Learning Visual Parkour from Generated Images

Anonymous Author(s)
Affiliation
Address
`email`

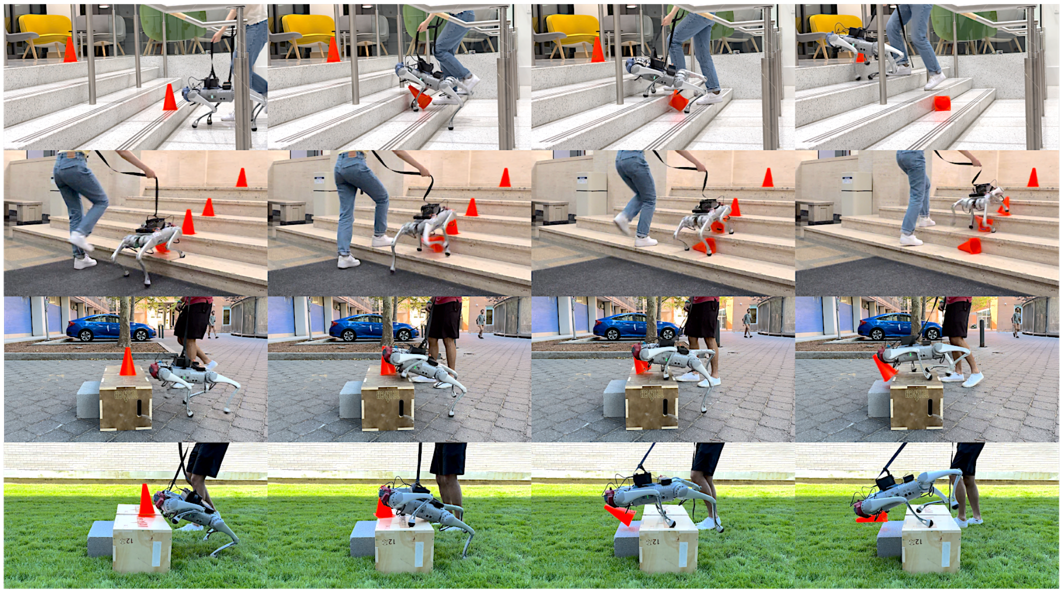


Figure 1: **Learning Visual Parkour from Generated Images.** Top to bottom: (1,2) robot climbing stairs. (3) robot climbing hurdles on a stone ground (4) on a grassy courtyard. Notice the different box color.

Abstract: Fast and accurate physics simulation is an essential component of a modern, learning-based approach to robotics, where robots can explore unsafe scenarios that would otherwise be infeasible in the real world. Yet, it remains difficult to incorporate perception into the sim-to-real pipeline to match the real world in its diversity and richness. This work uses visual parkour on a quadruped robot as a challenging testbed. We demonstrate that robots can learn to scale tall obstacles with precise eye-body coordination purely from generated images. We provide comprehensive empirical validation of the robustness of the resulting visual policy both in the real world and via a collection of high-fidelity digital replicas of scenes captured in the wild. Our result shows that a visual policy trained purely from generated images in LucidSim is robust enough to transfer directly to the real world using an off-the-shelf webcam. Website: https://lucidsim.github.io/

1 Introduction

What does it take to build an autonomous robot that can operate alongside us in a dynamic and open environment, such as a busy city street? Consider a small quadruped robot carrying goods across an intersection – it must understand traffic signals, recognize and avoid colliding with pedestrians, and possess the ability to jump onto a tall curb when it reaches the other side of the street. Central to this picture is the robot's ability to perceive and understand the world around it for a variety of purposes.

Submitted to the 8th Conference on Robot Learning (CoRL 2024). Do not distribute.

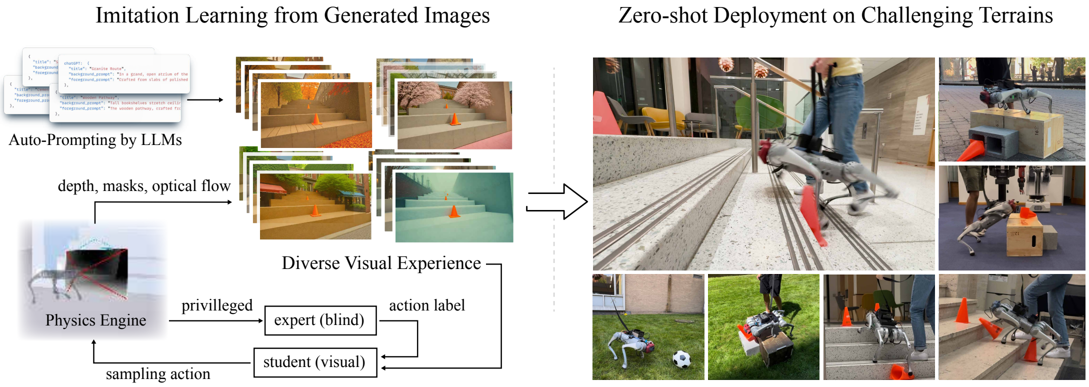

Imitation Learning from Generated Images
Zero-shot Deployment on Challenging Terrains
Auto-Prompting by LLMs
depth, masks, optical flow
Diverse Visual Experience
Physics Engine
privilleged
expert (blind)
action label
student (visual)
sampling action

Figure 2: **LucidSim.** Left: Imitation learning from generated images. We collect a large number of diverse, structured image prompts from an LLM, that is combined with the depth map and semantic masks to produce diverse visual data. The student is iteratively improved via DAgger using its own on-policy samples. Right: The resulting policy is sufficiently robust to be deployed in a variety of challenging terrains, including obstacles that are comparable in size to the robot's body height.

It also needs to be robust enough to handle the open world, which, unlike controlled laboratory settings, is full of unfamiliar encounters that demand an appropriate response.

