# The Neverwhere Visual Parkour Benchmark Suite

**Anonymous Author(s)**
Affiliation
Address
email


Figure 1: **The Neverwhere Benchmark Suite.** We offer over *sixty* high-quality Gaussian splatting-based evaluation environments, and the Neverwhere graphics tool-chain for producing accurate collision mesh. Our aim promote reproducible robotics research via fully automated, continuous testing in closed-loop evaluation.

## Abstract

State-of-the-art visual locomotion controllers are increasingly capable at handling complex visual environments, making evaluating their real-world performance before deployment increasingly difficult. This work intends to narrow this train/evaluation gap by developing a collection of hyper-photo-realistic, closed-loop evaluation environments – The Neverwhere Benchmark Suite – comprised of over sixty 3D Gaussian Splatting of urban indoor and outdoor scenes. Our goal is to encourage large-scale and reproducible robot evaluation by making it easier to create and integrate Gaussian splats-based reconstructions into simulated continuous testing setups. We also underscore the potential pitfalls of relying exclusively on 3D Gaussian-generated data for training, by providing policy checkpoints trained over multiple Neverwhere scenes and their performance when evaluated in novel scenes. Our analysis illustrates the necessity of sourcing diverse data to ensure performance. Anonymous Website: link

Submitted to 39th Conference on Neural Information Processing Systems (NeurIPS 2025). Do not distribute.

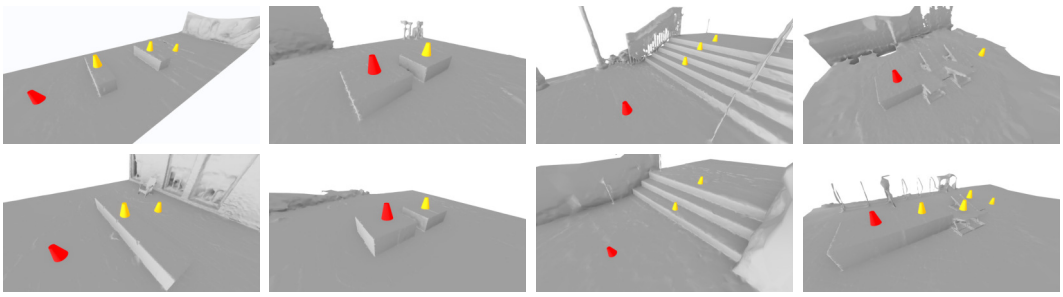


Figure 2: **Task Waypoints Layout.** From left to right: Hurdle, Gaps, Stairs, Ramp. The robot starts from the right. **Yellow cones** indicate waypoints, whereas **red cones** are the final goal.

## 1 Introduction

The past few years witnessed a rapid acceleration in progress in robotics. Data-driven, general-purpose learning algorithms, which treat specific tasks as data points from a general problem class, proved to be the scalable approach to producing robots that are robust, capable, and intelligent. As our robots graduate the confined lab environments to face the open world, real-world evaluation and hand-crafted simulation environments are proving insufficient. We need an evaluation strategy that is equally scalable to quantify progress. How do we create abundantly diverse and realistic environments to test our robot?

This work aims to develop a scalable approach to testing real-world visuomotor policies in automated, closed-loop simulations. We focus on visual locomotion in legged robots as our test bed, a class of robotic tasks where perception is tightly coupled with actions. Our intention is to start with a domain where the 3D environment is complex, but the physics is relatively simple. Our main contribution is Neverwhere (see Fig. 1), a collection of over 60 high-fidelity digitally recreated scenes that covers diverse urban structure, including stairs, speed bumps, indoor carpeted lab spaces and the outdoor, with and without vegetation. An equally essential objective is to empower the community to build their own set of benchmarks.

The Neverwhere tool-chain address three essential challenges in building evaluation environments for robots: The primary challenge is to capture the world in its full messiness which exceeds the expressivity of traditional 3D mesh. The second challenge is that in practice, quadruped robots observe the world from an angle that sits out of the distribution of human camera views. This coupled with the extraordinary expressivity of the Gaussian substrate, results in poorly rendered ego camera input. The final challenge is about geometry. It remains difficult, in practice, to obtain detailed collision mesh from 3D Gaussian that are modeled using hand-held iPhone videos. We solve all three challenges, by developing a better initialization scheme that takes advantage of traditional multi-vew stereo reconstruction (Sec. 4).

Our contributions are summarized as follows:

- We introduce the Neverwhere benchmark suite, featuring over 60 high-quality environments powered by 3D Gaussians, encompassing a diverse range of urban indoor and outdoor scenes.
- We present a data collection toolchain that facilitates the generation of new benchmark environments with minimal human intervention, allowing users to create reconstructed scenes directly from uncalibrated images or videos.
- We provide and release visual parkour policy checkpoints trained directly on the Neverwhere 3D Gaussian environments, offering baseline results to support further research and exploration.

## 2 Related Works

Robotics research has a long tradition of using physics simulation engines to evaluate planners and policies prior to real-world deployment [6, 27]. More recently, improvements in learning-based approach has enabled neural controllers to directly map high-dimension visual data into joint configurations [7, 5] while also expanding along the dimension of the *number* of skills learned by a

