Phys2Real: Physically-Informed Gaussian Splatting for Adaptive Sim-to-Real Transfer in Robotic Manipulation

Maggie Wang[1], Stephen Tian[2], Jiajun Wu[2], and Mac Schwager[1]

***Abstract*—Learning robotic manipulation policies directly in the real world can be expensive and time-consuming, motivating the use of simulation for scalable training. However, effective sim-to-real transfer remains a central challenge in reinforcement learning for robotic manipulation, particularly for tasks that require precise physical dynamics. We present Phys2Real, a real-to-sim-to-real pipeline that generates object-centric digital twins using geometry from 3D reconstructions and physical priors inferred from vision-language models (VLMs). Our approach combines 3D Gaussian Splatting (GSplat) for high-fidelity geometric reconstructions with VLM-based estimates of physical parameters, such as friction and center of mass (CoM). Unlike domain randomization, which trains policies to be robust across broad parameter ranges and often results in averaged behaviors that may not account for object-specific dynamics, Phys2Real conditions reinforcement learning (RL) policies on known physical parameters during training and VLM-inferred parameter estimates at test time. This conditioning enables precise adaptation to novel objects. We evaluate our method on two planar pushing tasks: a T-block with low friction and a hammer with off-center mass distribution, showing improvement in accuracy, success rate (100% vs 60%), and task completion time compared to domain-randomization baselines. Phys2Real offers a step toward more adaptable manipulation systems that integrate visual reconstruction, physical reasoning, and adaptive control.**

***Index Terms*—sim-to-real, reinforcement learning, robotics**

I. Introduction

DEPLOYING robotic manipulation policies trained in simulation to the real world remains a fundamental challenge, especially for tasks requiring fine-grained physical dynamics. Robots must adapt to varying object properties such as friction, mass distribution, and compliance, which significantly affect manipulation outcomes but are difficult to model precisely. While learning from demonstrations has shown significant promise, it often lacks the physical grounding and reasoning needed to adapt to novel objects. Reinforcement learning (RL) provides a mechanism for real-time adaptation, but bridging the sim-to-real gap remains a critical obstacle.

Domain randomization (DR) has been the dominant approach for sim-to-real transfer when training robotic policies with RL. By training policies across randomized simulation parameters, DR aims to develop policies robust to real-world variations [1], but they may generalize poorly to out-of-distribution object physical properties. Even when dynamics lie within range, policies often default to averaged behaviors that may not account for object-specific variations.

In this work, we leverage object-centric digital twins—simulation assets that replicate real-world object geometry—to create realistic training environments for learning manipulation policies. However, most digital twins capture only shape or appearance, not physical properties. This motivates the question: **Can using more accurate digital twins, where the policy is conditioned on its *physical properties*, improve robot manipulation performance in real-world environments after training in simulation?**

Human manipulation capabilities offer inspiration for addressing this challenge. When encountering a new object, humans form initial judgments about its physical properties from visual appearance, then refine these estimates through interaction. This integration of perception and physical reasoning enables humans to adapt their manipulation strategies to specific object properties without requiring extensive experience with each new object. Our approach seeks to provide robots with a similar ability to estimate and adapt to physical properties.

We propose Phys2Real, a framework that bridges the sim-to-real gap by creating **physically-informed digital twins from real-world observations**. Phys2Real comprises three stages: (1) real-to-sim reconstruction, (2) physics-conditioned policy learning, and (3) sim-to-real transfer using VLM-based parameter estimation. To our knowledge, our approach is the first to combine 3D Gaussian Splatting (GSplat) reconstructions with vision-language model (VLM)-based physical parameter estimation to create physics-informed digital twins for robotic manipulation. By conditioning RL policies on these estimated parameters, where the parameter comes from simulation during training and the VLM during test time, Phys2Real enables object-specific adaptation that outperforms conventional domain randomization. The VLM provides a "warm start" estimate from visual input, and future work will explore refining these estimates online through interaction.

We evaluate our approach on two planar pushing tasks, a class of non-prehensile tasks that require an understanding of friction, mass distribution, and contact dynamics. Our experiments focus on (1) **T-block pushing**, where the object's friction affects its rotational dynamics, and (2) **Hammer pushing**, where the object's off-center center of mass (CoM) results in complex motion dynamics.

