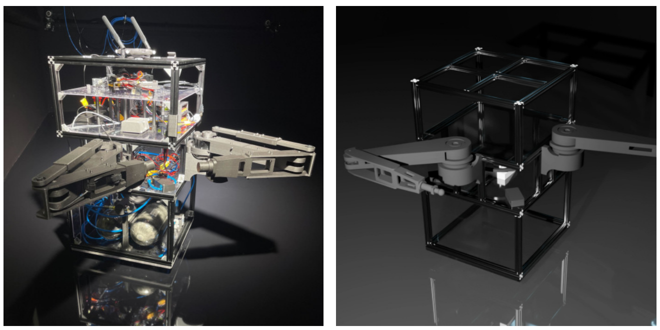
The primary validation of any generative pipeline is unassisted physical deployment. We tested this via a continuous dynamic waypoint tracking task, successfully transferring neural policies trained solely inside the simulated digital cousins directly to a physical Leo Rover. The four-wheeled skid-steer rover was deployed in the lunar analogue facility and navigated shifting basalt gravel that introduces severe, non-linear terramechanics such as variable wheel slip and sinkage.

To systematically evaluate policy robustness, the agent was commanded to follow a pre-configured looping capsule trajectory, with the target waypoint moving at constant velocities of 5, 15, and 25 cm/s. The policy operated at a 25 Hz control frequency. To isolate the evaluation of the control policy from potential onboard state estimation errors, ground-truth localization was provided by an OptiTrack motion capture system. The agent received a sparse observation vector containing the relative 2D position to the target waypoint and a 2D vector representing the sine and cosine of the relative yaw orientation error, injected with both persistent and transient synthetic noise to emulate real-world sensor jitter. The action space consisted of a 2D continuous vector mapped to the rover's desired linear and angular velocities.



(a) Physical Microgravity Facility (b) Single Digital Cousin Variant (c) Scalable Cluster of Generative Cousins

Fig. 6: Digital cousin of the microgravity analogue: (a) The physical testbed features robotic arms and epoxy floor; (b) A single variant of a digital cousin; (c) A parallel group of digital cousins with varying environmental conditions.



(a) Physical Platform (b) Simulated Platform

Fig. 7: The real and simulated versions of the free-floating platform used within the microgravity analogue.

B. Algorithmic Comparison

Our first experiment identified the most suitable reinforcement learning paradigm for this domain. We evaluate three algorithms that represent distinct learning paradigms. Proximal Policy Optimization (PPO) [13] is a standard on-policy algorithm, and we also test a variant with a Long Short-Term Memory (LSTM) [14]. Twin Delayed Deep Deterministic Policy Gradient (TD3) [15] represents modern off-policy methods. Finally, DreamerV3 [16] is selected as a representative model-based algorithm due to its demonstrated performance across diverse tasks.

TABLE I: SIM-TO-REAL PERFORMANCE AND INFERENCE LATENCY OF 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	13h30 (100M)	25h00 (100M)	15h00 (100M)	17h30 (20M)
Infer (CPU)	**0.24±0.1 ms**	0.71±0.2 ms	0.43±0.1 ms	2.38±0.2 ms

As shown in Table I, DreamerV3 achieved a substantially lower Average Tracking Error (ATE) across all tested velocities. At 15 cm/s, DreamerV3 recorded a highly precise ATE of just 3.3 cm and 1.9°. Furthermore, the model-based agent exhibited vastly superior sample efficiency, converging in just 20M simulation steps compared to the 100M steps required by the model-free algorithms. This superior accuracy is qualitatively confirmed in Fig. 9, which illustrates the real-world trajectories of the different agents. The path traced by the DreamerV3 policy is visibly smoother and more closely aligned with the target trajectory, while other agents exhibit larger deviations and more erratic behavior.