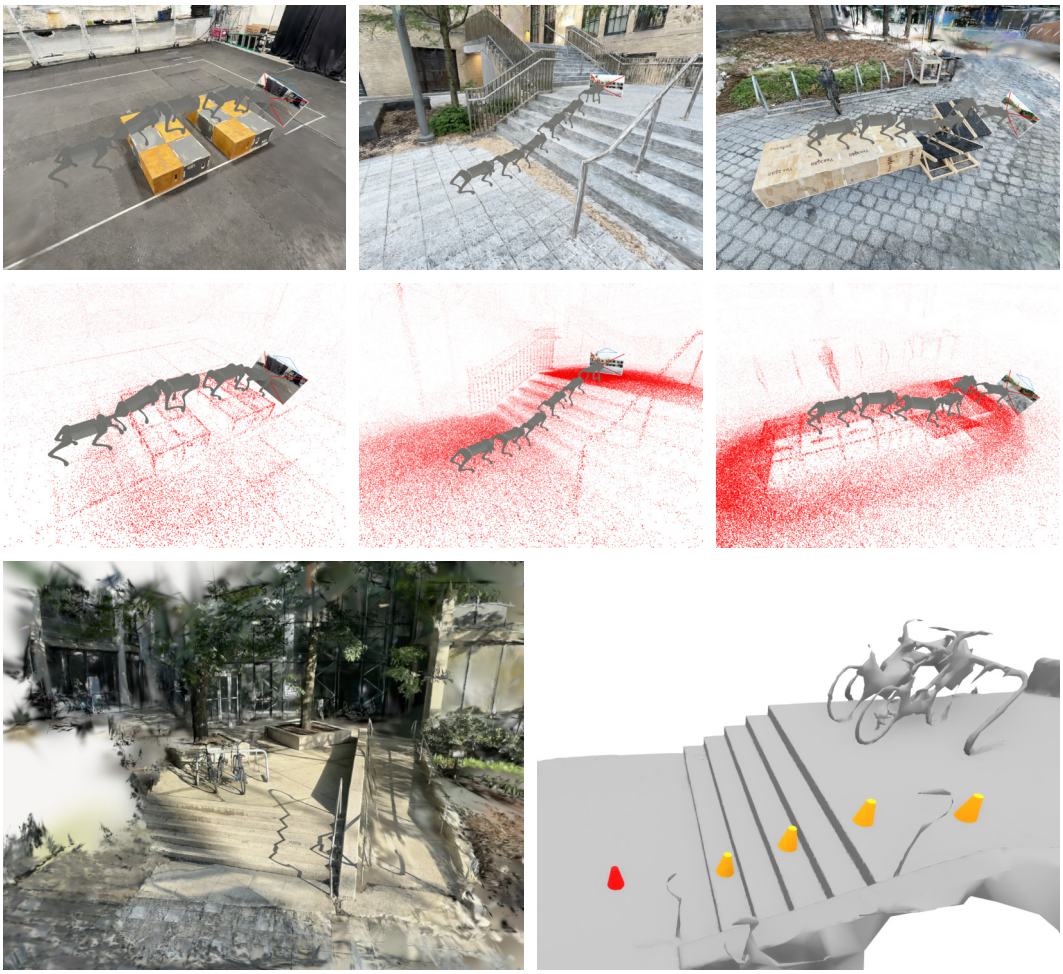


Figure 3: **Example Scenes and Collision Geometry.** Showing a subset of the environments, on four parkour locomotion tasks: Hurdles, Gaps, Stairs, and Ramps. The bottom row displays annotated trails with labeled waypoints that define each evaluation task. All images are rendered.

single, general-purpose control policy [34, 1]. In contract, the rendering setup used by early robotics benchmarks [11] was relatively primitive as the intended purpose was for humans to visualize and debug failure, as opposed to simulating accurate camera sensor readings for a policy.

Both the quality, and the range of physics and material have been improved significantly in more recent physics engines such as IssacSim [22] and ManiSkill3 [8, 26], extending coverage to deformable material, fluid, and caustics. Despite these improves, lack of 3D content remains a bottleneck. Recent benchmark efforts significantly raised the bar: BiGym [4], for instance, provided a high-quality, manually CAD'ed 3D collision mesh for an articulated dish washer. At the scene level, RoboCasa [20] provides fourteen manually designed kitchen scenes.

Neverwhere differs from these prior efforts [21, 16, 10, 15, 4] in two ways: First, advancements in neural scene representation made investment in traditional assets and lighting setup less critical. Neverwhere uses 3D Gaussian Splatting to replace mesh-based rendering, which not only simplifies construction but also enables the creation of highly detailed digital replicas, which can be done by the end-user using Neverwhere's open-source toolkit. Second, Neverwhere builds upon the MuJoCo [27] physics engine and aims to provide accurate collision geometry. It does so without requiring LiDAR sensors and depth measurements. The closest are simulators from autonomous driving that are used for closed-loop evaluation of self-driving vehicles (SDV). Among them, UniSim [30] uses detailed mapping data to create the environment and replay pre-recorded driving episodes to produce safety-critical scenarios involving pedestrians and other vehicles. Neverwhere is similar in spirit but currently lacks UniSim's advanced scene decomposition and the ability to animate other actors.

Nevertheless, our goal is to use Neverwhere as a foundation for building open-source, customized evaluation setups for robotics, with plans to integrate more sophisticated capabilities in the future.

Generative AI is increasingly being used in robotics to generate task scenarios, rewards, and assets. Platforms such as RoboCasa, RoboMimic, and MimicGen [20, 29, 18] employ generative models to diversify training environments. However, Neverwhere takes a different approach by emphasizing photo-realism and accurate modeling over diversity. We argue that when it comes to evaluation benchmarks, a deterministic, curated set of challenging test cases prioritizes signal over noise. Neverwhere's use of 3D Gaussian splats is better suited for evaluation than for training, as the former favors reduced variability.

# 3 Tasks and Reinforcement Learning Setup

Neverwhere offers four types of parkour tasks. For each, there is a group of physical environments, created from 3D scans of two university campuses, featuring both outdoor and indoor domains with different obstacle layouts and appearances. The tasks were written in Python, and physics simulation is implemented with MuJoCo [27]. The locomotion setup follows:

**Action.** The action space consists of twelve target joint positions for the quadruped robot, with each of the robot's four legs having three actuated joints: hip abduction/adduction, hip flexion/extension, and knee flexion/extension.

**Observation.** The observation includes the robot's ego state, $\mathbf{e} = \{v, \mathbf{q}, \dot{\mathbf{q}}\}$, where $v$ is the linear velocity, $\mathbf{q}$ the joint positions, and $\dot{\mathbf{q}}$ the joint velocities, along with the robot's previous actions. For evaluating visual policies, we provide visual observations in different data modalities, including RGB renders from gsplat, depth maps, point clouds, and semantic maps. The rendering pipeline detailed in Sec. 3.2 provide these diverse data modalities for visual policies' input.