Our results indicate that policies conditioned on physical parameters from the VLM improve sim-to-real transfer in execution time, accuracy, and success rate, compared to standard DR. This work highlights the potential of combining physical

[1]Authors are with the Department of Aeronautics and Astronautics, Stanford University, Stanford, CA 94305, USA. {mbwang, schwager}@stanford.edu

[2]Authors are with the Department of Computer Science, Stanford University, Stanford, CA 94305, USA. {tians, jiajunw}@stanford.edu

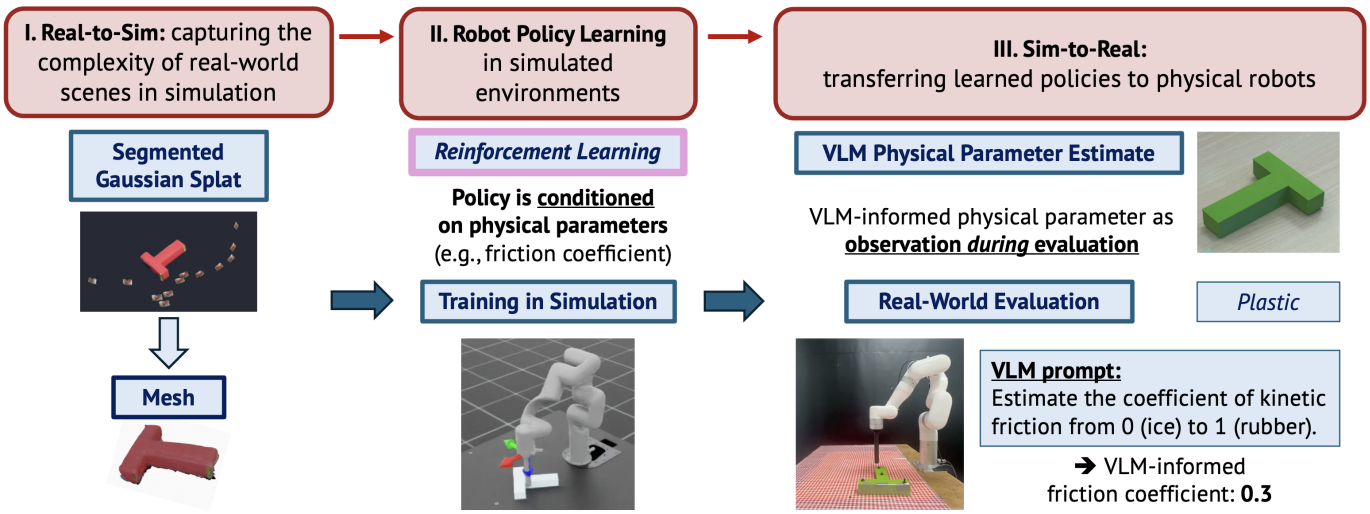

Fig. 1: Phys2Real Overview: We reconstruct real-world objects using GSplats to generate simulation-ready, object-centric meshes. During training, the policy is conditioned on known physical parameters (e.g., friction, center of mass) in simulation. At test time, a vision-language model estimates these parameters from images, enabling sim-to-real transfer with object-specific physical priors.

priors, visual perception, and structured simulation to enable more general and adaptive robotic manipulation.

II. Related Work

Our approach bridges multiple research directions in robotics and AI: sim-to-real transfer methods, policy adaptation techniques, digital twin reconstruction, and physical reasoning with foundation models. While prior work has explored these areas individually, Phys2Real uniquely combines high-fidelity 3D reconstructions with VLM-based physical parameter estimates to create physics-informed digital twins for robotic manipulation.

A. Sim-to-real transfer

The sim-to-real gap remains a fundamental challenge for deploying policies learned in simulation to real-world environments. Domain randomization (DR), introduced by Tobin et al. [1], addresses this gap by randomizing simulation properties during training to develop policies robust to a wide range of environmental variations. This approach was extended to Automatic Domain Randomization (ADR), which was key to OpenAI's successful manipulation of a Rubik's cube with a robotic hand [2].

Despite its popularity, DR suffers from significant limitations. DR policies typically default to averaged behaviors that sacrifice performance for robustness, failing to adapt to object-specific variations. Even when successful, a policy trained with DR may generalize broadly but cannot actively compensate for specific sim-to-real discrepancies during deployment, leading to suboptimal performance when real-world objects deviate significantly from the training distribution, as shown in our experiments in Section IV.

