X-SIM: Cross-Embodiment Learning via Real-to-Sim-to-Real

Prithwish Dan* Kushal Kedia* Angela Chao Edward W. Duan
Maximus A. Pace Wei-Chiu Ma Sanjiban Choudhury
Cornell University

***Abstract*—Human videos offer a scalable way to train robot manipulation policies, but lack the action labels needed by standard imitation learning algorithms. Existing cross-embodiment approaches try to map human motion to robot actions, but often fail when the embodiments differ significantly. We propose X-SIM, a real-to-sim-to-real framework that uses object motion as a dense and transferable signal for learning robot policies. X-SIM starts by reconstructing a photorealistic simulation from an RGBD human video and tracking object trajectories to define object-centric rewards. These rewards are used to train a reinforcement learning (RL) policy in simulation. The learned policy is then distilled into an image-conditioned diffusion policy using synthetic rollouts rendered with varied viewpoints and lighting. To transfer to the real world, X-SIM introduces an online domain adaptation technique that aligns real and simulated observations during deployment. Importantly, X-SIM does not require any robot teleoperation data. We evaluate it across 5 manipulation tasks in 2 environments and show that it: (1) improves task progress by 30% on average over hand-tracking and sim-to-real baselines, (2) matches behavior cloning with 10× less data collection time, and (3) generalizes to new camera viewpoints and test-time changes. Website: https://portal.cs.cornell.edu/X-Sim/**

I. INTRODUCTION

Human videos offer a natural and scalable source of demonstrations for robot policy learning. However, recent advances in robot foundation models [25, 18] rely entirely on large-scale datasets of robot embodiments [34, 24]. Collecting such data requires labor-intensive and expensive teleoperation to provide high-quality expert demonstrations, making it intractable to scale across diverse tasks and environments. In contrast, human videos (e.g. from YouTube) are abundant and capture a wide range of tasks in natural environments.

Despite their potential, human videos cannot be directly used in widely-adopted imitation learning pipelines [10, 57], as they lack explicit robot action labels. To bridge this gap, prior work attempts to map human trajectories to robot actions, typically assuming visual or kinematic compatibility. Some methods retarget human hand motion to the robot's end-effector [6], but this assumes that human movements are feasible for the robot to replicate [39], which is rarely the case in practice. Other methods reduce the human-robot visual gap by overlaying robot arms on human videos [27, 28], but these rely on solving inverse kinematics, which may be ill-posed due to embodiment mismatch. Another line of work directly translates human videos into robot actions [21, 20, 47], but requires paired human-robot demonstrations, which are expensive and difficult to collect at scale.

We tackle the problem of generating robot training data from action-less human videos. ***Our key insight is that, while human actions are unavailable, the object motion they produce provides a dense supervisory signal for training robot policies in simulation.*** By reconstructing a photorealistic simulation [17] of the human video and tracking object trajectories [48], we define object-centric reward functions that guide RL agents to reproduce the effects of human behavior — even when the robot must take entirely different actions. This enables distillation into real-world image-conditioned robot policies ***without any robot teleoperation data***.

We propose X-SIM, a real-to-sim-to-real framework that bridges the human-robot embodiment gap by learning robot policies in simulation on rewards generated from human videos (Fig. 1). X-SIM first extracts object states from a RGBD human video and transfers them into a photorealistic simulation. It defines a dense object-centric reward to efficiently train state-based RL policies in simulation. X-SIM generates a large synthetic dataset of paired image-action data by rolling out the trained RL policy and rendering the resulting scenes under varied robot poses, object states, viewpoints, and lighting. Using this dataset, it trains an image-conditioned diffusion policy and transfers directly to the real-world without needing any real robot action data. To narrow the sim-to-real gap at deployment, X-SIM utilizes an online domain adaptation technique to align the robot's real world and simulation observations. Our contributions are summarized as:

1) We propose X-SIM, a real-to-sim-to-real framework that learns image-based robot policies from action-less human videos by tracking object states and matching their motion in simulation.
2) We introduce an online domain adaptation technique to continually reduce the sim-to-real gap by aligning real-world observations with simulation at test time, enabling robust sim-to-real transfer.
3) We evaluate X-SIM across 5 manipulation tasks in 2 environments, showing that it (1) improves task progress by 30% on average over hand-tracking and sim-to-real baselines, (2) matches behavior cloning with 10x less data collection time, and (3) enables generalization to test-time changes, including novel camera viewpoints.