C. Impact of Procedural Diversity

To empirically isolate the specific impact of our generative digital cousin methodology, we conducted a rigorous physical ablation study comparing training regimes with varying degrees of diversity and fidelity:

1) **Static Twin:** A single procedural terrain shared across all parallel environments.
2) **DR:** Static terrain augmented with extensive Domain Randomization (physics, noise, latency).
3) **DR & PCG (Digital Cousins):** Full procedural generation providing each parallel agent with a unique topological terrain alongside DR.
4) **DR & PCG + PF:** The median Digital Cousin agent fine-tuned across 8 parallel environments using high-fidelity discrete particle physics.

TABLE II: PERFORMANCE ACROSS TRAINING REGIMES

	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°

As detailed in Table II, relying on a Static Twin results in significant real-world degradation. Conversely, the fully generative *DR & PCG* approach achieves the lowest ATE at higher velocities. At 15 cm/s, this procedural cousin methodology exhibited a $\sim$21% lower position error (from 4.2 cm to 3.3 cm) and a massive 72% reduction in orientation error (from 6.8° to 1.9°) compared directly to the static baseline.

Interestingly, fine-tuning with computationally expensive discrete particle physics (*+PF*) provided only a marginal 0.1 cm improvement at the lowest speed (5 cm/s) and actually degraded performance at 25 cm/s (error increased to 4.3 cm).