An alternative approach is system identification [3], which explicitly calibrates simulation parameters to match real-world observations. However, these methods often require manual parameter tuning and yield static models that cannot adapt to varying conditions. Our work combines elements of both approaches by first estimating physical parameters via VLMs and then enabling adaptation through policy conditioning.

B. Policy adaptation

Rapid Motor Adaptation (RMA) [4], initially demonstrated for legged locomotion, trains an RL policy with an adaptation module that uses privileged information during simulation training and infers environmental properties through using history during runtime. Liang et al. introduce RMA for manipulator arms [5], showing improved generalization to novel objects and disturbances.

While RMA and similar policy adaptation frameworks enable robots to adjust to new conditions, they typically rely on learning adaptation strategies from data without explicit physical grounding. In contrast, Phys2Real conditions policies directly from physically interpretable parameters (such as friction and CoM) estimated from visual observations, creating an interpretable adaptation mechanism that leverages prior knowledge about physical dynamics.

C. Digital twin simulations and photorealistic rendering

Recent advances in neural scene reconstruction have enabled the creation of photorealistic digital twins for robotics. Neural Radiance Fields (NeRF) [6] and Gaussian Splatting (GSplat) [7] can reconstruct 3D scenes from a series of images, reducing the visual sim-to-real gap. Frameworks including SplatSim [8], RoboGSim [9], and RL-GSBridge [10] have demonstrated GSplat's effectiveness as a simulation renderer.

However, current digital twin approaches focus primarily on visual fidelity while neglecting object physical properties. They create visually realistic environments, but rely on conventional physics engines with default parameters that may not match real-world dynamics. Torne et al. addressed this limitation in RialTo [11] by enabling users to specify physical properties for scanned environments, but their approach requires manual annotation rather than automatic estimation.

Phys2Real advances this field by creating digital twins that are both visually accurate and physically grounded. We leverage GSplats for high-fidelity visual reconstruction to generate simulation assets and augment these digital twins with estimated physical parameters during test time.

D. VLMs for physical reasoning

Recent work has demonstrated that large vision-language models (VLMs) show capability for physical reasoning. PhysObjects [12] fine-tunes InstructBLIP to estimate physical attributes such as material properties, weight, and fragility from visual inputs. However, this work focuses primarily on using these estimates for high-level planning rather than low-level control.

Phys2Real builds on these insights by using VLMs to estimate physical parameters from images. Unlike previous work that uses VLMs primarily for high-level planning, we directly incorporate VLM-estimated physical parameters into the control policy, enabling more accurate manipulation performance.

In summary, Phys2Real represents a novel integration of structured 3D reconstruction, physical parameter estimation via VLMs, and adaptive policy conditioning. By combining these components, we create a system that generates physically-grounded digital twins from real-world observations and leverage them for improved sim-to-real transfer in robotic manipulation tasks.

III. Methods

Phys2Real consists of three stages, as illustrated in Figure 1: real-to-sim reconstruction, physics-informed policy learning, and sim-to-real transfer with VLM-based parameter estimation. Physical information is incorporated during policy learning and test-time adaptation to improve transfer performance.

A. Real-to-sim reconstruction

Our reconstruction pipeline transforms real-world objects into simulation-ready assets. As shown in Figure 3, images of real-world objects are segmented with SAM-2 [13] and reconstructed into object-centric GSplats using SuGaR [14]. We mirror the GSplat across its primary axis of symmetry and apply the Marching Cubes algorithm to extract a clean, watertight mesh. While this pipeline works well for approximately symmetric objects like T-blocks and hammers, mirroring can distort the true shape and mass distribution of asymmetric objects. Extending to asymmetric objects would require alternative meshing strategies to preserve geometric fidelity.

B. Policy learning

We train our policy using Proximal Policy Optimization (PPO) [15] with 4096 parallel environments and an asymmetric actor-critic architecture in IsaacLab [16]. As shown in Figure 2, the actor is conditioned on object pose, end-effector position, and object physical properties (e.g., friction, CoM). The critic receives privileged observations including object velocity and pose. Both actor and critic share a feedforward MLP with hidden layers of size [128, 64] and ELU activations.

During training, the policy is conditioned on known physical properties in simulation. During evaluation, we use GPT-4o [17] to predict these parameters from images, enabling physical adaptation during sim-to-real transfer. We compare against domain randomization and LSTM-based baselines.