**Privileged Observation.** This includes a heightmap of the scene, offering a top-down view of the terrain, which can further be processed into ScanDots observation for lightweight inference [3]. The moving direction, represented as a single angle value, can also be provided to guide navigation.

## 3.1 Parkour Tasks

We designed four challenging scenarios to evaluate a robot's ability to generalize locomotion skills, adapt to varying terrain, and handle physically demanding tasks, following the task design in [3](introducing in **ascending order of difficulty**.): 1) overcoming obstacles of a certain height (**hurdles**), 2) jumping across gaps of varying lengths (**gaps**), 3) navigating sloped surfaces (**ramps**), and 4) walking up stairs (**stairs**).

We labeled waypoints to define a specific trail in each scene, outlining the exact task and target for the robot. We then measured the robot's performance by calculating two metrics: the **Success Rate**, determined by the percentage of waypoints reached, and **X displacement**, the total distance moved in the +X direction. The following section provides detailed task definitions and the common waypoint distribution for each task. Illustrations of the waypoints are listed in Fig. 3.

**Hurdles.** Most hurdle scenes are manually constructed. Each hurdle consists of several boxes arranged side by side to form a low barrier. Typically, 1 to 3 such barriers are placed consecutively within a scene, with waypoints labeled on top of each one. In addition to these manually created scenes, a small portion of the dataset uses natural outdoor elements, e.g. long stone benches, as hurdle obstacles.

**Gaps.** All gap scenes are manually set up, either indoors or outdoors. We construct two box-based platforms with a 12-inch or 16-inch gap between them to simulate jumping over a gap. Waypoints are labeled on top center of the platforms.

**Ramps.** Ramp scenes are designed to test the robot's ability to walk on inclined surfaces. Around four sloped boards are placed in a staggered arrangement alongside a raised platform built from boxes. Waypoints are primarily placed at the end of each trail.

**Stairs.** Stair scenes involve real-world staircases of various heights, materials, and textures, captured both indoors and outdoors. The robot is tasked with climbing up the stairs, and waypoints are placed along the steps.

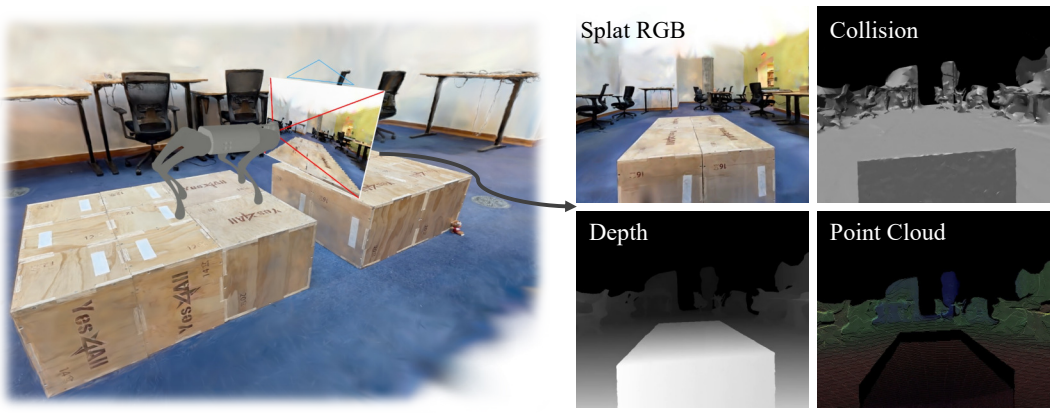



Figure 4: **Rendering Wrappers.** We designed a rendering pipeline that provides diverse wrappers for multi-modal observations, including but not limited to: color images from Gaussian splats, depth maps, heightmaps, and LiDAR projections, to support a wide range of visual based policies.

### 3.2 Rendering Wrappers

**Gaussian Rendering.** We adopt 3DGS [12] for photo-realistic rendering to reduce the domain gap in policy training and evaluation. 3DGS provides high-quality visual observations and real-time performance, making it well-suited for close-looped evaluation. To incorporate visual targets (e.g., cones) that are not present in the original 3DGS scene, we blend them into the rendered view using semantic masks rendered from MuJoCo [27], enabling consistent integration of task-relevant cues. Through experiments (in Sec. 5.3), we found visual cones are strong visual cues that lead to better policies.

**Depth.** Depth is obtained by converting MuJoCo-rendered maps [27] into MiDaS-style inverted depth [23]. These depth maps serve as observation inputs for robot policy learning. They can also be used as conditioning inputs for depth-conditioned generative models [33], enabling robust training and zero-shot transfer to real-world RGB inputs [31].

**Semantics Mask.** We generate semantic masks by grouping objects based on naming rules, enabling simple target-background segmentation for downstream tasks.

**Heightmap.** A top-down bird's-eye-view heightmap is rendered to capture terrain geometry while excluding movable objects, aiding navigation and privileged policy training.

**LiDAR Projection.** Simulated LiDAR rays produce point clouds based on scene geometry, supporting policies that rely on spatial awareness and obstacle detection.

## 4 From Photons to Splats: The Neverwhere Environment Builder

We aim to develop a scalable and efficient toolchain for creating benchmark scenes, enabling users to easily generate their own digital twins. Existing 3DGS techniques [13, 35, 14, 28, 32, 19, 9, 17] excel at visual fidelity, but the resulting meshes may lack the physical accuracy required for reliable physics-based interactions. Our robot needs precise collision geometry to be evaluated correctly. Developing high-fidelity digital replicas for robot simulation necessitates both high visual quality with minimal domain gap and accurate physical modeling of real-world geometry to facilitate robot interaction. By leveraging the power of robust and efficient SfM and MVS modules within a unified pipeline, Neverwhere automatically converts pose-free multi-view images into registered pairs of 3D Gaussians and high-quality collision geometry. Thus, by combining the strengths of 3D Gaussians for appearance representation with spatially aligned meshes for the robot simulation platform, we construct a complete physics-aware robot simulation environment.