Existing efforts in robot learning have approached building such systems from two primary directions. The first, prevalent in robot manipulation domains, focuses on imitating human demonstrations collected in the real world, operating under the assumption that scaling up data collection will eventually encompass sufficiently diverse scenarios to produce a versatile policy [1, 2, 3, 4]. However, relying on real-world data collection has limitations, as it is impractical to cover unsafe scenarios or tasks that are not feasible for human teleoperation. The second approach involves learning in simulation and then transferring the knowledge to the real world. This method has achieved impressive results, demonstrating high levels of dexterity [5, 6], agility [7, 8, 9, 10], and robustness in real-world environments [11, 12], while allowing environment designers full control over even the smallest details.

With great power, comes the great burden of having to specify everything. At the heart of the problem is how to produce diverse visual data without the unwanted burden, while simultaneously retaining control. The goal of this project is to reconsider visuomotor learning via sim-to-real in this context, and explore ways to learn from generated data. We choose the task of visual parkour as our testbed, where a small quadruped robot must scale obstacles comparable in height to its own body with precise eye-body coordination [10, 9, 13]. Figure 2 illustrates our approach. We begin with a low-poly terrain geometry in a physics simulation engine. We then unroll an expert policy that has been trained in simulation but cannot be deployed in the real world due to its reliance on privileged access to the height map. Using the rendered depth map from the robot's egocentric view and accompanying semantic masks for parts of the scene that we intend to control, we can shape the material properties, weather, and cultural details while maintaining tight alignment with the terrain geometry. Knowledge of the terrain geometry is key to visual parkour and can be difficult to extract from a single RGB camera view. We generate stylistically consistent stacks of image observations by warping the initial frame using dense optical flow, so that the robot can infer important knowledge of the terrain through the natural movement of its head-mounted camera.

Complementary to our image generation pipeline is a scalable way to source diverse prompts from a language model. We offer details on our meta prompt strategy and our way to scale it up to thousands of prompts (see Sec. 3.1). Finally, the ability to simulate and collect on-policy data, backed by a scalable systems implementation that distributes rendering and trajectory unroll over many GPUs, enabled us to run Dataset Aggregation (DAgger). We show that this greatly improves the robustness of the resulting visual parkour policy over baselines trained on teacher trajectories alone.

Our contributions are three fold: First, a technique for producing geometrically and dynamically correct, multi-frame image stacks for robot parkour. Second, a technique to produce diverse and complex imagery, by sourcing a large number of detailed, structured image prompts from an LLM, that is quite steerable in practice. Finally, we provide the first empirical demonstration of an agile visual parkour policy that is trained purely on generated data, that out-performs domain randomization baselines in the real-world.

2 Problem Formulation

This work concerns the scenario where we have partial knowledge of the target environment $\mathcal{D}$ in which our legged robot will be deployed, and we want to construct a generative learning environment $\mathcal{G}$ such that it offers a similar experience as what the robot will encounter in $\mathcal{D}$. We assume a sim-to-real setup, where deployment occurs in the real world without additional training.

Prior-Assisted Domain Generation. Consider the target environment $\mathcal{D}$ as a Partially Observable Markov decision process given by the tuple $\mathcal{D} = \langle \mathcal{S}, \mathcal{A}, \mathcal{O} \rangle$, where $\mathcal{S}$ is the physical state of the environment, $\mathcal{A}$ is the action space, and $\mathcal{O}$ is the space of observations consisting of proprioceptive observations of the robot, o^p and the vision input o^v. We assume before training that we are given a rough description in text, ℓ, or in some cases, a reference image x of the target environment $\mathcal{D}$. Our goal is to use our limited knowledge to steer our generative learning environment $\mathcal{G}$ towards what our legged robots will experience when it is deployed. This problem is ill-defined because ℓ and x do not contain sufficient statistics of $\mathcal{D}$. Therefore, some type of *prior* knowledge *has* to appear in our construction of $\mathcal{G}$. We refer to this class of problems as Prior-Assisted Domain Generation (PADG) to explicitly acknowledge the role of such priors, and to distinguish our approach from prior work that does so implicitly.

Structured Video Generation with Geometric and Physical Guidance. By construction, we want to sample paired vision and proprioceptive observations o^v and o^p from the learning environment $\mathcal{G}$. Since a single image only offers partial observability to the geometry, we need to collect a sequence of images $o^v = [x_t, x_{t-1}, x_{t-2}, \ldots]$ that are consistent with the corresponding sequence of states $[s_t, s_{t-1}, s_{t-2}, \ldots]$ in the physics simulation. We make the simplified assumption that the only information we need to know about the scene is the rough collision geometry of the scene, g, and the semantic and physical properties associated with each part of it $\{c_i\}$, and we do not need detailed textures and lighting. We need the state of the robot at each timestep, $s_t = \langle \mathbf{p}, \dot{\mathbf{p}}, r, p, \dot{y} \rangle$, which consists of the joint poses $\mathbf{p}$ and its velocity $\dot{\mathbf{p}}$; roll r, pitch p, and the yaw rate $\dot{y}$. Our goal is to construct the sampling function $x_t \sim f(s_t, g, \{c_i\})$.

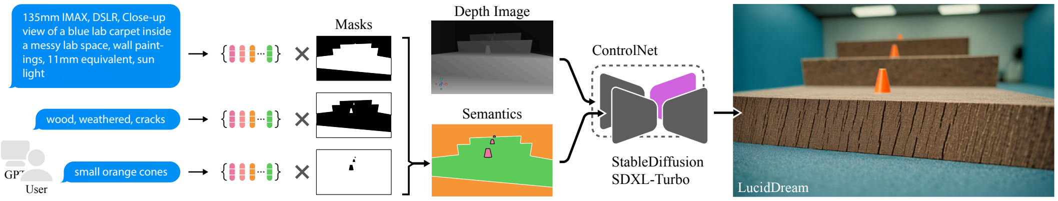
3 Learning Visual Parkour from Generated Images

We present our approach for learning visual parkour from generated images. Doing so involves solving four separate problems: First, how to align generated imagery with the simulated physics; second, how to extend a single generated image to multiple coherent frames; third, how to drive diversity using automatic prompting from an LLM; and finally, how to make the policy robust enough to deploy as-is in the real world with on-policy samples and teacher supervision.