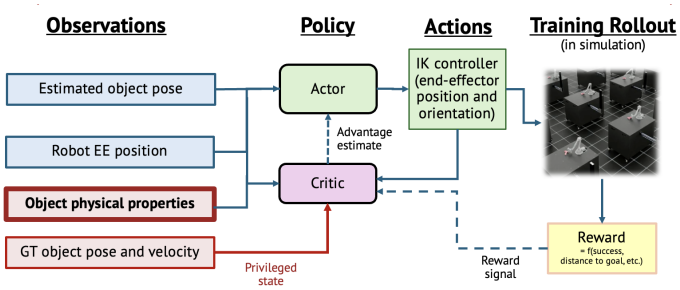

Fig. 2: Overview of the Phys2Real policy training setup. The actor is conditioned on object physical properties (e.g., friction, CoM), while the critic uses privileged ground-truth information in simulation to estimate advantage values and compute reward. During training, the physical properties are provided directly from the simulator. At test time, these values are estimated by a vision-language model (VLM) from visual input.

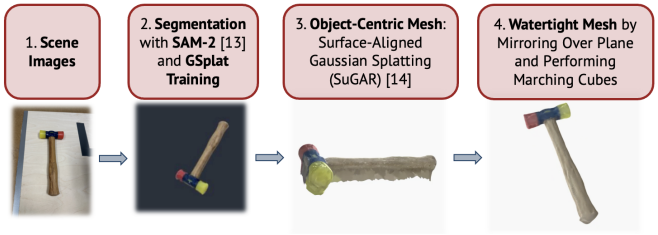

Fig. 3: Real-to-sim mesh reconstruction pipeline. Starting from a video of the object, we extract frames and segment the target object using SAM-2. We then train a GSplat and convert the reconstruction into a surface-aligned object-centric mesh using SuGaR [14]. Finally, we generate a clean, watertight mesh, resulting in a simulation-ready asset.

C. Sim-to-real transfer with physical parameter estimation

An image from the reconstruction is passed into a VLM (GPT-4o) to estimate relevant physical parameters. We estimate task-relevant physical parameters for each object using prompts that are designed to elicit numerical estimates. The prompts can be found in Appendix A. For the T-block, we estimate the friction coefficient, which affects rotational dynamics during pushing. For the hammer, we estimate the CoM along the primary axis, which affects its motion dynamics during manipulation.

We demonstrate our approach on planar pushing tasks using a 6-DOF UFactory xArm robotic arm in simulation and real-world evaluation. The robot uses a cylindrical end-effector to push objects on a table, with observations including object pose, robot state, and estimated physical parameters, while actions are the end-effector xy positions.

For real-world evaluation, we use motion capture to track object poses and evaluate performance. While the current setup relies on motion capture for accurate pose estimation, the pipeline is designed to work with visual inputs alone, and we aim to replace motion capture with perception-based tracking in future work.

TABLE I: Performance comparison on the T-block pushing task in a low-friction regime, where both the object and table surface are covered in plastic. We compare Phys2Real, which uses a VLM-estimated friction coefficient of 0.3, against standard domain randomization (DR) and a DR+LSTM baseline. Metrics include success rate over five trials, final position error, orientation error, and task completion time. Dashes indicate that the policy failed to complete the task, so those metrics were not recorded.

Method	Success Rate (%)	Pos. Error (m)	Orient. Error (deg)	Time (s)
Phys2Real [0.3]	**100**	0.0107	**1.258**	**42.92**
DR [0.3, 1.5]	60	**0.0057**	1.807	70.03
DR [0.3, 1.5] + LSTM	0	–	–	–

TABLE II: Performance comparison on the hammer pushing task. Phys2Real is conditioned on a VLM-estimated CoM located at 0.09m along the hammer's main axis. The DR baseline uses a parameter range of [–0.11, 0.11]m, chosen to span the full possible variation in CoM along the hammer's 0.22m length. All DR baseline trials failed to complete the task, so no error or timing metrics are reported.

Method	Success Rate (%)	Pos. Error (m)	Orient. Error (deg)	Time (s)
Phys2Real [0.09]	**100**	**0.0182**	**1.918**	**40.798**
DR [-0.11, 0.11]	0	–	–	–

IV. Results

We evaluate Phys2Real on two non-prehensile manipulation tasks that require accurate physical modeling for successful execution:

- **T-block pushing:** This task tests adaptation to varying friction coefficients, which impacts the block's rotational dynamics during pushing.
- **Hammer pushing:** The off-center CoM creates complex motion dynamics that must be accounted for during manipulation.

