A Conclusion, Limitations, and Future Work

In this work, we introduced SMS, a generalizable framework for physically grounded robot planning. Our method generates interactive, object-aware scene models to optimize robot behaviors that produce complex, physically dynamic interactions—a capability we demonstrated in both billiards-inspired manipulation and quadrotor landing tasks. By combining the semantic knowledge and generalization ability of foundation models with the principled predictive power of physical simulation, SMS addresses the limitations of prior approaches: specialized methods often fail to generalize beyond narrow domains; learning-based methods struggle to account for novel or out-of-distribution physical behaviors; and approaches relying solely on foundation models sacrifice the precision required for tasks demanding exact physical outcomes.

In future work, we aim to expand these physical reasoning capabilities to a wider range of robots and environments, and identify several key directions for further development.

Towards Improved Scene Reconstruction: While Gaussian splatting provides high-quality reconstructions, our implementation requires 2 to 3.5 seconds per frame; with our experiments using 60 frames per scene, reconstruction takes several minutes. Moreover, as with all differentiable rendering methods, unobservable or heavily occluded regions cannot be accurately reconstructed. This proved especially challenging for the interfaces between stacked objects in our quadrotor experiments, which necessitated post-processing to ensure the interface geometries were reasonably modeled. Incorporating generative models for 3D scene completion, such as [65, 66, 67], could simultaneously address both challenges: reducing reconstruction time and leveraging learned priors to plausibly infer occluded or unseen regions.

Incorporating Feedback for Closed-Loop Planning: A current limitation of SMS is its open-loop design: we query material properties, optimize actions in simulation, and execute them without incorporating feedback. As a result, SMS cannot adapt to inaccurate attribute estimates or unforeseen changes during execution. Incorporating strategies for visual feedback and system identification [e.g., 34, 68, 32] could address these limitations by enabling online updates to the scene reconstruction and physical parameters based on observed outcomes.

Expanding Action Expressivity: In our implementation, SMS uses a gradient-free optimization strategy, which is effective for action spaces that can be compactly parameterized. However, scaling to more expressive or high-dimensional control behaviors would benefit from differentiable physics simulators that support efficient gradient-based optimization.[2] Furthermore, language-based task specification [70] can facilitate objective function formulation for more complex applications and extend the utility of this framework to non-expert users.

B Additional Details for Billiards Scenario

B.1 Scene Generation

For our billiards experiment, we generated 18 scenes with the assistance of a procedural generation tool. This tool was written to randomly place a cue ball, a target ball, and a goal position. In all scenes but the first, random objects are then sampled and scattered throughout the scene. After this step, elements of the scene were manually adjusted to ensure that the target ball could reasonably achieve the goal position. We show our full set of environments in Figure B.1.

B.2 Scene Imaging and Reconstruction

For each billiards scene, we acquire 60 RGBD images at a resolution of 1280×800 pixels using a simulated Orbbec Gemini 2 camera. The camera begins at a frontal viewpoint capturing the full workspace, then moves outward along a vertical plane that faces the workspace while sampling viewpoints. Upon reaching the surface of a sphere of 1-meter radius, centered at the manipulator base, the camera continues its trajectory along the hemisphere. The camera continuously repeats this

[2] At present, no differentiable physics simulator met our project requirements. While we attempted to use the differentiable capabilities of Nvidia Warp [69], we encountered numerical instability without prohibitively small time steps. Genesis [58] has announced upcoming support for differentiable rigid-body simulation, but this was not available at the time of writing.

path three times, with its height gradually increasing from 0.1 to 0.25 meters as it moves to ensure complete coverage of the scene objects. The camera is always pointed at the workspace center.

Object detection is performed on the first frame at native resolution to maximize detection accuracy with OWLv2 [6]. Detected bounding boxes are used to prompt the SAM 2 [7] segmentation model. Occasionally, predicted masks slightly over- or under-segment object boundaries by a few pixels. To address this, we identify the edges of each segmentation mask and consider all pixels within a 5-pixel margin. For these pixels, we use a distance-based majority vote with k-nearest neighbors in the backprojected point cloud to assign corrected class labels.

For computational efficiency, Gaussian splatting is applied to all RGBD and segmentation images after downsampling to half-resolution (640×400): color images are downsampled via cubic interpolation, and depth and segmentation images via nearest-neighbor interpolation.

To obtain material properties, we iteratively query OpenAI's GPT-4o VLM with an image of the workspace with the entity of interest annotated with a bounding box and a text prompt. We provide the prompt template on the following page.