3.1 Aligning Image Generation with Physics

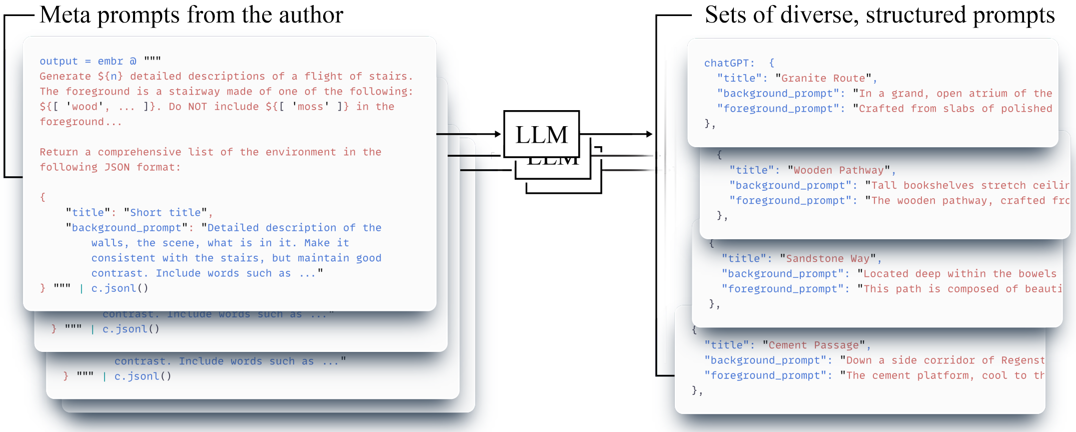
We augment a vanilla text-to-image model with additional semantic and geometric control. First, we replace the text prompt for the whole image with a set of prompt and semantic mask pairs that each specifies a part of the image (Fig.3). For instance, in the stairs scene, we specify the material and texture of the steps, plus a coarse silhouette for the mask. We avoid fine-grained semantic masking and let the model come up with those details. To make the images geometrically consistent, we take an off-the-shelf ControlNet that is trained on monocular depth estimates from MiDAS, and render inverse depth from the robot's perspective as input. The two conditioning images are fed into the diffusion and ControlNet model.

Generating diverse images via auto-prompting. Our early experiments showed that images prompted by GPT are often richer and more complex in composition than those prompted by hu-


135mm IMAX, DSLR, Close-up view of a blue lab carpet inside a messy lab space, wall paintings, 11mm equivalent, sun light
wood, weathered, cracks
small orange cones
GPT
User
Masks
Depth Image
Semantics
ControlNet
StableDiffusion
SDXL-Turbo
LucidDream

(a) Semantic Control (b) Conditioning via Depth (c) Generated View

Figure 3: **Image Generation Pipeline.** a) Using the CLIP embedding of the text prompt and the semantic mask rendered from the scene, we can semantically control the appearance of each object. b) Condition the image generation by the scene geometry using ControlNet. c) The generated images are consistent with the collision geometry, and contains a high level of detail.


Meta prompts from the author
LLM
Sets of diverse, structured prompts

```
output = embr @ """
Generate ${n} detailed descriptions of a flight of stairs.
The foreground is a stairway made of one of the following:
${[ 'wood', ... ]}. Do NOT include ${[ 'moss' ]} in the
foreground...

Return a comprehensive list of the environment in the
following JSON format:

{
    "title": "Short title",
    "background_prompt": "Detailed description of the
        walls, the scene, what is in it. Make it
        consistent with the stairs, but maintain good
        contrast. Include words such as ..."
} """ | c.jsonl()
```

```
chatGPT:  {
  "title": "Granite Route",
  "background_prompt": "In a grand, open atrium of the
  "foreground_prompt": "Crafted from slabs of polished
},
  {
    "title": "Wooden Pathway",
    "background_prompt": "Tall bookshelves stretch ceilir
    "foreground_prompt": "The wooden pathway, crafted frc
  },
 {
   "title": "Sandstone Way",
   "background_prompt": "Located deep within the bowels
   "foreground_prompt": "This path is composed of beauti
 },
{
  "title": "Cement Passage",
  "background_prompt": "Down a side corridor of Regenst
  "foreground_prompt": "The cement platform, cool to th
},
```

A collection of meta-templates + control-flow

Each produces *a collection* of structured prompts for image generation.

Figure 4: **Driving the diversity in image generation via automatic prompting from an LLM.** left: the author provide meta prompts that are used to solicit a large number of diverse, structured image prompts from the LLM. Note the use of a JSON format for the output that is then parsed. right: each meta prompt is used once to produce $20 \sim 30$ prompts. We limit the number to stay within the 4096 token limit. In total, each experiment involve anywhere from 600 to more than a thousand prompts.

mans. We also observe that diverse prompts produce diverse images, since images sampled from the same prompt can be degenerate in the overall theme. Figure 4 illustrates our strategy: we prompt chatGPT to generate batches of image prompts with a "meta" prompt that contains a title block, details of the request, and a final question asking for structured output in JSON. We can generate at max 30 prompts reliably in each query without exceeding the 4096 token limit of the OpenAI API, and can request images of a particular time of day, weather, and lighting conditions. Manually applying edits to those generated prompts is impractical. Instead, we tweak the meta prompt by rendering a small subset of GPT-generated prompts into images, and iterate until the generated prompts produce acceptable images. We then sample multiple batches of images prompts, each using a slightly modified meta prompt. We do not edit the generated image prompts manually.

3.2 Dreams In Motion: Video Generation via Image Warping

Inferring scene geometry from a single view is an ill-posed problem, but our robot is always on the move, so we can take advantage of this natural movement and infer geometry from a stack of camera views. Since state-of-the-art video generation models are not open-source, we developed Dreams In Motion (DIM), an alternative that takes advantage of our access to the scene geometry, to warp a single generated image into a coherent short video.