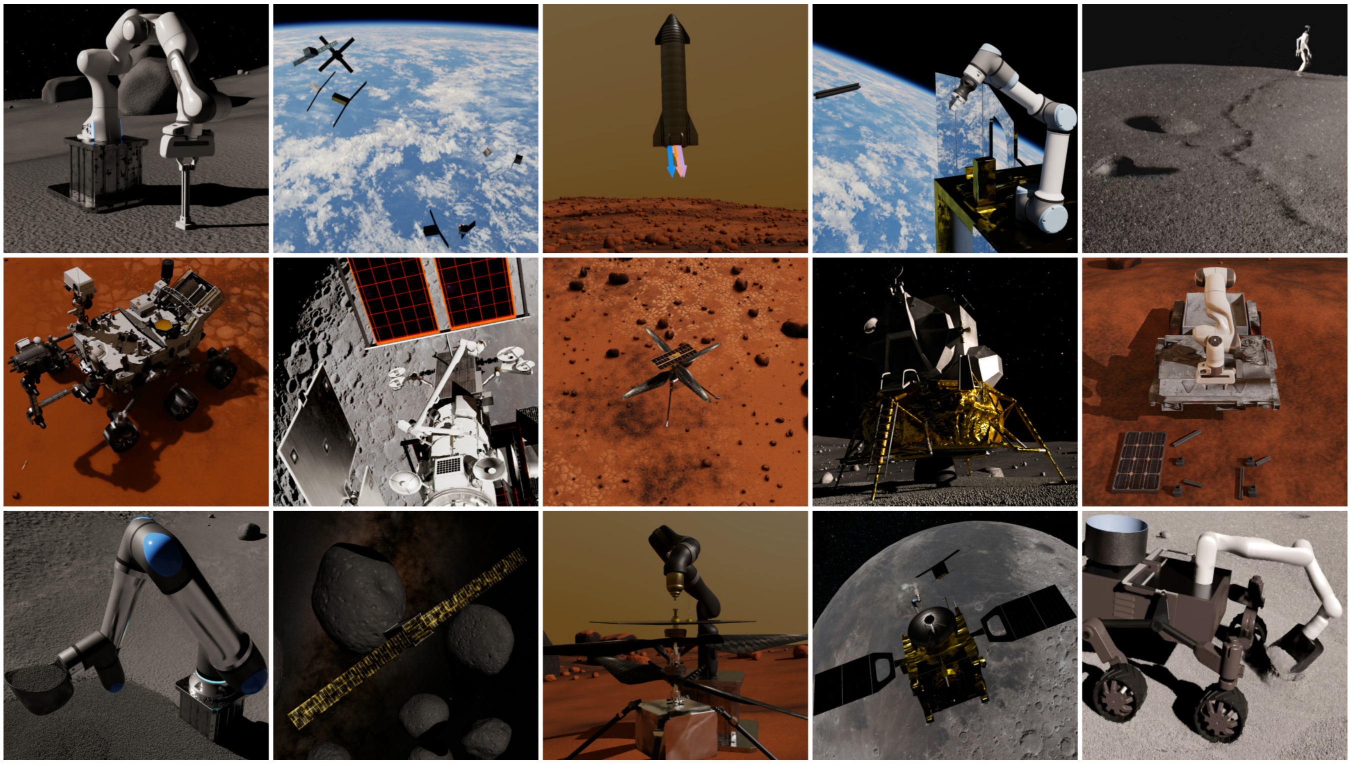


Fig. 8: The diversity of tasks, domains, and robot morphologies across the Space Robotics Bench.

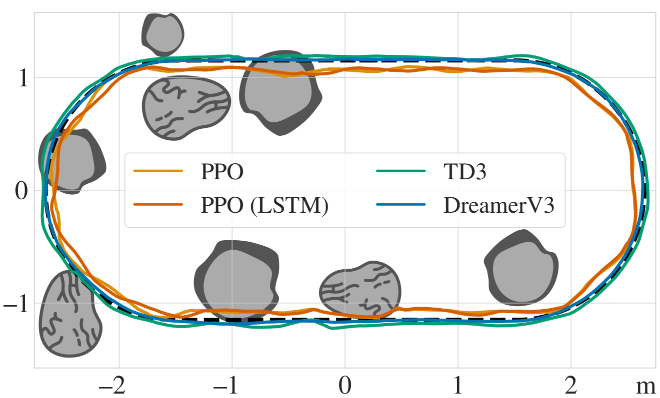


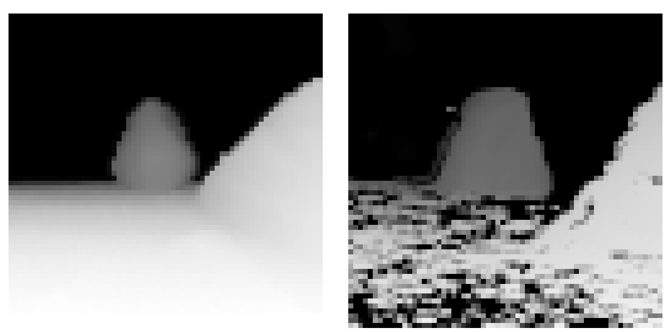
Fig. 9: Real-world *capsule* trajectory for policies trained with different RL algorithms. Major environmental obstacles (craters and boulders) are graphically indicated for context.

By restricting the fine-tuning data distribution to only eight particle-enabled environments, the policy lost some of the broad generalizability it acquired from the vast procedural diversity. This strongly reinforces our core hypothesis: high-throughput morphological diversity acts as a far more critical and cost-effective strategy for generalization than hyper-realistic, high-fidelity physical modeling.

D. Vision-Based Control Limitations

Finally, we investigated the feasibility of end-to-end visuomotor control by deploying a policy trained with access to a 64×64 pixel depth map acquired from an onboard RealSense D455 camera. While the zero-shot transfer was functional, the tracking accuracy severely degraded to an ATE of 9.2 cm and 6.5° at 15 cm/s.

We attribute this drop directly to the profound perceptual sim-to-real gap of the unstructured extraterrestrial analogue. While the generative simulation successfully produces clean geometric depth maps shown in Fig. 10, the highly specular and porous physical basalt gravel creates a noisy signal plagued with substantial infrared dropouts. This result highlights that while procedural diversity resolves terramechanic uncertainty, closing the perceptual gap for complex granular media via higher-fidelity generative sensor noise modeling remains a critical frontier for future work.



(a) Simulated Depth Map (b) Real-World Depth Map

Fig. 10: The perceptual sim-to-real gap for depth maps.