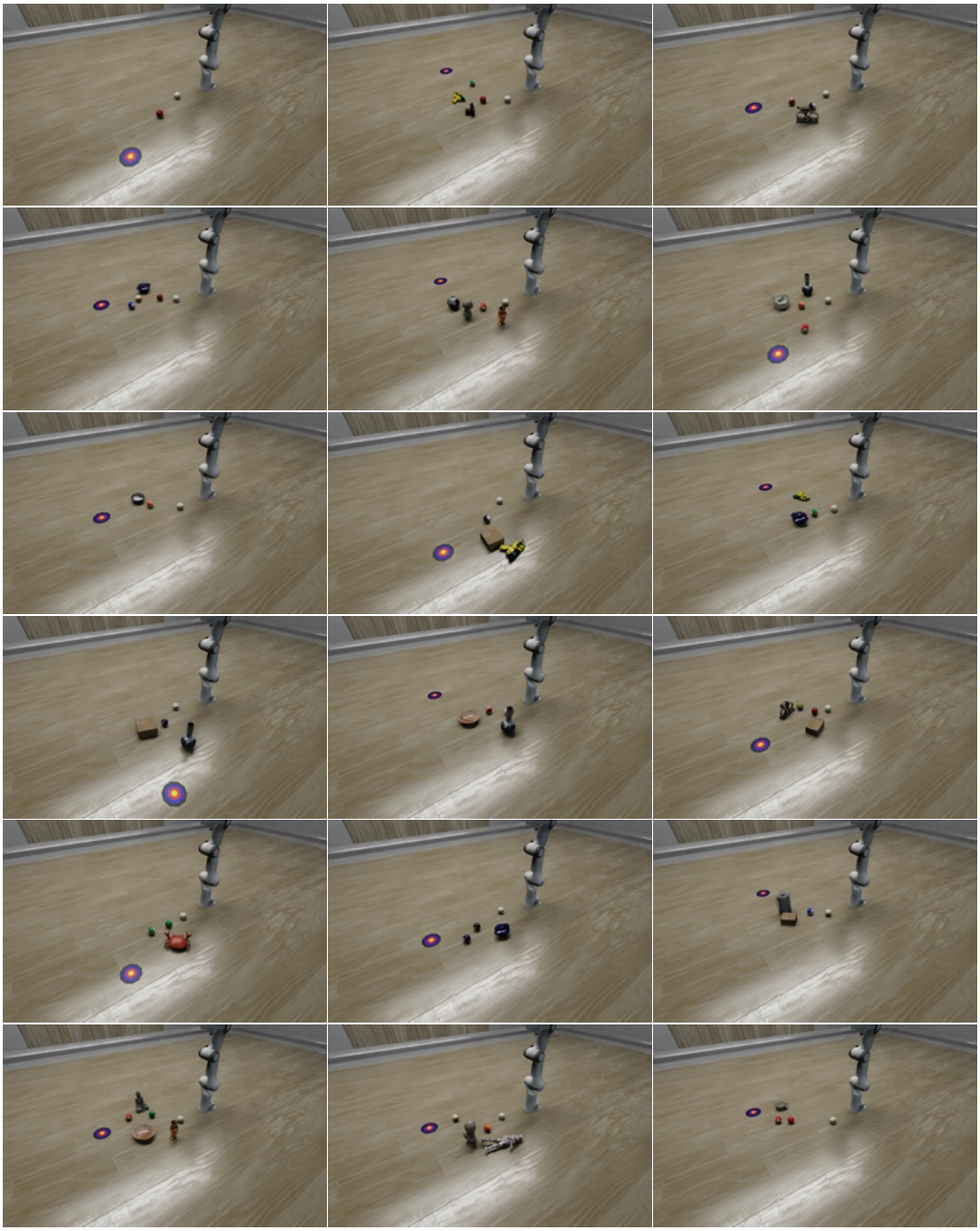

Figure B.1: Visualizations of the 18 billiards scenes evaluated in our experiments.

Material Query Prompt Template

System Message

```
You are an assistant to an autonomous robot.  Your job is
to interpret the robot's visual observations and answer
its questions.  The robot will provide its input query and
relevant context under "Input." It will provide more specific
requirements pertaining to the query under "Task." Please
provide your response in the specified format.
```

User Message

```
<Image>

Input:
The robot observes a scene and detects an object of interest.
The image is provided with a bounding box indicating the object
of interest.  The robot has a preliminary annotation of the
object's identity, and indicates it to be <annotation>, which
may or may not be correct.  The robot seeks verification of the
object's identity and physical property estimation.

Task:
Verify the identity of the object inside the bounding box.  If
the annotation is accurate, confirm it.  If it is inaccurate,
provide the most appropriate object label.  Furthermore,
provide a general description of the object's identity, physical
appearance, and purpose.

The robot requires an estimate of the object's physical
properties to calculate and plan for environment interactions.
First, determine the most appropriate material for the object.
If the object appears to be composed of multiple materials or
the material is indiscernible, please provide the most prevalent
or representative material.  Then, estimate the following
physical properties of the material:

Density (kg/m^3)
Friction Coefficient
Coefficient of Restitution
Young's Modulus (Pa)
Poisson's Ratio

Please provide a single best numerical estimate for each
physical property.  The output should be structured as a JSON
file, with the following fields:

-Annotation Accuracy
-Object Label
-General Description
-Material
-Density
-Friction Coefficient
-Coefficient of Restitution
-Young's Modulus
-Poisson's Ratio
```

B.3 Manipulator Control and Action Modeling

The FR3 manipulator is controlled with an operational space torque controller [71]. Each strike action is parameterized by the cue contact position, contact speed, and strike angle. The action is executed as a linear sweep comprising two 0.1-meter segments: an acceleration phase, where the end effector accelerates to the target speed at the contact point, and a deceleration phase, wherein it slows to rest after contact. This controller was used to compute torque commands at a rate of 400 Hz.

B.4 Virtual Environment

We use PyBullet [57] as our virtual physics environment and simulate dynamics with a time step of 0.0025 seconds. PyBullet only supports collision detection on convex meshes, therefore we perform a convex decomposition on our meshes using [72] before loading our reconstructed objects into the simulator. PyBullet simulations were run exclusively on the CPU.

B.5 Baseline Implementation Details

The baseline planner conducts an exhaustive grid search over candidate strike parameters at the cue ball contact point. Strike speed is discretized into 20 evenly spaced values ranging from 0.2 to 0.85 m/s, while strike angle is sampled at 60 increments from -10° to $+10^\circ$ relative to the vector connecting the centers of the cue and target balls. The planner assumes that, immediately after contact, the cue ball acquires the specified strike velocity and angle. For each candidate action, we analytically predict the immediate target ball post-collision state, modeling the cue–target ball collision as perfectly elastic. Specifically, we assume the normal component of the cue ball transfers to the target ball immediately after the collision.