For both tasks, we measure four performance metrics: (1) success rate, (2) final position error in meters, (3) final orientation error in degrees, and (4) task completion time in seconds. We define success as achieving less than 3cm positional error and less than 20° orientation error relative to the target pose. Each method was evaluated over 5 trials.

A. T-block pushing

Table I compares the performance of Phys2Real against two baselines: standard domain randomization (DR) across the range of [0.3, 1.5] and DR with an LSTM-based adaptation module. The results show that Phys2Real achieved a success rate of 100% compared to 60% for standard DR and 0% for the LSTM approach. While the DR baseline achieved a slightly lower positional error when successful, Phys2Real shows better orientation accuracy and reduced task completion time. The failure of the LSTM adaptation approach suggests that naive adaptation mechanisms struggle to transfer from simulation to the real world when the object dynamics differ from training.

B. Hammer pushing

The hammer pushing task highlights the limitations of DR when applied to objects with asymmetric mass distributions. As shown in Table II, Phys2Real achieved a 100% success rate by explicitly conditioning on the estimated CoM (9 cm offset from the object's center), while the DR baseline fails entirely. The failure of the DR baseline demonstrates its inability to handle the dynamics of objects with off-center mass.

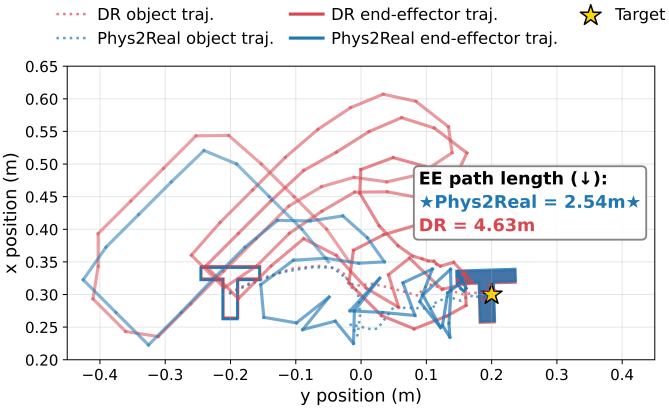

Fig. 4: Comparison of two real-world T-block pushing trajectories using Phys2Real (blue) and the DR baseline (red). Each dashed line shows the object's trajectory, with the lines indicating the end-effector paths. Final object poses are filled (with the star marking the target final position), and initial poses are outlined. The Phys2Real policy completes the task with a shorter end-effector trajectory (2.54m) compared to the DR policy (4.63m), leading to a faster policy execution.

V. Conclusion

We present Phys2Real, a real-to-sim-to-real pipeline that improves robot manipulation by using physical properties, such as friction and CoM, estimated from VLMs. To our knowledge, this is the first approach to combine 3D Gaussian Splatting reconstructions with VLM-based physical parameter estimation for robotic manipulation, showing that VLMs can help bridge the physical sim-to-real gap.

This work moves toward building world models that integrate visual geometry, semantic reasoning, and physical priors, offering a step toward more general and adaptive robotic systems that learn from both perception and physical interaction. Future work will explore integrating learning-based adaptation methods such as Rapid Motor Adaptation (RMA) [4] to refine physical estimates in real time, and extending our framework to prehensile manipulation and deformable objects.

ACKNOWLEDGMENTS

The authors thank John Tucker, Ola Shorinwa, and Rohan Thakker for valuable discussions and feedback. This work is supported by the NASA NSTGRO Fellowship.

APPENDIX A
VLM PROMPTS

We input the following prompts to GPT-4o to estimate task-relevant physical parameters. Although we did not conduct a full prompt ablation study, the estimated values remained consistent across semantically similar prompts and produced stable control behavior when used for policy conditioning.

- **T-block**: "On a scale of 0 to 1, estimate the coefficient of kinetic friction for the object in this image. On this scale, 0 is ice, and 1 is rubber. Use visual details of the object as well as the surface that it is on to reference online material and make your estimate. Respond with ONLY a numerical estimate."
- **Hammer**: "Given this image of a hammer, can you help me estimate the normalized location of its center of mass (CoM) along the main axis of motion, where 0 corresponds to the leftmost end and 1 to the rightmost end? Please take into account the materials of each part of the hammer when calculating the CoM."