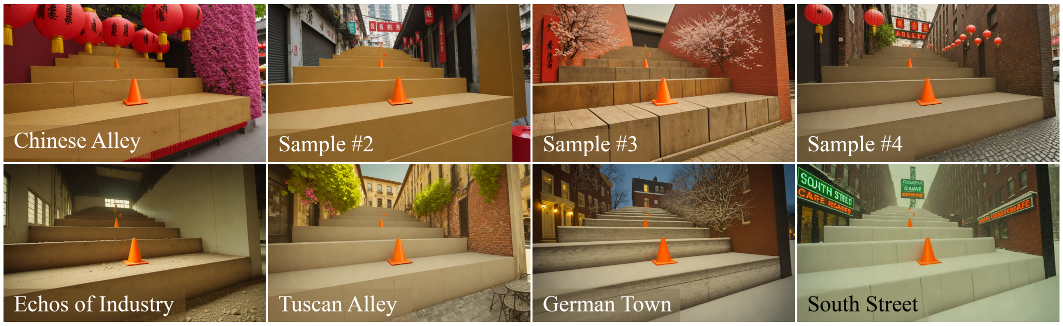


Figure 5: **LucidSim image samples from the stairs environment.** Each image is prompted with a different prompt sampled from chatGPT using a templated meta-prompt.

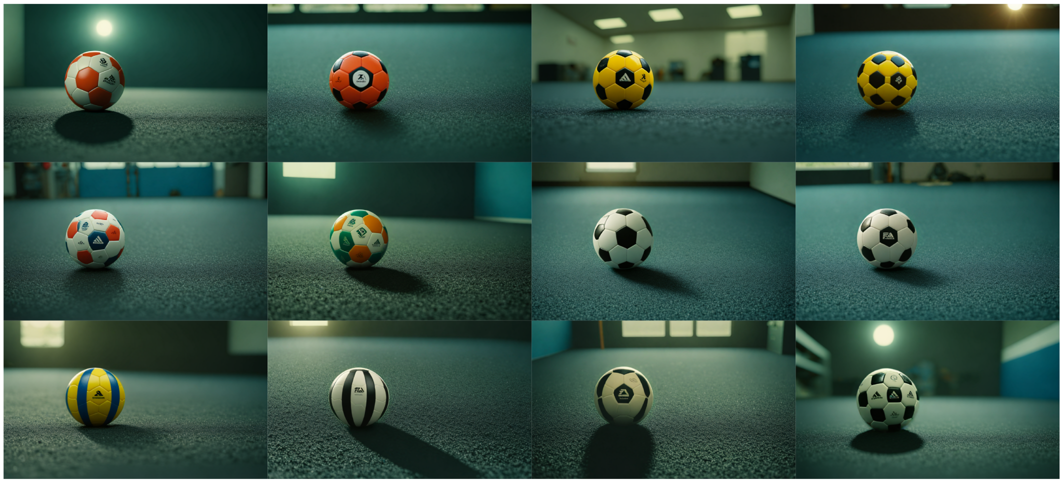


Figure 6: **Samples from a single prompt.** Some object descriptions can produce diverse visual results despite of its simple composition. We present multiple samples from the same text prompt: "Close-up view of a toy FIFA soccer ball, 135mm IMAX, very large." We include the generative workflow in the appendix.

DIM works as follows: first, we compute the ground-truth optical flow between the current ego view of the robot and that of the next time step using the terrain geometry (Figure 7b). Using this flow map, we can synthesize a proximal version of the next observation by warping the previous view. This way we can start with a single, generated 2D image, and create a coherent short video sequence. In practice, this also reduces the time spent on the image generation, making sampling faster. Quantitatively, this amounts to a speedup of almost 7 times (see Figure 8).

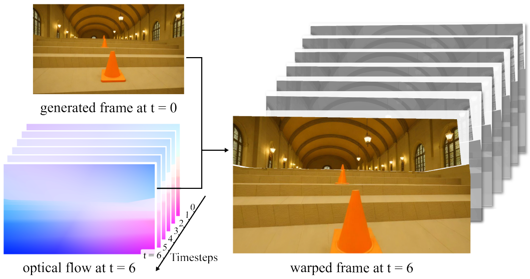


Figure 7: **Image Warping.** A single generated image is warped using ground-truth optical flow to provide the next k image observations.

3.3 Dataset Aggregation for Behavior Cloning

To collect trajectory data for behavior cloning, we begin by sampling intermediate checkpoints collected during the training of the privileged teacher. These are used to step the environment, with the teacher providing action labels. However, this initial dataset is insufficient for training a robust student that is capable of sampling on its own (see Figure 10). To improve the initial student, we perform three DAgger iterations, leveraging DIM to accelerate data collection.

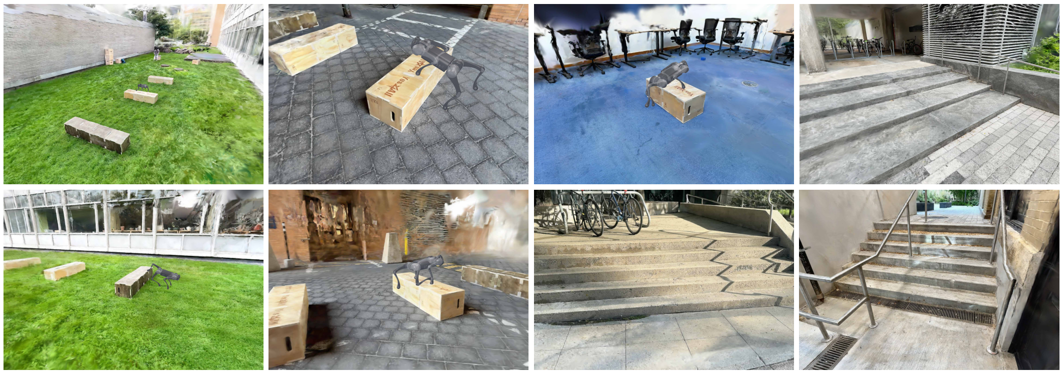


Figure 9: **Real-to-Sim Benchmark Environments.** Snapshots of a few environments we use for evaluation. Each scene is modeled using 3D Gaussian Splatting. The first-person view from the robots's perspective is highly photo-realistic.