To further account for secondary interactions with obstacles, we generate a top-down RGBD map of the workspace from the initial reconstruction by threshold-segmenting obstacles observed with a z coordinate greater than 0.005 meters to create a static occupancy mask. The simulated path of the target ball is traced through the workspace on this map. If a collision with an obstacle or workspace boundary is detected, the velocity of the target ball is reflected about the obstacle's normal direction, updating only its normal velocity component.

The grid search outputs the action predicted to bring the target ball closest to the goal according to this simplified physical model. This method does not update object states or account for subsequent changes in the environment during execution.

B.6 Supplementary Results

For completeness, an extended version of Figure 3 reporting predicted and realized performance results for each individual scene is also included as Figure B.2.

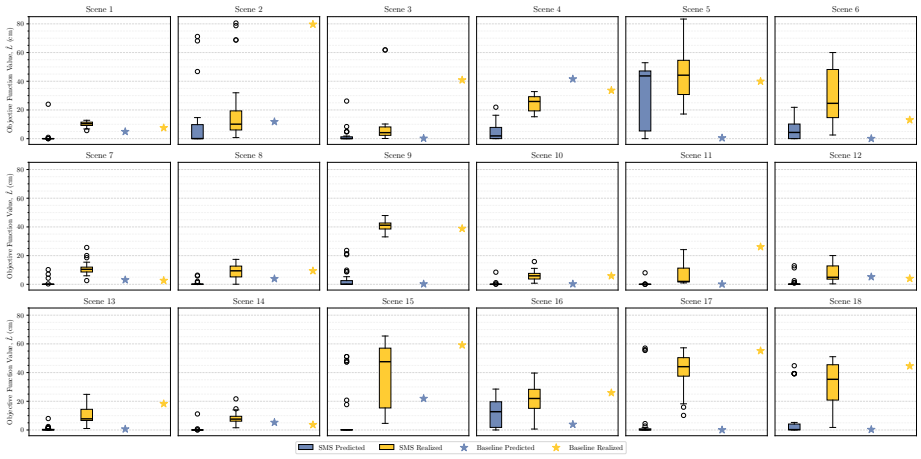

Figure B.2: Distributions of SMS performance over 30 repeated action optimizations. Baseline results are shown for comparison. Lower is better. For correspondence with Figure 3, Scenes A, B, C, and D respectively correspond to Scenes 10, 12, 16, and 5 here.

C Additional Details for Quadrotor Landing Scenario

C.1 Scene Generation

For the quadrotor landing scenario, we manually constructed four landing structures, each comprising a base and an overhanging landing platform. The base objects were chosen for their varied materials and geometries, while the platforms were selected as flat objects sufficiently large to accommodate the quadrotor. In each scene, the platform was positioned precariously with significant overhang. In two of the scenarios, we further cantilevered the platform by placing a heavy object as ballast, allowing us to evaluate landing behavior under different stability and load conditions. We show our full set of environments in Figure C.1.

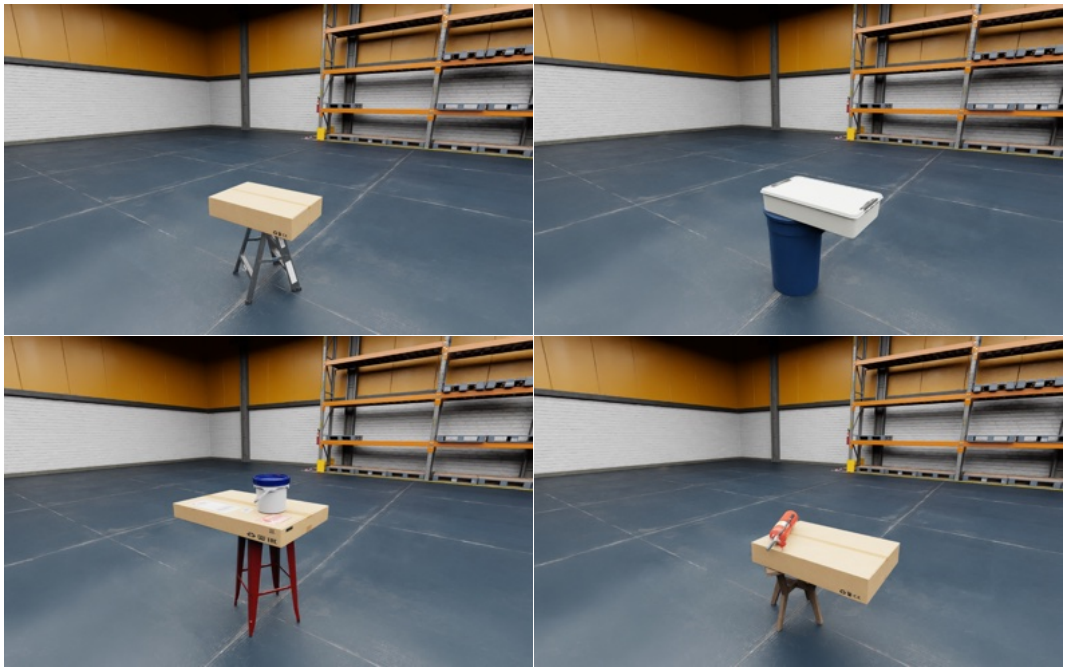

Figure C.1: Visualizations of the four quadrotor scenes evaluated in our experiments.