II. APPROACH

X-SIM learns real-world, image-conditioned robot policies from action-free RGBD human videos by using object motion

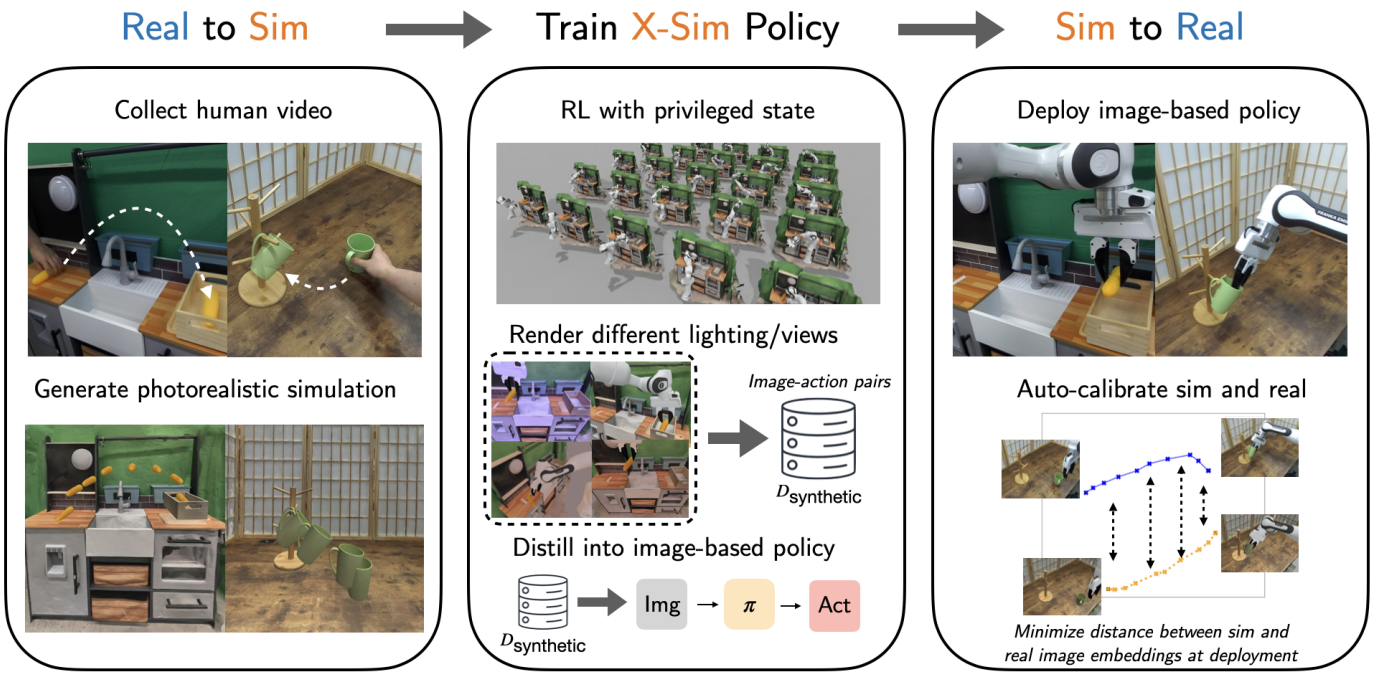


Fig. 1. **Overview of X-SIM:** We introduce X-SIM, a real-to-sim-to-real framework that bridges the human-robot embodiment gap by learning robot policies. **Real-to-Sim**. We generate photorealistic simulation using object-centric rewards generated from human videos. **Training X-Sim**. We first train RL policies with privileged state using GPU-parallelized environment. Then, we collect a diverse image-action dataset use it to distill behaviors into an image-conditioned policy. **Sim-to-Real.** Image-based policy is deployed in the real-world. Its observation encoder automatically calibrates itself by aligning real and sim image observations at test-time.

as supervision. The pipeline has three stages: (1) reconstructing a photo-realistic simulator from human videos and extracting object trajectories, (2) training RL policies in simulation to match object motion and generate synthetic image-action data, and (3) distilling behaviors into an image-conditioned diffusion policy with online domain adaptation.