4 Results

4.1 Simulated Evaluation

We construct a small set of benchmark environments using 3D Gaussian Splatting, a recent graphics technique that produces fast, complex, and photo-realistic digital replicas of static natural scenes. We provide performance statistics in these simulated benchmarks in four domains: tracking a soccer ball (**chase-soccer**); tracking an orange traffic cone (**chase-cone**), climbing over hurdles that are 75% of the robot's body height (**hurdle**); and traversing stairs featuring various material types (**stairs**). In chasing tasks, we randomly sample locations for the target objects within the view of the robot's camera frustum. For hurdle and stairs, waypoint locations are manually labeled and appear as orange traffic cones. Each task is evaluated in three replica scenes with 50 trials each, randomizing both the starting pose and waypoint location offsets. We report the fraction of goals reached (FGR) and forward displacement ($x_{\text{displacement}}$) toward each goal in Table 1 and 2.

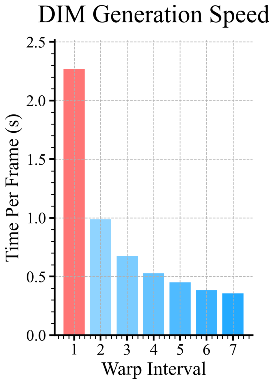


Figure 8: **DIM Accelerates Image Generation** Image warping requires minimal time for each frame. **Lower is better**.

We consider the following baselines: an expert policy that requires privileged terrain data as the oracle; a student policy trained to navigate using depth; a student policy trained using classical domain randomization over textures, and our method, LucidSim, trained with generated frame stacks using DIM.

Learning from Generated Images Out-Performs Domain Randomization We observe that LucidSim over performed classical domain randomization[14] in almost all evaluations. Surprisingly, we find that DR is able to climb stairs quite effectively in simulation, likely due to the repetitive gait that is induced after recognizing the first step. However, it struggles to perform on hurdles, where the timing of the jump is critical. We also observe a few factors affecting the performance of our oracle and depth baselines. The oracle struggles on one of the stairs environments (Marble) due to the presence of a railing, which it has never seen before in its privileged terrain information. However, because LucidSim is trained with behavior cloning on a simple terrain, it is not as adversely affected by such attributes in the testing environment. These challenges also affect the depth student, which is distracted by miscellaneous features in the benchmark environment (e.g. chairs, railings, walls).

4.2 Real World Results: Visual Parkour In The Wild

We deploy on a Unitree Go1 equipped with a budget RGB webcam, and run inference on the Jetson AGX Orin. Before deploying the policy, we analyze the camera latency and fine tune by applying

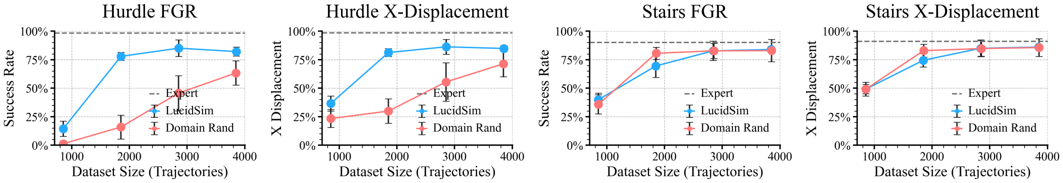


Figure 10: **Dagger Iterations Improve Policy Performance.** Each data point represents a new DAgger step. Increasing the number of DAgger iterations improves performance on the simulated benchmark environments. Evaluation include 50 unrolls on three environment instances for each task. Gray dotted line indicates the performance of the expert teacher.

Task	# of Trials	LucidSim	Domain Rand.
chase-cone	30	100.%	70.0%
chase-soccer	20	85.0%	35.0%
dark hurdle	15	86.7%	26.7%
light hurdles	15	73.3%	40.0%
stairs	10	100.%	50.0%

Figure 11: **Real-world Robot Results.** We measure the success rate of LucidSim and Domain Rand. student in a variety of real-world scenarios. Each task is evaluated over multiple environments, diverse in appearance.

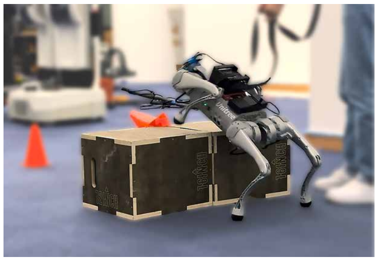


Figure 12: Robot climbing over a box that is on the same scale as its body height.

the measured delay to the existing dataset. Each task is evaluated on multiple scenes, and we record whether the robot reaches the target object (chase) or successfully traverses the obstacle.

We compare LucidSim to Domain Rand. and present the results in Figure 11. In the chasing tasks, we observe that Domain Rand. is able to identify color well (orange cones), but struggles with recognizing the patterns of the soccer ball. On the other hand, LucidSim is not only able to recognize the classic black and white soccer ball, but also generalizes to different colored soccer balls due to the rich diversity of the generated data it has seen before. For hurdles and stairs, Domain Rand. does not consistently recognize the obstacle in front of it, often resulting in a head-on collision, while LucidSim is able to consistently anticipate the obstacle and successfully traverse it.

4.3 Ablation: Image Generation without Conditioning

We present qualitative results on the effects of conditioning the image generation on depth. Without the depth map, the model failed to generate stairs. Instead, the image contains just flat ground. (see Fig. 13).

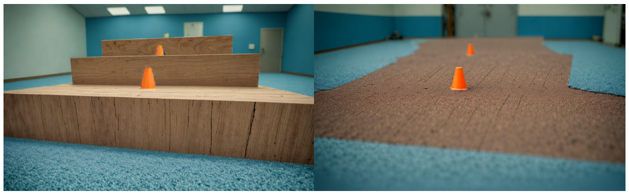


Figure 13: **Image Generation with and without conditioning on depth.** (left) With depth and open-text segmentation. (b-c) without depth, and segmentation alone.

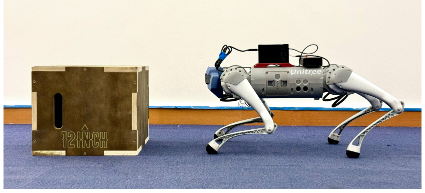


Figure 14: **Scale Reference.** 12 inch Hurdle in comparison to the robot 's body height.