C.2 Scene Imaging and Reconstruction

Imaging and scene reconstruction were performed exactly as in Appendix B.1, except that in this scenario, the camera follows a helical path around the landing structure. The camera begins at a height of 0.35 meters and ascends to 1.35 meters, capturing 60 RGBD images over three orbits. Throughout, the camera remains focused on the center of the structure.

C.3 Quadrotor Propeller Model

To simulate realistic propeller downwash effects in the quadrotor landing experiments, we model each propeller as an ideal actuator disk using principles from classical momentum disk theory. This approach allows us to analytically estimate the induced airflow velocity and mass flow rate generated by each propeller, and to calibrate the smoothed-particle hydrodynamics (SPH) emitter parameters accordingly.

Momentum Disk Theory Overview: For a hovering quadrotor, each propeller must generate a thrust

$$F_{\text{prop}} = \frac{m_{\text{quadrotor}} \cdot g}{n_{\text{propellers}}},$$

where $m_{\text{quadrotor}}$ is the vehicle mass, g is gravitational acceleration, and $n_{\text{propellers}}$ is the number of propellers.

The propeller is modeled as an ideal disk of area $A_{\text{prop}} = \pi r_{\text{prop}}^2$. The actuator disk accelerates the air from a velocity v_0, typically approximated as 0 m/s for hovering or low-speed flight, up to v_{exit} far

after the propeller. According to momentum theory, the thrust is related to the disk-induced velocity by:

$$F_{\text{prop}} = \dot{m} \cdot (v_{\text{exit}} - v_0) = \rho_{\text{air}} A_{\text{prop}} v_{\text{prop}} (v_{\text{exit}} - v_0),$$

where $\dot{m}$ is the mass flow rate, ρ_{air} is the density of air, and v_{prop} is the average velocity through the disk.

We use Bernoulli's equation to relate the pressure and velocity before and after the propeller disk. The total pressure ahead of the disk is the sum of the static pressure, P_0, and dynamic pressure term, $0.5\rho_{\text{air}} v_0^2$, as

$$P_{0,\text{total}} = P_0 + 0.5\rho_{\text{air}} v_0^2,$$

while downstream of the disk the static pressure is

$$P_{\text{exit,total}} = P_0 + 0.5\rho_{\text{air}} v_{\text{exit}}^2.$$

This model predicts a pressure difference at the disk of

$$\Delta P = P_{\text{exit,total}} - P_{0,\text{total}}.$$

As such,

$$F_{\text{prop}} = \Delta P A_{\text{prop}} = 0.5\rho_{\text{air}} A_{\text{prop}} (v_{\text{exit}}^2 - v_0^2).$$

Equating the two equations for F_{prop} yields $v_{\text{prop}} = 0.5(v_{\text{ext}} + v_0)$.

Thus, for hovering ($v_0 = 0$), the induced velocity through the disk is $v_{\text{prop}} = 0.5(v_0 + v_{\text{exit}}) = 0.5 v_{\text{exit}}$, with $v_0 = 0$, and

$$v_{\text{exit}} = \sqrt{\frac{F_{\text{prop}}}{0.5\rho_{\text{air}} A_{\text{prop}}}}.$$

Mass Flow Rate and Particle Emitter Calibration: The mass flow rate through the propeller is

$$\dot{m} = \rho_{\text{air}} A_{\text{prop}} v_{\text{prop}}.$$

To represent this in the SPH simulation, each emitter produces $n_{\text{particles}}$ cylindrical fluid particles per time step Δt, each with length ℓ_{particle} and diameter d_{particle}. The particle volume is

$$V_{\text{particle}} = \pi \left(\frac{d_{\text{particle}}}{2} \right)^2 \ell_{\text{particle}}.$$

The required effective density for each simulated particle to match the physical mass flow is then

$$\rho_{\text{particle}} = \frac{\dot{m} \Delta t}{n_{\text{particles}} V_{\text{particle}}}.$$

where Δt is the simulation time step.

Each emitter in the simulation is thus configured with particles emitted downward ($-z$ direction), using v_{prop} as the particle velocity and ρ_{particle} as the particle density, so that the total airflow and momentum closely match the theoretical estimate from the real propeller.

In our simulation, we use $m_{\text{quadrotor}} = 1.182$ kg, $r_{\text{prop}} = 0.0685$, $\rho_{\text{air}} = 1.225$ kg/m^3, and, of course, $n_{\text{propellers}} = 4$. Other particle parameters such as $n_{\text{particles}}$, $\ell_{\text{particles}}$, and $d_{\text{particles}}$ are adaptively determined by the simulator as a function of time step and emission velocity. We show an image of the quadrotor's virtual environment with simulated propeller fluid particle emissions in Figure C.2.

C.4 Virtual Environment

Genesis [58] is used as the virtual physics environment, with dynamics simulated at a 0.01-second time step. To enable efficient fluid particle simulation, all computations are performed on the GPU.

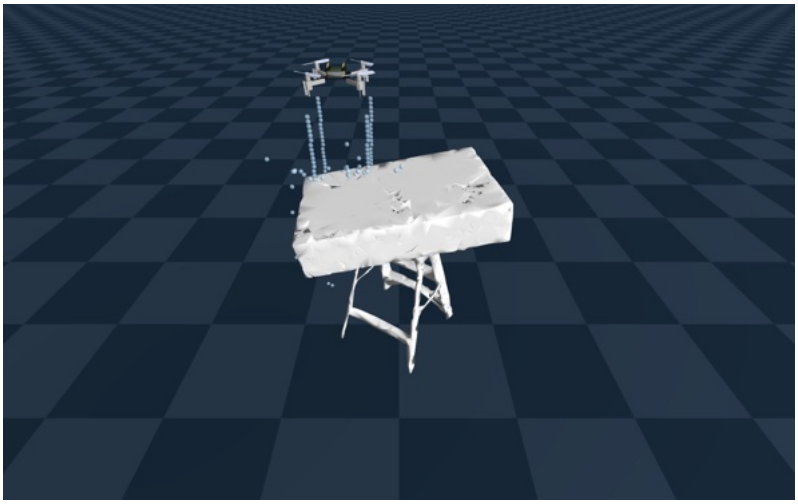

Figure C.2: Simulation of propeller downwash in Genesis [58], with fluid particles modeling the airflow from each propeller and its impact on the environment, as illustrated by the tilting box.