These VLM outputs are then used to condition the RL policy at test time.

REFERENCES

[1] 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," Mar. 2017, arXiv:1703.06907 [cs]. [Online]. Available: http://arxiv.org/abs/1703.06907

[2] OpenAI, I. Akkaya, M. Andrychowicz, M. Chociej, M. Litwin, B. McGrew, A. Petron, A. Paino, M. Plappert, G. Powell, R. Ribas, J. Schneider, N. Tezak, J. Tworek, P. Welinder, L. Weng, Q. Yuan, W. Zaremba, and L. Zhang, "Solving Rubik's Cube with a Robot Hand," Oct. 2019, arXiv:1910.07113 [cs]. [Online]. Available: http://arxiv.org/abs/1910.07113

[3] L. Ljung, *System identification: theory for the user*. USA: Prentice-Hall, Inc., 1986.

[4] A. Kumar, Z. Fu, D. Pathak, and J. Malik, "RMA: Rapid Motor Adaptation for Legged Robots," in *Robotics: Science and Systems XVII*. Robotics: Science and Systems Foundation, Jul. 2021. [Online]. Available: http://www.roboticsproceedings.org/rss17/p011.pdf

[5] Y. Liang, K. Ellis, and J. Henriques, "Rapid Motor Adaptation for Robotic Manipulator Arms," Mar. 2024, arXiv:2312.04670 [cs]. [Online]. Available: http://arxiv.org/abs/2312.04670

[6] 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," *CoRR*, vol. abs/2003.08934, 2020. [Online]. Available: https://arxiv.org/abs/2003.08934

[7] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis, "3d gaussian splatting for real-time radiance field rendering," 2023. [Online]. Available: https://arxiv.org/abs/2308.04079

[8] 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," Sep. 2024, arXiv:2409.10161 [cs]. [Online]. Available: http://arxiv.org/abs/2409.10161

[9] 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," Nov. 2024, arXiv:2411.11839. [Online]. Available: http://arxiv.org/abs/2411.11839

[10] Y. Wu, L. Pan, W. Wu, G. Wang, Y. Miao, F. Xu, and H. Wang, "Rl-gsbridge: 3d gaussian splatting based real2sim2real method for robotic manipulation learning," 2025. [Online]. Available: https://arxiv.org/abs/2409.20291

[11] 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," Mar. 2024, arXiv:2403.03949 [cs]. [Online]. Available: http://arxiv.org/abs/2403.03949

[12] J. Gao, B. Sarkar, F. Xia, T. Xiao, J. Wu, B. Ichter, A. Majumdar, and D. Sadigh, "Physically Grounded Vision-Language Models for Robotic Manipulation," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. Yokohama, Japan: IEEE, May 2024, pp. 12 462–12 469. [Online]. Available: https://ieeexplore.ieee.org/document/10610090/

[13] 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," Oct. 2024, arXiv:2408.00714 [cs]. [Online]. Available: http://arxiv.org/abs/2408.00714

[14] A. Guédon and V. Lepetit, "SuGaR: Surface-Aligned Gaussian Splatting for Efficient 3D Mesh Reconstruction and High-Quality Mesh Rendering," Dec. 2023, arXiv:2311.12775 [cs]. [Online]. Available: http://arxiv.org/abs/2311.12775

[15] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal Policy Optimization Algorithms," Aug. 2017, arXiv:1707.06347 [cs]. [Online]. Available: http://arxiv.org/abs/1707.06347

[16] M. Mittal, C. Yu, Q. Yu, J. Liu, N. Rudin, D. Hoeller, J. L. Yuan, R. Singh, Y. Guo, H. Mazhar, A. Mandlekar, B. Babich, G. State, M. Hutter, and A. Garg, "Orbit: A Unified Simulation Framework for Interactive Robot Learning Environments," *IEEE Robotics and Automation Letters*, vol. 8, no. 6, pp. 3740–3747, Jun. 2023. [Online]. Available: https://ieeexplore.ieee.org/document/10107764/

[17] OpenAI, A. Hurst, A. Lerer *et al.*, "GPT-4o System Card," Oct. 2024, arXiv:2410.21276 [cs]. [Online]. Available: http://arxiv.org/abs/2410.21276