In detail, given a set of $N$ uncalibrated images $\mathcal{I} = \{\mathbf{I}_i\}_{i=1}^{N}$, a camera pose estimation module $\Theta$ is used to estimate their poses (Fig. 5-(2)), yielding $\mathcal{P} = \{\mathbf{P}_i\}_{i=1}^{N}$. Subsequently, a mesh reconstruction

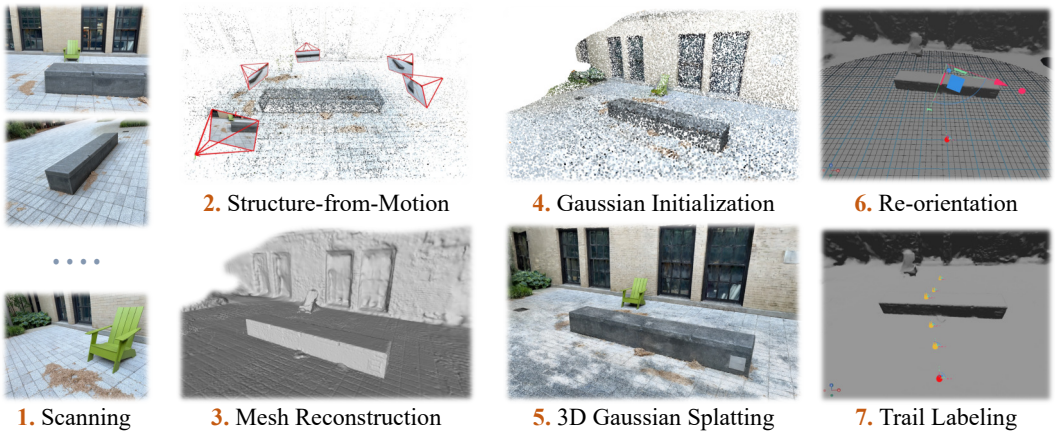



Figure 5: **Neverwhere Toolchain.** The toolchain takes (1) multi-view images as input and follows a sequential process: (2) A Structure-from-Motion modules is applied to obtain camera calibrations, (3) An optimization-based MVS module is used to estimate the scene geometry, (4) Points are sampled from the textured meshes, which are then used as initializations for (5) training 3D Gaussians to model the scene. Once the scene modeling is complete, human input is required for (6) reorienting the mesh to align with scene conventions and (7) labeling waypoints for visual parkour policies.

module $\Phi(\mathcal{I}, \mathcal{P})$ is employed to recover the scene mesh (Fig. 5-(3)). In our reconstruction toolchain, COLMAP [24, 25] is utilized as $\Theta$ for camera calibration, and OpenMVS [2] serves as $\Phi$ to process the calibrated poses and images into a fine-grained collision mesh $\mathcal{M}$.

Following this geometric reconstruction, we optimize 3D Gaussian Splats for scene appearance modeling (Fig. 5-(5)). Vanilla Gaussians are prone to overfitting to training views, often resulting in suboptimal novel view rendering quality when the viewpoint significantly deviates from the training data (e.g., views of a quadruped robot versus typical handheld camera views). To address this, we introduce geometrical constraints to achieve better novel view rendering. Therefore, we extract depth maps $\mathcal{D} = \{\mathbf{D}_i\}_{i=1}^{N}$ and confidence maps $\mathcal{C} = \{\mathbf{C}_i\}_{i=1}^{N}$ for each image in $\mathcal{I}$ from the patch-matched geometric cache of $\mathcal{M}$. These maps are then used to supervise 3D Gaussian training with the following loss term:

$$\mathcal{L} = (1 - \lambda_r) \left\| \mathbf{I} - \hat{\mathbf{I}} \right\|_1 + \lambda_r \mathcal{L}_{\text{SSIM}} + \lambda_D \left\| \mathbf{C} \odot (\mathbf{D} - \hat{\mathbf{D}}) \right\|_1 \tag{1}$$

Here, $\lambda_D$ is the weight for depth supervision. In practice, we initialize Gaussians with sampled colored points (Fig. 5-(4)) from the collision mesh $\mathcal{M}$, as this provides improved geometric priors compared to the standard initialization using SfM points.

SfM methods recover scene geometry up to an arbitrary scale and orientation for uncalibrated images. Thus, the reconstructed Splats and Mesh are not aligned with real-world scale or the z-up convention, making them unready for robot simulation. Additionally, evaluating parkour policies requires task definitions within the scene, such as waypoint trails to measure robot success rates. To address this, we developed an intuitive labeling tool that lets users quickly reorient and rescale the mesh to match real-world coordinates (Fig. 5-(6)) and define tasks by labeling waypoints (Fig. 5-(7)). After labeling, the system automatically generates simulation task configurations for the benchmark environments. The full process takes about two minutes (see supplementary materials for demo).

**Improving 3DGS Modeling from Hand-held iPhone Videos.** Accurate collision geometry is essential for precise contact and physics simulation. The typical solution involves using multi-view stereo methods to generate this geometry. However, geometry generated by mobile applications often lacks the details required for fine-grained contact simulation due to the limited computational power of mobile devices. Furthermore, these methods often require a depth sensor for robust reconstruction. We propose an alternative approach using OpenMVS to reconstruct physical geometry with enhanced quality and provide additional cache to improve 3DGS quality. This method produces high-quality, cost-effective geometry without requiring external sensors. As shown in Fig. 6, the meshes generated