C.5 Baseline Implementation Details

For the quadrotor baseline, we derive candidate landing sites and approach trajectories using a combination of geometric analysis and visual prompting. Starting with the quadrotor's RGBD observation, we extract a point cloud and identify surface points with an approximately vertical normal and a height of at least 0.5 meters above the ground, which filters out ground-level points. Up to 15 non-overlapping candidate landing sites are then sampled from these points, ensuring spatial diversity; each site is marked as a circle on the image and assigned a unique identifier. The annotated image is provided as input to GPT-4o, which is prompted to select the most appropriate landing site. Next, we generate 9 candidate approach paths to the selected site, parameterized as Bezier curves and spanning a range of directions. Each path is annotated on the image with its corresponding identifier. GPT-4o is queried again to select the preferred trajectory, yielding the final approach path and landing site for the baseline. An example of these annotated images along with the selected results are shown in Figure C.3. Prompt templates are also provided in the following pages.

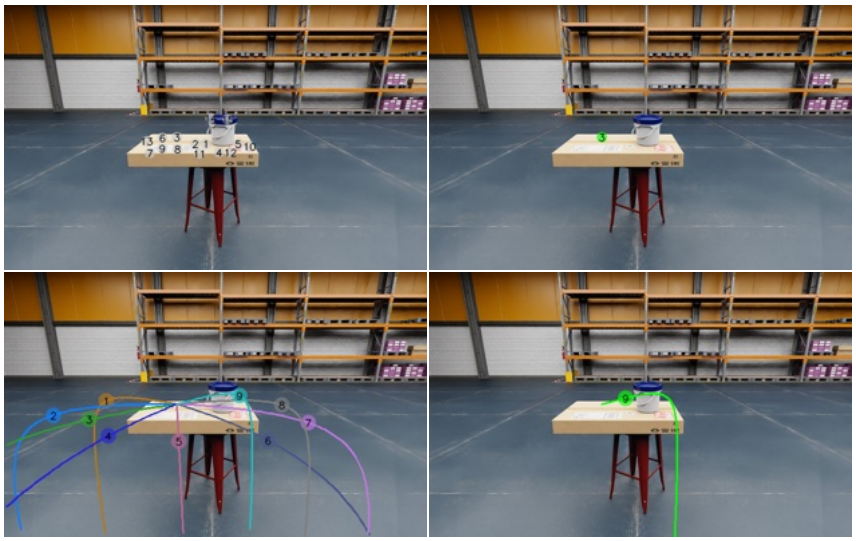

Figure C.3: Baseline landing site and approach selection. Top left: Candidate landing sites identified by surface geometry and height. Top right: Landing site selected by GPT-4o. Bottom left: Candidate approach paths to the selected site. Bottom right: Final approach chosen by GPT-4o.

Landing Position Query Prompt Template

System Message

```
You are an assistant to an autonomous robot.  Your job is
to interpret the robot's visual observations and answer
its questions.  The robot will provide its input query and
relevant context under "Input." It will provide more specific
requirements pertaining to the query under "Task." Please
provide your response in the specified format.
```

User Message

```
<Image>

Input:

The robot is a quadrotor looking to land on the
<landing_target>.  It needs to land on a stable position that
will neither cause it to fall nor topple the landing platform or
nearby objects.

Task:

An image of the quadrotor's current observation is provided.
Landing position candidates are provided and are annotated on
the image as circles with IDs.  Which of these landing sites
should the quadrotor choose as its landing position?

Please provide the output in the following format:

Reasoning:  (e.g., What should the quadrotor consider?  What are
the risks?  What are the safe areas?)
Decision:  (ID of landing location; Please only specify the ID
number)
```

Approach Path Query Prompt Template

System Message

```
You are an assistant to an autonomous robot.  Your job is
to interpret the robot's visual observations and answer
its questions.  The robot will provide its input query and
relevant context under "Input." It will provide more specific
requirements pertaining to the query under "Task." Please
provide your response in the specified format.
```

User Message

```
<Image>

Input:

The robot is a quadrotor looking to land on the
<landing_target>.  It needs to land on a stable position that
will neither cause it to fall nor topple the landing platform or
nearby objects.  It has identified a landing position and is now
attempting to determine the best approach path.  The approach
path must account for the quadrotor propeller wash, which can
impart a force on the objects below it, including the landing
platform (i.e., the <landing_target>).  The quadrotor should
seek a path that minimally disturbs objects that it will fly
over, and should especially try to avoid toppling the landing
area.

Task:

An image of the quadrotor's current observation is provided.
Approach path candidates are annotated as curves of different
colors, each with a corresponding numerical ID. Note that these
paths are projected to the height of the landing platform and
the quadrotor would be flying at some small distance above these
paths until it arrives at the landing position where it will
descend.  Which of these approach paths should the quadrotor
take?

Please provide the output in the following format:

Reasoning:  (e.g., What should the quadrotor consider?  What are
the risks?)
Decision:  (ID of approach path; Please only specify the ID
number)
```

References

[1] N. Fazeli, M. Oller, J. Wu, Z. Wu, J. B. Tenenbaum, and A. Rodriguez. See, feel, act: Hierarchical learning for complex manipulation skills with multisensory fusion. *Science Robotics*, 4 (26):eaav3123, 2019. doi:10.1126/scirobotics.aav3123. URL https://www.science.org/doi/abs/10.1126/scirobotics.aav3123.