A. Real-to-Sim Transfer from Human Videos

We treat object motion in human videos as task supervision. First, we use 3D scanning and FoundationPose [48] to track object poses across the video $\mathbf{v_H}$, yielding pose trajectories $\mathbf{s_H} = \{s_H^t\}_{t=1}^T$ with $s_H^t \in \mathrm{SE}(3)^K$. Next, we reconstruct the scene with 2D Gaussian Splatting [17] and import it into ManiSkill [32] to build a realistic simulation. We choose default physics parameters for all objects.

B. Generating Robot Actions in Simulation

We define object-centric rewards using $\mathbf{s_H}$, encouraging the robot to match human-demonstrated object poses:

$$r_{\text{goal}} \propto -d_{pos}(s_H^B, s_R^t) - d_{rot}(s_H^B, s_R^t) \tag{1}$$

where s_H^B is the current goal pose. A privileged-state policy is trained with PPO [40], and successful rollouts are rendered under randomized conditions to build a synthetic dataset $D_{\text{synthetic}} = \{(o_R^t, a_R^t)\}_{t=1}^N$.

C. Sim-to-Real Transfer of Image-Based Policies

We train a Diffusion Policy [10] $\pi_{\text{img}}(a|o)$ on $D_{\text{synthetic}}$ to operate on RGB inputs. To bridge the sim-to-real gap, we replay real robot rollouts in simulation and create paired observations $D_{\text{paired}} = \{(o_R^{\text{sim}}, o_R^{\text{real}})\}$. These are used to align visual features via a contrastive InfoNCE loss:

$$\mathcal{L}_{\text{calibration}} = -\sum_{(o_R^{\text{sim}}, o_R^{\text{real}})} \frac{\exp\left(\frac{s(\phi(o_R^{\text{sim}}), \phi(o_R^{\text{real}}))}{\tau}\right)}{\sum_{o_R'^{\text{real}}} \exp\left(\frac{s(\phi(o_R^{\text{sim}}), \phi(o_R'^{\text{real}}))}{\tau}\right)} \tag{2}$$

where ϕ is the encoder, s is cosine similarity, and τ is a temperature. This alignment improves robustness to real-world visual variation without using teleoperation data.

III. EXPERIMENTS

Experimental Setup. We conduct all experiments using a 7-DOF Franka arm across two real environments: *Kitchen* and *Tabletop* (Fig. 2). RGBD human videos are recorded using a ZED 2 stereo camera, with no constraints on motion or grasp style allowing for natural human execution. Tasks include pick-and-place (`Mustard Place`, `Corn in Basket`, `Shoe on Rack`), non-prehensile manipulation (`Letter Arrange`), and precise insertion (`Mug Insert`). We transfer human videos into simulation using our real-to-sim pipeline. For each task, we train privileged-state policies using PPO [40] in ManiSkill [32] and randomize object and robot poses around the initial demonstration state. Then, the RL policy is distilled into an image-only Diffusion Policy [10]. We assume approximate knowledge of the test-time camera viewpoint and render randomized viewpoints around it during training, adding robustness to small variations. At inference time, X-SIM operates solely on real RGB