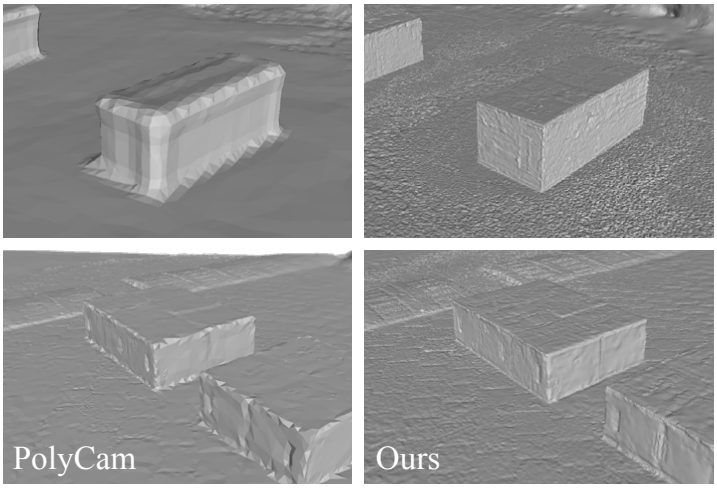



Figure 6: **OpenMVS provides detailed geometry.** (left) mesh taken from a consumer 3D scanning app PolyCam. (right) reconstruction from OpenMVS.

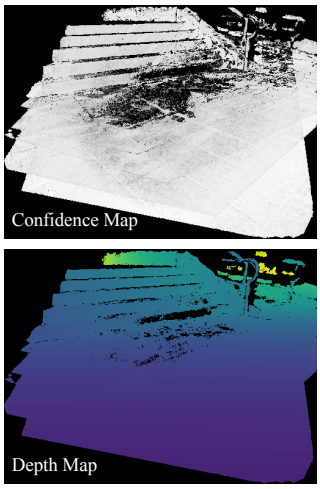



Figure 7: **Confidence and depth maps.** We extract these from the SfM pipeline.

by OpenMVS [2] exhibit fine geometric details and provide good scene coverage, matching or even surpassing the results from on-device mobile software.

**Geometry Guided Gaussian Initialization.** To achieve higher-quality 3DGS with improved geometry and consistent depth, we use colored points sampled from the textured mesh produced by OpenMVS [2] for initialization (Fig. 5-(4)). This approach is more geometrically organized than using scattered points from SfM alone, as illustrated in Fig. 8, resulting in better 3DGS for rendering geometry-consistent views from robot's views.

## 5 Experiments

Our design intention is for Neverwhere to be used as part of an automated, continuous testing setup that quickly and scalably uses closed-loop simulation to assess the policy before its real-world deployment. Training and testing environments have different requirements: the former benefits from system coverage and entropy, whereas the latter is better conducted deterministically to maximize interpretability. To explore the capabilities of the Neverwhere benchmark under constrained entropy and limited scenes, we conducted experiments on closed-loop training to provide additional insight. Although the results indicate limited generalization that restricts effective closed-loop training in this context, the benchmark consistently reflects the robot's capabilities from an evaluation standpoint.

### 5.1 Training Setup:

We performed closed-loop training using a teacher-student behavior cloning approach. We re-trained a privileged teacher policy from [3] to provide guidance. We trained both depth-based and RGB-

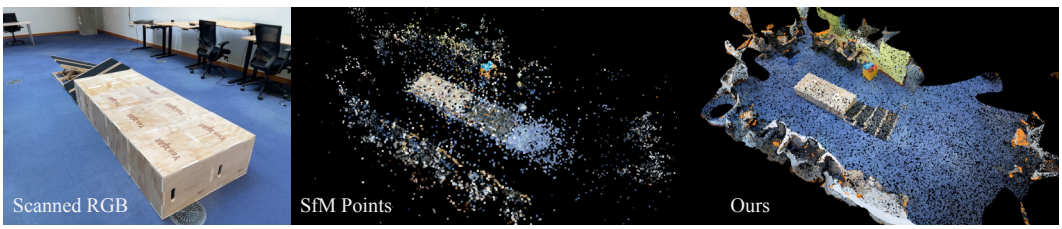



Figure 8: **Gaussians Initialization**. Instead of using SfM points, we utilize colored points sampled from the reconstructed textured mesh generated by our pipeline for initialization, ensuring improved geometric accuracy of the Gaussians.

based visual policies (rendered via a Gaussian Splats wrapper) through on-policy learning. Dataset aggregation (DAgger) was employed, sampling 1,000 trajectories per DAgger iteration. The teacher policy architecture follows the approach described in [3]. For the student policies, we adopted Action Chunking Transformers (ACT) to improve their ability to handle challenging tasks such as gaps and ramps. We tried two settings:

**(1) Single-scene Training:** We trained the policy on a single environment and performed 4 DAgger iterations. We evaluated its performance both on the training scene itself and on other scenes within the same task. Domain randomization was turned off for this setup.

**(2) Multi-scene Training:** We split 70% of the scenes from a specific task to create a training set, and the remaining 30% were used for evaluation. The goal was to explore the policy's ability to transfer across different simulated environments within the same task. For these experiments, we applied domain randomization during training: **I. Depth visual policy**, we added random noise to the depth maps and randomly zeroed some pixels. **II. RGB visual policy**, we applied random rotations, cropping, Gaussian blur, and color transformations to the input images.

All experiments were conducted using the Unitree Go1 robot, with physics simulation powered by MuJoCo.

### 5.2 Single-Scene Closed-Loop Training

We first test whether a visual policy can fit well in a single domain, to verify both the robot's learning ability and the effectiveness of our digital environment. We select three tasks in increasing order of difficulty: Hurdles (easy), Gaps (medium), and Stairs (hard). The policy is trained on one scene from each task and evaluated on all other scenes within the same task. To ensure variation, random noise is added to the trajectory, making the evaluation trails different from the training ones. As shown in Fig. 9, the policy performs well on the training scene but generalizes poorly on unseen scenes, which matches our expectation as the training domain is limited. One interesting finding is that performance slightly improves on some unseen scenes that share similar visual characteristics (e.g., both being outdoor environments) with the training domain, as observed in the bottom-right corner of each confusion matrix in Fig. 9.