[2] M. Greenspan, J. Lam, M. Godard, I. Zaidi, S. Jordan, W. Leckie, K. Anderson, and D. Dupuis. Toward a competitive pool-playing robot. *Computer*, 41(1):46–53, 2008.

[3] R. E. Moutaouaffiq. Billiardbot: Physics-aware planning for robotic billiards. https://rachad47.github.io/rwae/BilliardBot.html, 2025.

[4] D. B. D'Ambrosio, N. Jaitly, V. Sindhwani, K. Oslund, P. Xu, N. Lazic, A. Shankar, T. Ding, J. Abelian, E. Coumans, et al. Robotic table tennis: A case study into a high speed learning system. In *Robotics: Science and Systems*, 2023.

[5] 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/.

[6] M. Minderer, A. Gritsenko, and N. Houlsby. Scaling open-vocabulary object detection. *Advances in Neural Information Processing Systems*, 36:72983–73007, 2023.

[7] N. Ravi, V. Gabeur, Y.-T. Hu, R. Hu, C. Ryali, T. Ma, H. Khedr, R. Rädle, C. Rolland, L. Gustafson, E. Mintun, J. Pan, K. V. Alwala, N. Carion, C.-Y. Wu, R. Girshick, P. Dollár, and C. Feichtenhofer. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024. URL https://arxiv.org/abs/2408.00714.

[8] R. S. Sutton, A. G. Barto, et al. Reinforcement learning. *Journal of Cognitive Neuroscience*, 11(1):126–134, 1999.

[9] D. Yarats, I. Kostrikov, and R. Fergus. Image augmentation is all you need: Regularizing deep reinforcement learning from pixels. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=GY6-6sTvGaf.

[10] S. Nair, A. Rajeswaran, V. Kumar, C. Finn, and A. Gupta. R3m: A universal visual representation for robot manipulation. In *Conference on Robot Learning*, pages 892–909. PMLR, 2023.

[11] A. Hussein, M. M. Gaber, E. Elyan, and C. Jayne. Imitation learning: A survey of learning methods. *ACM Computing Surveys (CSUR)*, 50(2):1–35, 2017.

[12] M. Shridhar, L. Manuelli, and D. Fox. Perceiver-actor: A multi-task transformer for robotic manipulation. In *Conference on Robot Learning*, pages 785–799. PMLR, 2023.

[13] P. Florence, C. Lynch, A. Zeng, O. A. Ramirez, A. Wahid, L. Downs, A. Wong, J. Lee, I. Mordatch, and J. Tompson. Implicit behavioral cloning. In *Conference on robot learning*, pages 158–168. PMLR, 2022.

[14] C. Chi, Z. Xu, S. Feng, E. Cousineau, Y. Du, B. Burchfiel, R. Tedrake, and S. Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, 2024.

[15] S. Tunyasuvunakool, A. Muldal, Y. Doron, S. Liu, S. Bohez, J. Merel, T. Erez, T. Lillicrap, N. Heess, and Y. Tassa. dm_control: Software and tasks for continuous control. *Software Impacts*, 6:100022, 2020. ISSN 2665-9638. doi:https://doi.org/10.1016/j.simpa.2020.100022. URL https://www.sciencedirect.com/science/article/pii/S2665963820300099.

[16] M. Denil, P. Agrawal, T. D. Kulkarni, T. Erez, P. Battaglia, and N. de Freitas. Learning to perform physics experiments via deep reinforcement learning. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=r1nTpv9eg.

[17] B. D. Argall, S. Chernova, M. Veloso, and B. Browning. A survey of robot learning from demonstration. *Robotics and autonomous systems*, 57(5):469–483, 2009.

[18] R. Sinha, A. Sharma, S. Banerjee, T. Lew, R. Luo, S. M. Richards, Y. Sun, E. Schmerling, and M. Pavone. A system-level view on out-of-distribution data in robotics. *arXiv preprint arXiv:2212.14020*, 2022.

[19] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, X. Chen, K. Choromanski, T. Ding, D. Driess, A. Dubey, C. Finn, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. *arXiv preprint arXiv:2307.15818*, 2023.

[20] O. X.-E. Collaboration, A. O'Neill, A. Rehman, A. Gupta, 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. Bharadhwaj, 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. Vakil, 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. Tulsiani, 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. Kumar, 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.

[21] M. J. Kim, K. Pertsch, S. Karamcheti, T. Xiao, A. Balakrishna, S. Nair, R. Rafailov, E. P. Foster, P. R. Sanketi, Q. Vuong, et al. Openvla: An open-source vision-language-action model. In *8th Annual Conference on Robot Learning*, 2024.

[22] K. Black, N. Brown, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai, L. Groom, K. Hausman, B. Ichter, S. Jakubczak, T. Jones, L. Ke, S. Levine, A. Li-Bell, M. Mothukuri, S. Nair,

K. Pertsch, L. X. Shi, J. Tanner, Q. Vuong, A. Walling, H. Wang, and U. Zhilinsky. π_0: A vision-language-action flow model for general robot control, 2024. URL https://arxiv.org/abs/2410.24164.