5 Related Work

Robot parkour. Recent work in agile locomotion uses deep reinforcement learning and in-simulation behavior clone to achieve impressive levels of agility in quadrupeds [10, 9, 13] and humanoid robots [15]. These methods share the commonality that they all rely on depth images

Table 1: **Fraction of Goals Reached (FGR) In Simulated Benchmark Environments.**

		Chase-Cone			Chase-Soccer			Hurdle			Stairs		
Method	Obs. Space	Lawn	Lab	Urban	Lawn	Lab	Urban	Lawn	Lab	Urban	Bricks	Concrete	Marble
Oracle	state+terrain	98.6	96.2	97.9	98.6	96.2	97.9	95.8	100.0	99.0	97.0	100.0	73.4
Depth	depth	80.7	80.7	80.7	80.7	84.7	80.0	78.3	56.0	54.0	93.0	**86.0**	72.9
Domain Rand.	color	81.9	50.4	66.7	**97.3**	76.7	78.0	56.5	52.5	44.0	**95.5**	81.5	71.7
LucidSim	color	**96.7**	**84.0**	**98.0**	88.7	**90.7**	**94.7**	**84.8**	**79.5**	**76.5**	87.0	81.0	**83.7**

Table 2: **X-Displacement In Simulated Benchmark Environments.**

		Chase-Cone			Chase-Soccer			Hurdle			Stairs		
Method	Obs. Space	Lawn	Lab	Urban	Lawn	Lab	Urban	Lawn	Lab	Urban	Bricks	Concrete	Marble
Oracle	state+terrain	99.6	99.1	98.7	99.6	99.1	98.7	96.3	100.0	99.0	97.2	100.0	76.0
Depth	depth	95.8	**93.6**	93.8	95.0	92.9	92.9	80.7	70.4	59.1	93.5	**88.8**	76.5
Domain Rand.	color	91.6	81.2	84.9	**99.3**	89.2	89.5	66.6	61.6	57.1	**95.4**	85.1	76.5
LucidSim	color	**99.5**	92.7	**99.7**	92.3	**96.8**	**98.0**	**85.8**	**82.1**	**81.3**	88.8	**85.6**	**83.6**

as input. In contrast, our work does not depend on depth, and uses a low-cost, off-the-shelf webcam instead. To our best knowledge, this is the first reported result of visual robot parkour using RGB camera sensors, and the first that is trained completely in simulation with generated images.

Robot learning from demonstrations. Recent work in robot learning leverage low-cost hardware and expressive new policy classes borrowed from language modeling and image generation, to produce increasingly capable task planning and visuomotor controllers [1, 2, 3]. More recent work lowers this barrier-to-scale by removing the need for a robot arm retaining just the end effector itself [16, 17]. Data collection still involve setting up diverse scenes in the real-world [18, 19]. On the method side, LucidSim offers a reference implementation of a scalable generative learning environment for sampling diverse, on-policy visual data, thus bringing sim-to-real back to robot learning in the visual domain. On the capability side, we give legged robots the ability to see the world in full color. We eliminate our reliance on specialized depth cameras that fails deterministically under direct sunlight, against large reflective surfaces, and according to our experience, at night, when there are moving headlights of incoming traffic.

Real-to-sim and learning from digital twins. A simulated interactive environment is indispensable for generating counter-factual experiences that are either infeasible (like ego videos taken from a different camera pose) or too dangerous to attempt in the real world (such as driving against the traffic or into obstacles). Recent efforts in drone-racing [20], autonomous-driving [21], and humanoid soccer [22] took this approach to produce robust, but highly specialized controllers. In comparison, LucidSim takes a generative approach with the added benefit of being able to bias data according to demand. This work employs real-to-sim for evaluation and benchmarks, where targeted assessment via a small number of high-quality digital scans can be highly effective.

6 Conclusion

In this work, we discuss a scalable technique for producing geometrically and dynamically correct, multi-frame image stacks for robot learning. We also provide the first empirical demonstration of a visual parkour policy on a quadruped robot that is trained entirely using generated data. Although preliminary, we consider these results a promising proof-of-concept that points towards a more common-place usage of generative learning environments for difficult robotic tasks.

Limitations. The best strategy we found for curating the data distribution still involves human in the loop for feedback, although the incorporation of the assistance of an AI assisted, iterative curation procedure greatly reduces the amount of cognitive load on the experimenter. A second limitation is that we still rely on manually designed scene geometry, as we assume some prior knowledge of the test scene. Automating this aspect of the pipeline would be desirable for paving the way toward a complete generative learning environment for robots.

References

[1] C. Chi, S. Feng, Y. Du, Z. Xu, E. Cousineau, B. Burchfiel, and S. Song. Diffusion policy: Visuomotor policy learning via action diffusion. *arXiv preprint arXiv:2303.04137*, 2023.

[2] T. Z. Zhao, V. Kumar, S. Levine, and C. Finn. Learning Fine-Grained Bimanual Manipulation with Low-Cost Hardware. Apr. 2023.

[3] Z. Fu, T. Z. Zhao, and C. Finn. Mobile ALOHA: Learning Bimanual Mobile Manipulation with Low-Cost Whole-Body Teleoperation. Jan. 2024.

[4] P. Florence, C. Lynch, A. Zeng, O. Ramirez, A. Wahid, L. Downs, A. Wong, J. Lee, I. Mordatch, and J. Tompson. Implicit behavioral cloning. *Conference on Robot Learning (CoRL)*, 2021.

[5] T. Chen, J. Xu, and P. Agrawal. A system for general in-hand object re-orientation. In A. Faust, D. Hsu, and G. Neumann, editors, *Proceedings of the 5th Conference on Robot Learning*, volume 164 of *Proceedings of Machine Learning Research*, pages 297–307. PMLR, 08–11 Nov 2022. URL https://proceedings.mlr.press/v164/chen22a.html.

[6] Z.-H. Yin, B. Huang, Y. Qin, Q. Chen, and X. Wang. Rotating without seeing: Towards in-hand dexterity through touch. *Robotics: Science and Systems*, 2023.