### 5.3 Multi-Scene Closed-Loop Training

Given that policies fit well in a single domain, we further investigate whether our scenes support effective closed-loop training for visual policies. We evaluated on both training set and evaluation set, the performance gap between training and evaluation sets is large (about 50% on average) for the Stairs task Fig. 10-(A), but relatively small (about 10% on average) for the Gaps task Fig. 10-(B). This suggests that the trained visual policies exhibit limited generalization on our benchmark, particularly for more challenging tasks.

**Ablation on Observation Types:** The above experiments do not use include those cones as observation. We further investigate how different observation types affect policy performance. (1) RGB vs.

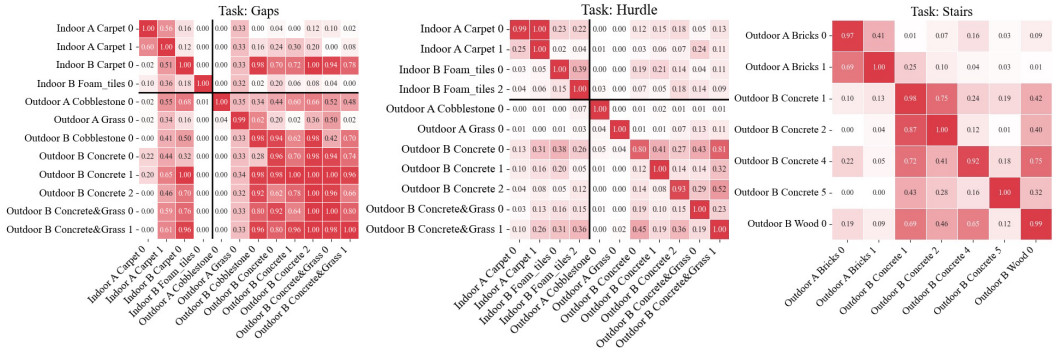
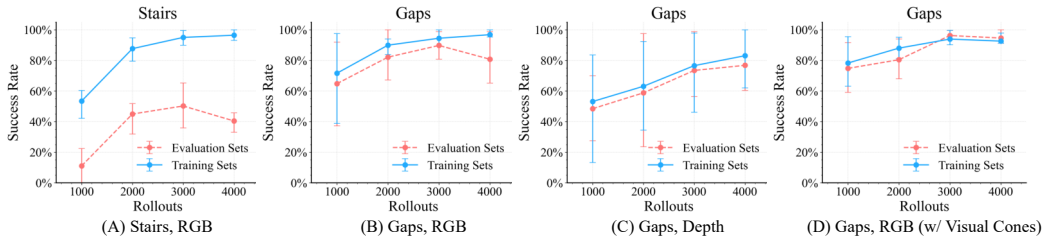



Figure 9: **Generalization of Single-Scene Policies.** Each policy (row) is trained in a single environment. The cross-scene generalization shows clear clustering.




(A) Stairs, RGB (B) Gaps, RGB (C) Gaps, Depth (D) Gaps, RGB (w/ Visual Cones)

Figure 10: **Results on Multi-Scene Closed-Loop Training.** We split each task's scenes into 70% for training and 30% for evaluation. For each scene, we perform 50 rollouts and report the average success rate over all rollouts in the train and evaluation sets. See the supplementary material for visual references of the listed scene names.

Depth: As shown in Fig. 10-(B) and (C), both inputs are trained with domain randomization. While RGB yields moderate results, Depth performs poorly even on the training set, suggesting that depth represented in our benchmark is currently less effective for learning. (2) With vs. without visual cones: Comparing Fig. 10-(C) and (D), adding visual cones effectively improves training efficiency and overall performance on both training and evaluation sets. This highlights the benefit of consistent, explicit visual cues (e.g., cones) in aiding policy learning under diverse visual domains.

### 5.4 Evaluating Visual Parkour Policies

Neverwhere is designed to test robot policies before real-world deployment. We evaluate visual policy checkpoints trained by *Lucidsim* [31], analyzing their performance gap between simulation and real environments. Results show that Lucidsim achieves reasonable success rates, with some scenes exceeding 95% and most scenes above 50%. This aligns roughly with Lucidsim's reported results of 73.3% for hurdles and 100% for stairs. Note that the real-robot test environments differ from our benchmark, so performance differences are expected.

Table 1: Evaluating *Lucidsim* [31] with Our benchmark.

| Tasks | Scenes | Rollouts | Average | Highest | Median | Lowest |
|---|---|---|---|---|---|---|
| hurdle | 15 | 50 | 59.67% | 95.33% | 68.67% | 0.00% |
| stairs | 14 | 50 | 55.82% | 93.16% | 55.37% | 2.78% |

## 6 Conclusion

We proposed the Neverwhere benchmark suite along with a real-to-sim toolchain. Our goal is to provide the community with a practical tool for testing policies before real-world deployment, potentially as part of a continuous testing setup. This work aims to accelerate the development of scalable and efficient approaches for robot evaluation, as current robot policies are becoming increasingly capable while existing evaluation methods remain inefficient.

Although the Neverwhere toolchain was initially designed for our locomotion evaluation benchmark suite, its capability for creating contact-aware real-world digital twins is broadly applicable across various domains. This unified framework, built on freely accessible pipelines, is designed to support the real-to-sim-to-real research community.

**Limitations.** Although Neverwhere has provided over 60 diverse scenes, expanding the benchmark with additional diverse scenes requires further human intervention and effort, as the reconstructed 3D splats and meshes are not automatically aligned with real-world scale or the standard z-up orientation. This necessitates manual reorientation, rescaling, and task labeling before they can be used in robot simulations. Our future work will explore learning-based methods for automatic alignment and scene labeling of the 3D reconstructions.

# References


[1] Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Lucy Xiaoyang Shi, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. $\pi_0$: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024.

[2] Dan Cernea. OpenMVS: Multi-view stereo reconstruction library. 2020.