[23] J. Wen, Y. Zhu, Z. Tang, J. Li, Y. Peng, C. Shen, and F. Feng. Dexvla: Vision-language model with plug-in diffusion expert for visuomotor policy learning. *arXiv preprint arXiv:2502.05855*, 2025.

[24] H. Huang, F. Liu, L. Fu, T. Wu, M. Mukadam, J. Malik, K. Goldberg, and P. Abbeel. Otter: A vision-language-action model with text-aware feature extraciton. *arXiv preprint arXiv:2503.03734*, 2025.

[25] C. Zhou, Q. Li, C. Li, J. Yu, Y. Liu, G. Wang, K. Zhang, C. Ji, Q. Yan, L. He, et al. A comprehensive survey on pretrained foundation models: A history from bert to chatgpt. *International Journal of Machine Learning and Cybernetics*, pages 1–65, 2024.

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

[27] S. Fridovich-Keil, A. Yu, M. Tancik, Q. Chen, B. Recht, and A. Kanazawa. Plenoxels: Radiance fields without neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5501–5510, 2022.

[28] T. Müller, A. Evans, C. Schied, and A. Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Trans. Graph.*, 41(4):102:1–102:15, July 2022. doi: 10.1145/3528223.3530127. URL https://doi.org/10.1145/3528223.3530127.

[29] T. Chen, P. Culbertson, and M. Schwager. Catnips: Collision avoidance through neural implicit probabilistic scenes. *IEEE Transactions on Robotics*, 2024.

[30] T. Chen, O. Shorinwa, J. Bruno, A. Swann, J. Yu, W. Zeng, K. Nagami, P. Dames, and M. Schwager. Splat-nav: Safe real-time robot navigation in gaussian splatting maps. *arXiv preprint arXiv:2403.02751*, 2024.

[31] X. Lei, M. Wang, W. Zhou, and H. Li. Gaussnav: Gaussian splatting for visual navigation. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (01):1–14, 2025.

[32] J. Abou-Chakra, K. Rana, F. Dayoub, and N. Suenderhauf. Physically embodied gaussian splatting: A realtime correctable world model for robotics. In *8th Annual Conference on Robot Learning*, 2024. URL https://openreview.net/forum?id=AEq0onGrN2.

[33] M. Zhang, K. Zhang, and Y. Li. Dynamic 3d gaussian tracking for graph-based neural dynamics modeling. In *8th Annual Conference on Robot Learning*.

[34] K. M. Jatavallabhula, M. Macklin, F. Golemo, V. Voleti, L. Petrini, M. Weiss, B. Considine, J. Parent-Levesque, K. Xie, K. Erleben, L. Paull, F. Shkurti, D. Nowrouzezahrai, and S. Fidler. gradsim: Differentiable simulation for system identification and visuomotor control. *International Conference on Learning Representations (ICLR)*, 2021. URL https://openreview.net/forum?id=c_E8kFWfhp0.

[35] R. Liu, A. Canberk, S. Song, and C. Vondrick. Differentiable robot rendering. In *8th Annual Conference on Robot Learning*.

[36] J. Low, M. Adang, J. Yu, K. Nagami, and M. Schwager. Sous vide: Cooking visual drone navigation policies in a gaussian splatting vacuum. *arXiv preprint arXiv:2412.16346*, 2024.

[37] X. Li, J. Li, Z. Zhang, R. Zhang, F. Jia, T. Wang, H. Fan, K.-K. Tseng, and R. Wang. Robogsim: A real2sim2real robotic gaussian splatting simulator. *arXiv preprint arXiv:2411.11839*, 2024.

[38] M. N. Qureshi, S. Garg, F. Yandun, D. Held, G. Kantor, and A. Silwal. Splatsim: Zero-shot sim2real transfer of rgb manipulation policies using gaussian splatting. *arXiv preprint arXiv:2409.10161*, 2024.

[39] A. Quach, M. Chahine, A. Amini, R. Hasani, and D. Rus. Gaussian splatting to real world flight navigation transfer with liquid networks. In *8th Annual Conference on Robot Learning*.

[40] M. Torne, A. Simeonov, Z. Li, A. Chan, T. Chen, A. Gupta, and P. Agrawal. Reconciling reality through simulation: A real-to-sim-to-real approach for robust manipulation. *arXiv preprint arXiv:2403.03949*, 2024.

[41] L. Barcellona, A. Zadaianchuk, D. Allegro, S. Papa, S. Ghidoni, and E. Gavves. Dream to manipulate: Compositional world models empowering robot imitation learning with imagination. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=3RSLW9YSgk.

[42] L. Meyer, F. Erich, Y. Yoshiyasu, M. Stamminger, N. Ando, and Y. Domae. Pegasus: Physically enhanced gaussian splatting simulation system for 6dof object pose dataset generation. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 10710–10715. IEEE, 2024.

[43] J. Kerr, C. M. Kim, K. Goldberg, A. Kanazawa, and M. Tancik. Lerf: Language embedded radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19729–19739, 2023.

[44] S. Zhou, H. Chang, S. Jiang, Z. Fan, Z. Zhu, D. Xu, P. Chari, S. You, Z. Wang, and A. Kadambi. Feature 3dgs: Supercharging 3d gaussian splatting to enable distilled feature fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21676–21685, 2024.

[45] M. Qin, W. Li, J. Zhou, H. Wang, and H. Pfister. Langsplat: 3d language gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20051–20060, 2024.

[46] W. Shen, G. Yang, A. Yu, J. Wong, L. P. Kaelbling, and P. Isola. Distilled feature fields enable few-shot language-guided manipulation. In *Conference on Robot Learning*, pages 405–424. PMLR, 2023.