VIII. Discussion and Limitations

Our experiments demonstrate that exposing agents to a vast generative distribution of environments is a highly effective strategy for bridging the kinodynamic sim-to-real gap. To the learning agent, the physical lunar analogue is no longer an out-of-distribution domain, but simply another procedural variation. This prevents spatial memorization and forces the generalization of underlying terramechanic principles.

Crucially, our findings highlight that scaling procedural diversity is far more effective and computationally tractable than integrating high-fidelity discrete particle physics. Because extreme computational costs restrict particle-based training to a narrow set of environments, policies are prone to overfitting and lose broad generalizability.

Despite the success of proprioceptive and kinodynamic transfer, a significant limitation remains in the perceptual sim-to-real gap. Vision-based control suffered due to unmodeled optical artifacts, specifically severe infrared dropouts caused by the highly specular and porous basalt gravel. Closing this optical gap without incurring the prohibitive computational costs of real-time ray tracing inside the RL loop remains a critical challenge. Future work will explore integrating lightweight generative AI, such as real-time image-to-image diffusion models, to act as active domain adaptation filters that match the complex noise profiles of physical space optics.

IX. Conclusion

This paper formalizes a scalable framework for resilient space robotics via generative digital cousins. By discarding the restrictive need for singular, exact digital twins, we present the Space Robotics Bench to procedurally synthesize vast environment distributions anchored to empirical analogue testbeds.

Our successful zero-shot transfer of dynamic waypoint tracking to a physical rover on granular media confirms that massive procedural diversity fundamentally outperforms localized, high-fidelity physical modeling for real-world generalization. Combined with sample-efficient model-based RL, this generative paradigm provides a validated and computationally viable pathway for deploying reliable autonomous systems across the unpredictable and unstructured environments of the final frontier.

References

[1] J. Liu, X. Zeng, C. Li, X. Ren, W. Yan, X. Tan, X. Zhang, W. Chen, W. Zuo, Y. Liu, B. Liu, D. Liu, Q. Zhou, and Z. Ouyang, "Landing Site Selection and Overview of China's Lunar Landing Missions," *Space Science Reviews*, vol. 217, no. 6, 2021.

[2] 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.

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

[4] M. Mittal, C. Yu, Q. Yu, J. Liu, N. Rudin, D. Hoeller, J. L. Yuan, R. Singh, Y. Guo, H. Mazhar, A. Mandlekar, B. Babich, G. State, M. Hutter, and A. Garg, "Orbit: A Unified Simulation Framework for Interactive Robot Learning Environments," *IEEE Robotics and Automation Letters*, vol. 8, no. 6, pp. 3740–3747, 2023.

[5] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "Nerf: Representing scenes as neural radiance fields for view synthesis," in *ECCV*, 2020.

[6] A. Byravan, J. Humplik, L. Hasenclever, A. Brussee, F. Nori, T. Haarnoja, B. Moran, S. Bohez, F. Sadeghi, B. Vujatovic, and N. Heess, "Nerf2real: Sim2real transfer of vision-guided bipedal motion skills using neural radiance fields," *arXiv preprint*, 2022.

[7] M. Deitke, E. VanderBilt, A. Herrasti, L. Weihs, J. Salvador, K. Ehsani, W. Han, E. Kolve, A. Farhadi, A. Kembhavi, and R. Mottaghi, "Procthor: Large-scale embodied ai using procedural generation," *arXiv preprint*, 2022.

[8] 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.

[9] 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.

[10] M. Towers, A. Kwiatkowski, J. Terry, J. U. Balis, G. De Cola, T. Deleu, M. Goulão, A. Kallinteris, M. Krimmel, A. KG, *et al.*, "Gymnasium: A Standard Interface for Reinforcement Learning Environments," *arXiv:2407.17032*, 2024.

[11] Blender Foundation and Community, "Blender," 2025. [Online]. Available: http://blender.org

[12] 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 *IAC*, 2020.

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

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

[15] 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.

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