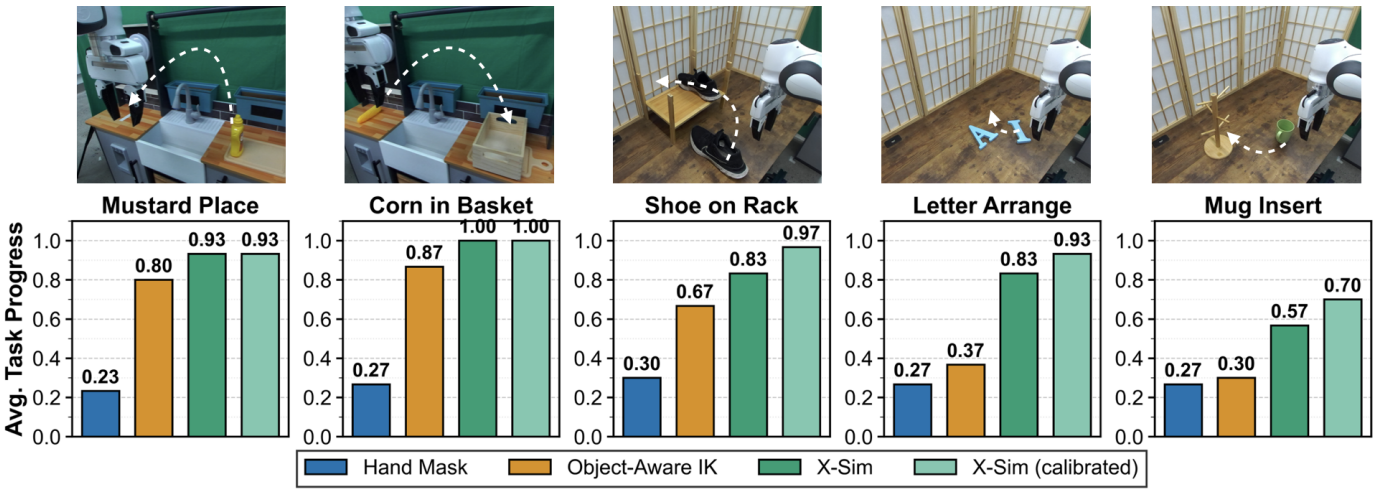


Fig. 2. **Performance on Real-World Tasks:** We report *Avg. Task Progress* on 5 tasks across two environments, and find that X-SIM both with and without calibration consistently outperforms hand-tracking baselines that attempt to retarget human hand motion. A rough sketch of each task is visualized on top.

image input. To align the observation encoder for X-SIM (CALIBRATED), we rollout 10 trajectories of X-SIM in the real-world to collect paired real and sim data. More details about each task and hyperparameters are in the Appendix.

Evaluation Metrics. We report *Average Task Progress* as our primary metric, which captures partial credit across distinct stages of task completion rather than relying on binary success. For grasp-based tasks (`Mustard Place`, `Corn in Basket`, `Shoe on Rack`, `Mug Insert`), progress is divided into three stages: approaching the correct object, successfully grasping it, and completing the manipulation to match the goal configuration from the human video. For the non-prehensile task (`Letter Arrange`), the stages correspond to approaching, rotating, and placing the object correctly. We evaluate all methods over 10 trials, each with slight variations in the object's initial position relative to the demonstrated human video.

A. Bridging the Embodiment Gap via Simulation

We evaluate whether X-SIM can overcome the limitations of hand-retargeting approaches. We compare against two representative baselines:

- **Hand Mask:** [27, 28] Applies a black mask over the human hand in demonstration videos to train an image-conditioned behavior cloning policy. At inference time, the robot arm is similarly masked. This approach, used in PHANTOM [28], assumes all human hand poses can be replicated by the robot. Without this assumption, we do not overlay a robot arm during training.
- **Object-Aware Inverse Kinematics (IK):** [46, 31, 45] Extracts hand trajectories relative to nearby objects, and replays them by applying IK to move the robot end-effector along the same path.

Neither baseline uses simulation. Both extract action labels from human hand pose estimates using HAMER [37], using the same procedure as PHANTOM [28]. We evaluate X-SIM and baselines across 10 real-world rollouts per task (Fig. 2). **Hand Mask** fails due to a large visual gap between human and robot observations, retaining only object location information and rarely progressing beyond the approach phase (Fig. 3). **Object-Aware IK** performs well in *Kitchen* tasks where human and robot have similar execution styles, but breaks down in *Tabletop* tasks due to kinematic infeasibility and mismatched dynamics when directly mimicking human motions. In contrast, **X-SIM**, even without sim-to-real calibration, learns feasible strategies in simulation and transfers them effectively—achieving consistently higher task progress and over 30% gains in the most mismatched settings.