[7] J. Tan, T. Zhang, E. Coumans, A. Iscen, Y. Bai, D. Hafner, S. Bohez, and V. Vanhoucke. Sim-to-real: Learning agile locomotion for quadruped robots. *ArXiv*, abs/1804.10332, 2018. URL https://api.semanticscholar.org/CorpusID:13750177.

[8] G. Margolis, G. Yang, K. Paigwar, T. Chen, and P. Agrawal. Rapid locomotion via reinforcement learning. In *Robotics: Science and Systems*, 2022.

[9] Z. Zhuang, Z. Fu, J. Wang, C. Atkeson, S. Schwertfeger, C. Finn, and H. Zhao. Robot parkour learning. In *Conference on Robot Learning (CoRL)*, 2023.

[10] X. Cheng, K. Shi, A. Agarwal, and D. Pathak. Extreme parkour with legged robots. *arXiv preprint arXiv:2309.14341*, 2023.

[11] A. Agarwal, A. Kumar, J. Malik, and D. Pathak. Legged locomotion in challenging terrains using egocentric vision. In *6th Annual Conference on Robot Learning*, 2022. URL https://openreview.net/forum?id=Re3NjSwf0WF.

[12] J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter. Learning quadrupedal locomotion over challenging terrain. *Science Robotics*, 5(47):eabc5986, 2020. doi:10.1126/scirobotics.abc5986. URL https://www.science.org/doi/abs/10.1126/scirobotics.abc5986.

[13] D. Hoeller, N. Rudin, D. V. Sako, and M. Hutter. Anymal parkour: Learning agile navigation for quadrupedal robots. *Science Robotics*, 9, 2023. URL https://api.semanticscholar.org/CorpusID:259261813.

[14] 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, 2017.

[15] Z. Zhuang, Z. Fu, J. Wang, C. Atkeson, S. Schwertfeger, C. Finn, and H. Zhao. Robot parkour learning. In *Conference on Robot Learning*, 2023. URL https://api.semanticscholar.org/CorpusID:261696935.

[16] C. Chi, Z. Xu, C. Pan, E. Cousineau, B. Burchfiel, S. Feng, R. Tedrake, and S. Song. Universal Manipulation Interface: In-The-Wild Robot Teaching Without In-The-Wild Robots. Feb. 2024.

[17] J. Wang, S. Dasari, M. K. Srirama, S. Tulsiani, and A. Gupta. Manipulate by seeing: Creating manipulation controllers from pre-trained representations. 2023.

[18] A. Khazatsky, K. Pertsch, S. Nair, A. Balakrishna, S. Dasari, S. Karamcheti, S. Nasiriany, M. K. Srirama, L. Y. Chen, K. Ellis, P. Fagan, J. Hejna, M. Itkina, M. Lepert, Y. Ma, P. T. Miller, J. Wu, S. Belkhale, S. Dass, H. Ha, A. Jain, A. Lee, Y. Lee, M. Memmel, S. Y. Park, I. Radosavovic, K. Wang, A. Zhan, K. Black, C. Chi, K. B. Hatch, S. Lin, J. Lu, J.-P. Mercat, A. Rehman, P. R. Sanketi, A. Sharma, C. B. Simpson, Q. U. Vng, H. Walke, B. Wulfe, T. Xiao, J. H. Yang, A. Yavary, T. Zhao, C. Agia, R. Baijal, M. G. Castro, D. L. Chen, Q. Chen, T. Chung, J. Drake, E. P. Foster, J. Gao, D. A. Herrera, M. Heo, K. Hsu, J. Hu, D. Jackson, C. Le, Y. Li, K. Lin, R. Lin, Z. Ma, A. Maddukuri, S. Mirchandani, D. Morton, T. Nguyen, A. O'Neill, R. M. Scalise, D. Seale, V. Son, S. Tian, E. Tran, A. E. Wang, Y. Wu, A. Xie, J. Yang, P. Yin, Y. Zhang, O. Bastani, G. Berseth, J. Bohg, K. Goldberg, A. Gupta, A. Gupta, D. Jayaraman, J. J. Lim, J. Malik, R. Mart'in-Mart'in, S. Ramamoorthy, D. Sadigh, S. Song, J. Wu, M. C. Yip, Y. Zhu, T. Kollar, S. Levine, and C. Finn. Droid: A large-scale in-the-wild robot manipulation dataset. *ArXiv*, abs/2403.12945, 2024. URL https://api.semanticscholar.org/CorpusID:268531351.