[47] G. Lu, S. Zhang, Z. Wang, C. Liu, J. Lu, and Y. Tang. Manigaussian: Dynamic gaussian splatting for multi-task robotic manipulation. In *European Conference on Computer Vision*, pages 349–366. Springer, 2024.

[48] T. Xie, Z. Zong, Y. Qiu, X. Li, Y. Feng, Y. Yang, and C. Jiang. Physgaussian: Physics-integrated 3d gaussians for generative dynamics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4389–4398, 2024.

[49] R.-Z. Qiu, G. Yang, W. Zeng, and X. Wang. Feature splatting: Language-driven physics-based scene synthesis and editing. In *European Conference on Computer Vision (ECCV)*, 2024.

[50] H. Zhao, H. Wang, X. Zhao, H. Fei, H. Wang, C. Long, and H. Zou. Efficient physics simulation for 3d scenes via mllm-guided gaussian splatting, 2025. URL https://arxiv.org/abs/2411.12789.

[51] C. Jiang, C. Schroeder, J. Teran, A. Stomakhin, and A. Selle. The material point method for simulating continuum materials. In *Acm siggraph 2016 courses*, pages 1–52. 2016.

[52] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

[53] V. Yugay, Y. Li, T. Gevers, and M. R. Oswald. Gaussian-slam: Photo-realistic dense slam with gaussian splatting. *arXiv preprint arXiv:2312.10070*, 2023.

[54] M. Turkulainen, X. Ren, I. Melekhov, O. Seiskari, E. Rahtu, and J. Kannala. Dn-splatter: Depth and normal priors for gaussian splatting and meshing. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2025.

[55] Q.-Y. Zhou, J. Park, and V. Koltun. Open3d: A modern library for 3d data processing.

[56] E. Todorov, T. Erez, and Y. 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. doi:10.1109/IROS.2012.6386109.

[57] E. Coumans and Y. Bai. Pybullet, a python module for physics simulation for games, robotics and machine learning. `http://pybullet.org`, 2016–2023.

[58] G. Authors. Genesis: A universal and generative physics engine for robotics and beyond, December 2024. URL `https://github.com/Genesis-Embodied-AI/Genesis`.

[59] B. Delaunay. Sur la sphère vide. a la mémoire de georges voronoï. *Bulletin de l'Acadeémie des Sciences de l'URSS. Classe des sciences matheématiques et naturelles.*, (6):793–800, 1934.

[60] C. B. Barber, D. P. Dobkin, and H. Huhdanpaa. The quickhull algorithm for convex hulls. *ACM Transactions on Mathematical Software (TOMS)*, 22(4):469–483, 1996.

[61] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24 (6):381–395, 1981.

[62] J. A. Nelder and R. Mead. A simplex method for function minimization. *The computer journal*, 7(4):308–313, 1965.

[63] D. Koschier, J. Bender, B. Solenthaler, and M. Teschner. Smoothed particle hydrodynamics techniques for the physics based simulation of fluids and solids. *arXiv preprint arXiv:2009.06944*, 2020.

[64] S. Nasiriany, F. Xia, W. Yu, T. Xiao, J. Liang, I. Dasgupta, A. Xie, D. Driess, A. Wahid, Z. Xu, et al. Pivot: Iterative visual prompting elicits actionable knowledge for vlms. In *International Conference on Machine Learning*, pages 37321–37341. PMLR, 2024.

[65] Z. Fan, K. Wen, W. Cong, K. Wang, J. Zhang, X. Ding, D. Xu, B. Ivanovic, M. Pavone, G. Pavlakos, Z. Wang, and Y. Wang. Instantsplat: Sparse-view gaussian splatting in seconds, 2024.

[66] J. Yang, J. Huang, B. Ivanovic, Y. Chen, Y. Wang, B. Li, Y. You, A. Sharma, M. Igl, P. Karkus, et al. Storm: Spatio-temporal reconstruction model for large-scale outdoor scenes. In *The Thirteenth International Conference on Learning Representations*.

[67] S. Szymanowicz, J. Y. Zhang, P. Srinivasan, R. Gao, A. Brussee, A. Holynski, R. Martin-Brualla, J. T. Barron, and P. Henzler. Bolt3D: Generating 3D Scenes in Seconds. *arXiv:2503.14445*, 2025.

[68] S. Le Cleac'h, H.-X. Yu, M. Guo, T. Howell, R. Gao, J. Wu, Z. Manchester, and M. Schwager. Differentiable physics simulation of dynamics-augmented neural objects. *IEEE Robotics and Automation Letters*, 8(5):2780–2787, 2023.

[69] M. Macklin. Warp: A high-performance python framework for gpu simulation and graphics. `https://github.com/nvidia/warp`, March 2022. NVIDIA GPU Technology Conference (GTC).

[70] Z. Huang, F. Chen, Y. Pu, C. Lin, H. Su, and C. Gan. Diffvl: Scaling up soft body manipulation using vision-language driven differentiable physics. *Advances in Neural Information Processing Systems*, 36:29875–29900, 2023.

[71] D. Morton and M. Pavone. Safe, task-consistent manipulation with operational space control barrier functions. In *IEEE/RSJ Int. Conf. on Intelligent Robots & Systems*, 2025. URL https://arxiv.org/pdf/2503.06736.

[72] X. Wei, M. Liu, Z. Ling, and H. Su. Approximate convex decomposition for 3d meshes with collision-aware concavity and tree search. *ACM Transactions on Graphics (TOG)*, 41(4):1–18, 2022.