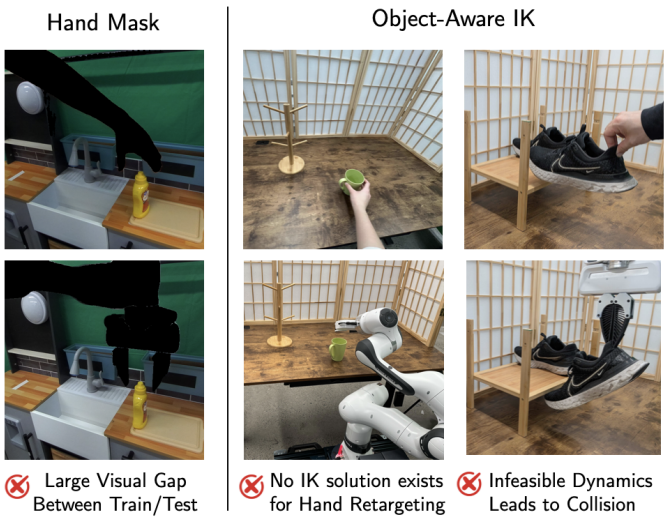


Fig. 3. **Hand Re-targeting Failure Modes: Hand Mask** fails due to a significant visual domain gap between human and robots, even when the motions are physically feasible for the robot. **Object-Aware IK** fails under execution mismatch, as certain human hand motions are kinematically or dynamically infeasible.

B. Sim-to-Real Policy Transfer

Comparison with State-Based Policy. We evaluate X-SIM's ability to transfer from simulation to the real world using only RGB images, and compare it to policy learning approaches based on privileged state, such as object poses. A closely related method, **Human2Sim2Robot** [14], learns in simulation using accurate 6D object poses and attempts to replicate this setup in the real world through object tracking. However, even small tracking errors at inference can push pose-based policies out-of-distribution, leading to failure.

These methods often rely on precise observations that are hard to obtain in practice due to occlusions, depth noise, and imperfect vision models. In contrast, **X-SIM** uses raw images, which provide a more robust and transferable representation. Image-based inputs are less sensitive to real-world noise and align well with modern visuomotor policy architectures. On the `Letter Arrange` task, X-SIM significantly outperforms pose-based baselines in sim-to-real transfer (Table 4), showing that images are a more practical and effective observation modality for real-world deployment.

Metric ↓	H2S2R	X-SIM
Avg. Task Progress	43.3%	83.3%

Fig. 4. We evaluate Avg. Task Progress of X-SIM with image observations against a sim-to-real baseline that uses object state observations on the `Letter Arrange` task.

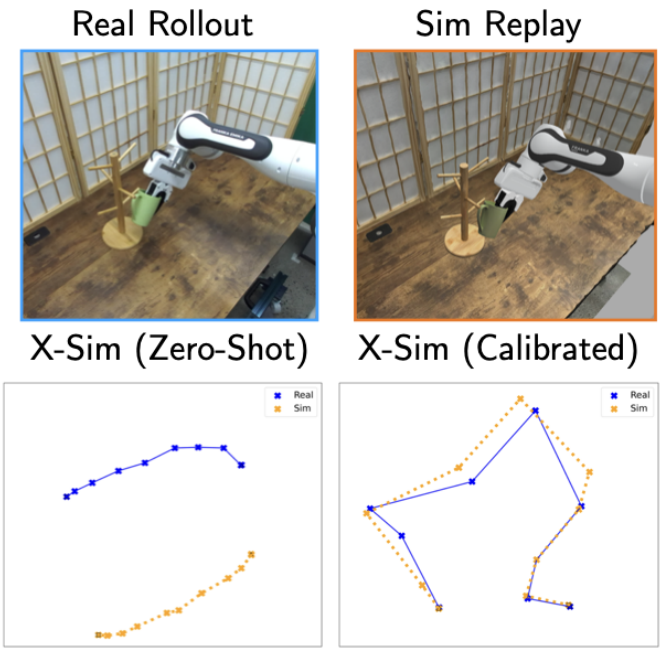


Fig. 5. **Sim-to-Real Calibration:** We compare X-SIM image embeddings using t-SNE before and after calibration for one rollout, and find that our calibration procedure helps remedy the sim-to-real visual gap between the photorealistic simulation and the real-world.