[3] Xuxin Cheng, Kexin Shi, Ananye Agarwal, and Deepak Pathak. Extreme parkour with legged robots. *arXiv preprint arXiv:2309.14341*, 2023.

[4] Nikita Chernyadev, Nicholas Backshall, Xiao Ma, Yunfan Lu, Younggyo Seo, and Stephen James. Bigym: A demo-driven mobile bi-manual manipulation benchmark. In *Proceedings of The 8th Conference on Robot Learning*, pages 4201–4217. PMLR, 2025.

[5] Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. In *Robotics: Science and Systems (RSS)*, 2023.

[6] Erwin Coumans and Yunfei Bai. Pybullet, a python module for physics simulation for games, robotics and machine learning. http://pybullet.org, 2016.

[7] Chelsea Finn, Xin Yu Tan, Yan Duan, Trevor Darrell, Sergey Levine, and Pieter Abbeel. Deep spatial autoencoders for visuomotor learning. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 512–519. IEEE, 2016.

[8] Jiayuan Gu, Fanbo Xiang, Xuanlin Li, Zhan Ling, Xiqiang Liu, Tongzhou Mu, Yihe Tang, Stone Tao, Xinyue Wei, Yunchao Yao, Xiaodi Yuan, Pengwei Xie, Zhiao Huang, Rui Chen, and Hao Su. Maniskill2: A unified benchmark for generalizable manipulation skills. In *International Conference on Learning Representations*, 2023.

[9] Binbin Huang, Zehao Yu, Anpei Chen, Andreas Geiger, and Shenghua Gao. 2d gaussian splatting for geometrically accurate radiance fields. *ACM Transactions on Graphics (TOG)*, 43(4):32:1–32:15, 2024.

[10] Stephen James, Zicong Ma, David Rovick Arrojo, and Andrew J. Davison. Rlbench: The robot learning benchmark & learning environment. *arXiv preprint arXiv:1909.12271*, 2019.

[11] Stephen James, Zicong Ma, David Rovick Arrojo, and Andrew J Davison. RLBench: The robot learning benchmark & learning environment. *IEEE Robot. Autom. Lett.*, 5(2):3019–3026, April 2020.

[12] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4), July 2023.

[13] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics (TOG)*, 42(4):1–14, 2023.

[14] Shakiba Kheradmand, Daniel Rebain, Gopal Sharma, Weiwei Sun, Yang-Che Tseng, Hossam Isack, Abhishek Kar, Andrea Tagliasacchi, and Kwang Moo Yi. 3d gaussian splatting as markov chain monte carlo. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. Spotlight Presentation.

[15] Vikash Kumar, Rutav Shah, Gaoyue Zhou, Vincent Moens, Vittorio Caggiano, Jay Vakil, Abhishek Gupta, and Aravind Rajeswaran. Robohive: A unified framework for robot learning. *arXiv preprint arXiv:2310.06828*, 2023.

[16] Chengshu Li, Ruohan Zhang, Josiah Wong, Cem Gokmen, Sanjana Srivastava, Roberto Martín-Martín, Chen Wang, Gabrael Levine, Michael Lingelbach, Jiankai Sun, Mona Anvari, Minjune Hwang, Manasi Sharma, Arman Aydin, Dhruva Bansal, Samuel Hunter, Kyu-Young Kim, Alan Lou, Caleb R. Matthews, Ivan Villa-Renteria, Jerry Huayang Tang, Claire Tang, Fei Xia, Silvio Savarese, Hyowon Gweon, Karen Liu, Jiajun Wu, and Li Fei-Fei. Behavior-1k: A benchmark for embodied ai with 1,000 everyday activities and realistic simulation. In *Proceedings of The 6th Conference on Robot Learning*, pages 80–93. PMLR, 2023.

[17] Tao Lu, Mulin Yu, Linning Xu, Yuanbo Xiangli, Limin Wang, Dahua Lin, and Bo Dai. Scaffold-gs: Structured 3d gaussians for view-adaptive rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20654–20664, 2024.

[18] Ajay Mandlekar, Soroush Nasiriany, Bowen Wen, Iretiayo Akinola, Yashraj Narang, Linxi Fan, Yuke Zhu, and Dieter Fox. MimicGen: A data generation system for scalable robot learning using human demonstrations. *arXiv [cs.RO]*, October 2023.

[19] Nicolas Moenne-Loccoz, Ashkan Mirzaei, Or Perel, Riccardo de Lutio, Janick Martinez Esturo, Gavriel State, Sanja Fidler, Nicholas Sharp, and Zan Gojcic. 3d gaussian ray tracing: Fast tracing of particle scenes. *ACM Transactions on Graphics (TOG)*, 43(6):232:1–232:19, 2024.

[20] Soroush Nasiriany, Abhiram Maddukuri, Lance Zhang, Adeet Parikh, Aaron Lo, Abhishek Joshi, Ajay Mandlekar, and Yuke Zhu. Robocasa: Large-scale simulation of everyday tasks for generalist robots. In *Robotics: Science and Systems*, 2024.

[21] Soroush Nasiriany, Abhiram Maddukuri, Lance Zhang, Adeet Parikh, Aaron Lo, Abhishek Joshi, Ajay Mandlekar, and Yuke Zhu. Robocasa: Large-scale simulation of everyday tasks for generalist robots. *arXiv preprint arXiv:2406.02523*, 2024.

[22] NVIDIA Corporation. Nvidia isaac sim. https://developer.nvidia.com/isaac/sim, 2025. Accessed: 2025-05-16.

[23] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(3), 2022.

[24] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[25] Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. Pixelwise view selection for unstructured multi-view stereo. In *European Conference on Computer Vision (ECCV)*, 2016.

[26] Stone Tao, Fanbo Xiang, Arth Shukla, Yuzhe Qin, Xander Hinrichsen, Xiaodi Yuan, Chen Bao, Xinsong Lin, Yulin Liu, Tse kai Chan, Yuan Gao, Xuanlin Li, Tongzhou Mu, Nan Xiao, Arnav Gurha, Zhiao Huang, Roberto Calandra, Rui Chen, Shan Luo, and Hao Su. Maniskill3: Gpu parallelized robotics simulation and rendering for generalizable embodied ai. *arXiv preprint arXiv:2410.00425*, 2024.