[19] O. X.-E. Collaboration, A. O'Neill, A. Rehman, A. Maddukuri, A. Gupta, A. Padalkar, A. Lee, A. Pooley, A. Gupta, A. Mandlekar, A. Jain, A. Tung, A. Bewley, A. Herzog, A. Irpan, A. Khazatsky, A. Rai, A. Gupta, A. Wang, A. Kolobov, A. Singh, A. Garg, A. Kembhavi, A. Xie, A. Brohan, A. Raffin, A. Sharma, A. Yavary, A. Jain, A. Balakrishna, A. Wahid, B. Burgess-Limerick, B. Kim, B. Schölkopf, B. Wulfe, B. Ichter, C. Lu, C. Xu, C. Le, C. Finn, C. Wang, C. Xu, C. Chi, C. Huang, C. Chan, C. Agia, C. Pan, C. Fu, C. Devin, D. Xu, D. Morton, D. Driess, D. Chen, D. Pathak, D. Shah, D. Büchler, D. Jayaraman, D. Kalashnikov, D. Sadigh, E. Johns, E. Foster, F. Liu, F. Ceola, F. Xia, F. Zhao, F. V. Frujeri, F. Stulp, G. Zhou, G. S. Sukhatme, G. Salhotra, G. Yan, G. Feng, G. Schiavi, G. Berseth, G. Kahn, G. Yang, G. Wang, H. Su, H.-S. Fang, H. Shi, H. Bao, H. B. Amor, H. I. Christensen, H. Furuta, H. Walke, H. Fang, H. Ha, I. Mordatch, I. Radosavovic, I. Leal, J. Liang, J. Abou-Chakra, J. Kim, J. Drake, J. Peters, J. Schneider, J. Hsu, J. Bohg, J. Bingham, J. Wu, J. Gao, J. Hu, J. Wu, J. Wu, J. Sun, J. Luo, J. Gu, J. Tan, J. Oh, J. Wu, J. Lu, J. Yang, J. Malik, J. Silvério, J. Hejna, J. Booher, J. Tompson, J. Yang, J. Salvador, J. J. Lim, J. Han, K. Wang, K. Rao, K. Pertsch, K. Hausman, K. Go, K. Gopalakrishnan, K. Goldberg, K. Byrne, K. Oslund, K. Kawaharazuka, K. Black, K. Lin, K. Zhang, K. Ehsani, K. Lekkala, K. Ellis, K. Rana, K. Srinivasan, K. Fang, K. P. Singh, K.-H. Zeng, K. Hatch, K. Hsu, L. Itti, L. Y. Chen, L. Pinto, L. Fei-Fei, L. Tan, L. J. Fan, L. Ott, L. Lee, L. Weihs, M. Chen, M. Lepert, M. Memmel, M. Tomizuka, M. Itkina, M. G. Castro, M. Spero, M. Du, M. Ahn, M. C. Yip, M. Zhang, M. Ding, M. Heo, M. K. Srirama, M. Sharma, M. J. Kim, N. Kanazawa, N. Hansen, N. Heess, N. J. Joshi, N. Suenderhauf, N. Liu, N. D. Palo, N. M. M. Shafiullah, O. Mees, O. Kroemer, O. Bastani, P. R. Sanketi, P. T. Miller, P. Yin, P. Wohlhart, P. Xu, P. D. Fagan, P. Mitrano, P. Sermanet, P. Abbeel, P. Sundaresan, Q. Chen, Q. Vuong, R. Rafailov, R. Tian, R. Doshi, R. Mart'in-Mart'in, R. Baijal, R. Scalise, R. Hendrix, R. Lin, R. Qian, R. Zhang, R. Mendonca, R. Shah, R. Hoque, R. Julian, S. Bustamante, S. Kirmani, S. Levine, S. Lin, S. Moore, S. Bahl, S. Dass, S. Sonawani, S. Song, S. Xu, S. Haldar, S. Karamcheti, S. Adebola, S. Guist, S. Nasiriany, S. Schaal, S. Welker, S. Tian, S. Ramamoorthy, S. Dasari, S. Belkhale, S. Park, S. Nair, S. Mirchandani, T. Osa, T. Gupta, T. Harada, T. Matsushima, T. Xiao, T. Kollar, T. Yu, T. Ding, T. Davchev, T. Z. Zhao, T. Armstrong, T. Darrell, T. Chung, V. Jain, V. Vanhoucke, W. Zhan, W. Zhou, W. Burgard, X. Chen, X. Chen, X. Wang, X. Zhu, X. Geng, X. Liu, X. Liangwei, X. Li, Y. Pang, Y. Lu, Y. J. Ma, Y. Kim, Y. Chebotar, Y. Zhou, Y. Zhu, Y. Wu, Y. Xu, Y. Wang, Y. Bisk, Y. Dou, Y. Cho, Y. Lee, Y. Cui, Y. Cao, Y.-H. Wu, Y. Tang, Y. Zhu, Y. Zhang, Y. Jiang, Y. Li, Y. Li, Y. Iwasawa, Y. Matsuo, Z. Ma, Z. Xu, Z. J. Cui, Z. Zhang, Z. Fu, and Z. Lin. Open X-Embodiment: Robotic learning datasets and RT-X models. https://arxiv.org/abs/2310.08864, 2023.

[20] M. Adamkiewicz, T. Chen, A. Caccavale, R. Gardner, P. Culbertson, J. Bohg, and M. Schwager. Vision-only robot navigation in a neural radiance world. *IEEE Robotics and Automation Letters*, PP:1–1, 2021. URL https://api.semanticscholar.org/CorpusID:238253331.

[21] Z. Yang, Y. Chen, J. Wang, S. Manivasagam, W.-C. Ma, A. J. Yang, and R. Urtasun. Unisim: A neural closed-loop sensor simulator. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1389–1399, 2023. URL https://api.semanticscholar.org/CorpusID:260438489.

[22] 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, 2022.

[23] G. Research. Pathways language model (palm): Scaling to 540 billion parameters for breakthrough performance, 2022. URL https://research.google/blog/pathways-language-model-palm-scaling-to-540-billion-parameters-for-breakthrough-
Accessed: 2023-10-10.

[24] N. M. Shazeer. Fast transformer decoding: One write-head is all you need. *ArXiv*, abs/1911.02150, 2019. URL https://api.semanticscholar.org/CorpusID:207880429.

[25] Comfyanonymous. Comfyui, 2023. URL https://github.com/comfyanonymous/ComfyUI. GitHub repository.

[26] Y. Zhu, J. Wong, A. Mandlekar, R. Martín-Martín, A. Joshi, S. Nasiriany, and Y. Zhu. robosuite: A modular simulation framework and benchmark for robot learning. In *arXiv preprint arXiv:2009.12293*, 2020.

[27] J. L. Schönberger, E. Zheng, M. Pollefeys, and J.-M. Frahm. Pixelwise view selection for unstructured multi-view stereo. In *European Conference on Computer Vision (ECCV)*, 2016.

[28] J. L. Schönberger and J.-M. Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[29] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4), July 2023. URL https://repo-sam.inria.fr/fungraph/3d-gaussian-splatting/.

[30] M. Turkulainen, X. Ren, I. Melekhov, O. Seiskari, E. Rahtu, and J. Kannala. Dn-splatter: Depth and normal priors for gaussian splatting and meshing, 2024.

[31] M. Tancik, E. Weber, E. Ng, R. Li, B. Yi, J. Kerr, T. Wang, A. Kristoffersen, J. Austin, K. Salahi, A. Ahuja, D. McAllister, and A. Kanazawa. Nerfstudio: A modular framework for neural radiance field development. In *ACM SIGGRAPH 2023 Conference Proceedings*, SIGGRAPH '23, 2023.