[27] Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5026–5033. IEEE, 2012.

[28] Yifan Wang, Kai Zhang, Yichong Wang, Jingyi Yu, and Yebin Zhang. Sugar: Surface-aligned gaussian splatting for accurate and fast radiance field rendering. *arXiv preprint arXiv:2312.14364*, 2023.

[29] Yufei Wang, Zhou Xian, Feng Chen, Tsun-Hsuan Wang, Yian Wang, Katerina Fragkiadaki, Zackory Erickson, David Held, and Chuang Gan. RoboGen: Towards Unleashing Infinite Data for Automated Robot Learning via Generative Simulation. *arXiv [cs.RO]*, November 2023.

[30] Ze Yang, Yun Chen, Jingkang Wang, Sivabalan Manivasagam, Wei-Chiu Ma, Anqi Joyce Yang, and Raquel Urtasun. Unisim: A neural closed-loop sensor simulator. In *CVPR*, 2023.

[31] Alan Yu, Ge Yang, Ran Choi, Yajvan Ravan, John Leonard, and Phillip Isola. Lucidsim: Learning agile visual locomotion from generated images. In *8th Annual Conference on Robot Learning*, 2024.

[32] Kai Zhang, Yifan Wang, Yichong Wang, Jingyi Yu, and Yebin Zhang. 3dgut: Enabling distorted cameras and secondary rays in gaussian splatting. *arXiv preprint arXiv:2403.12345*, 2024.

[33] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models.

[34] Bowen Zhao, Kelvin Wang, Yunzhu Zhang, Xinyang Wang, Yunzhu Li, Joshua B. Tenenbaum, Antonio Torralba, and Jiajun Wu. Aloha unleashed: A simple recipe for robot dexterity. In *Proceedings of the 8th Conference on Robot Learning (CoRL)*, 2024.

[35] Kun Zhou, Qiming Hou, Rui Wang, and Baining Guo. Gpu-based ray tracing of splats. In *Proceedings of the 2010 18th Pacific Conference on Computer Graphics and Applications*, pages 147–154. IEEE, 2010.

# NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract states the goal of developing a scalable way to evaluate quadruped visual locomotion in simulation before real-world deployment and introduces the Neverwhere benchmark suite with over fifty high-fidelity scenes. The introduction reiterates this aim, presenting Neverwhere as a collection of digital twins for testing visuomotor policies and a toolchain for creating these environments.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The paper mentions limitations in the abstract, stating an intent to "highlight the limitations of treating 3D Gaussian as the sole data source for training". It also discusses challenges in the pipeline, such as inaccurate collision geometry for certain materials and handling varying material properties within a single mesh representation. The experiments section also notes that trained visual policies exhibited "limited generalization on our benchmark, particularly for more challenging tasks".

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best

judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: The paper details its methodological approach, including the formulation of the loss function used for 3D Gaussian training with depth supervision (Equation 1 in Section 4). The components and rationale for this approach and the overall Neverwhere toolchain are described, outlining the conceptual basis for its construction and operation.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The paper details the tasks, reinforcement learning setup (action space, observation space, privileged observation), rendering wrappers used, and the training setup for experiments (teacher-student approach, DAgger, policy architecture, single-scene and multi-scene training configurations, domain randomization techniques). It also specifies the robot model and physics simulator (MuJoCo).

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.

(b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.

(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The paper states its intention to "release the Neverwhere benchmark suite, which comprises over 60 high-quality, ready-to-use scenes" and to provide "visual parkour policy checkpoints". It also mentions "We are committed to sharing the toolchain together with the benchmark suite".

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: The paper describes the experimental setup including data splits for multi-scene training (70% for training, 30% for evaluation), the use of DAgger with 1,000 trajectories per iteration, and the types of domain randomization applied. It also mentions the policy architecture used (Action Chunking Transformers for student policies) and the robot and simulator used. Specifics on hyperparameters like learning rates or optimizer details are stated in code.

   Guidelines:

   - The answer NA means that the paper does not include experiments.

- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: The paper reports average success rates from multiple rollouts, which evaluates Lucidsim, further provides median, highest, and lowest success rates, offering information about the distribution and variability of these results across different scenes.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The paper specify the type of compute workers, memory, or time of execution for the experiments conducted in appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

Justification: The authors have reviewed the NeurIPS Code of Ethics.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [NA]

Justification: The paper focuses on the technical contributions of the benchmark suite for evaluating visual parkour policies and does not contain a dedicated section addressing potential positive or negative societal impacts.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: The paper describes a benchmark suite based on 3D scans of university campuses and policy checkpoints for robot locomotion. While it involves real-world data, it doesn't seem to fall into the high-risk categories. The data is for robotic environment simulation.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.

- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The paper cites various existing works and tools it builds upon or uses, such as COLMAP, OpenMVS, MuJoCo, gsplat, Vuer, and references research papers for policy architectures and datasets.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The paper describes the new assets: the Neverwhere benchmark suite (over 50-60 scenes), the data collection toolchain, and visual parkour policy checkpoints. The paper details the scene construction process, task definitions, and the nature of the scenes (digital replicas of urban indoor and outdoor environments). The intention to release these assets with the toolchain implies documentation will be provided.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

Justification: The research described in the paper involves creating digital twins of environments and training robot policies in simulation. It does not appear to involve crowdsourcing experiments or research with human subjects as participants in studies.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: The paper does not describe research involving human subjects as study participants, so IRB approval or discussion of participant risks is not applicable.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: The paper focuses on visual parkour benchmarks, 3D scene reconstruction, and reinforcement learning for locomotion policies. There is no mention of Large Language Models (LLMs) being used as an important, original, or non-standard component of the core methods.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (https://neurips.cc/Conferences/2025/LLM) for what should or should not be described.