Calibration after Deployment. Recent sim-to-real methods [2, 33] often rely on co-training with real-world demonstrations to bridge the domain gap. In contrast, X-SIM uses only simulation data collected in a photorealistic environment, avoiding the need for teleoperation. While this reduces the observation gap, some visual discrepancies remain due to imperfections in 3D reconstruction and rendering. To address this, X-SIM (CALIBRATED) aligns real and simulated observations online using closed-loop rollouts, as described in Sec. **??**. Notably, this procedure is agnostic to success/failure, and can even benefit from unsuccessful rollouts. We find that X-SIM (CALIBRATED) leads to additional benefits over our base method, with an average increase of 8% in task progress across all tasks and most notably a 13% increase for the most challenging task `Mug Insert`, indicating the ability to learn even from failures (Fig. 2). To further analyze the effects of our calibration procedure, we probe policy observation encoders on a paired simulation/real robot videos and plot the t-SNE embeddings over time in Fig. 5. X-SIM (CALIBRATED) better aligns image embeddings compared to X-SIM, ensuring that the policy avoids overfitting to domain-specific attributes with its calibration loss while still encoding task relevant features with its action prediction loss.

C. Data Efficiency

We study how X-SIM's performance scales with data by modifying the `Mustard Place` task to significantly broaden the initial state distribution of the mustard bottle (visualizations in the Appendix). In this setting, behavior cloning requires extensive robot teleoperation data to cover the distribution. In contrast, X-SIM scales by collecting more human videos—which are faster to obtain (20s per video vs. 60s per robot demo)—and perturbing object poses in simulation for broader coverage. As shown in Fig. 6, X-SIM achieves 90% success with just 1 minute of human video data, compared to 70% success with 10 minutes of robot demonstrations. This highlights X-SIM's efficiency and scalability for training robust robot policies.

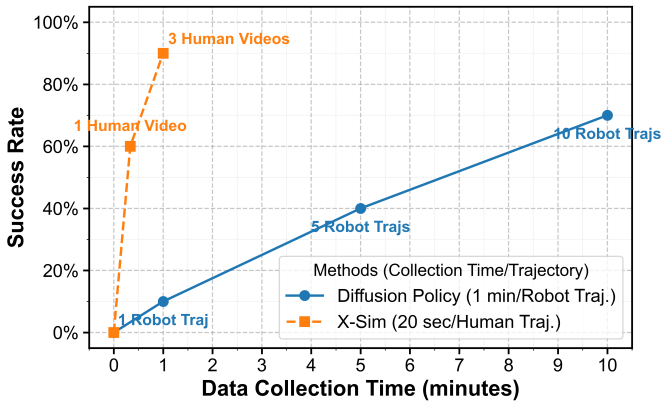


Fig. 6. **Data Efficiency:** X-SIM scales more efficiently with data collection time than behavior cloning from robot teleoperation, achieving comparable success on `Mustard Place` with 10× less time.

References

[1] Jad Abou-Chakra, Krishan Rana, Feras Dayoub, and Niko Sünderhauf. Physically embodied gaussian splatting: A visually learnt and physically grounded 3d representation for robotics. In *Conference on Robot Learning*.

[2] Lars Lien Ankile, Anthony Simeonov, Idan Shenfeld, Mª Mercedes Lopez Torné, and Pulkit Agrawal. From imitation to refinement – residual rl for precise assembly. 2024.

[3] Sridhar Pandian Arunachalam, Sneha Silwal, Ben Evans, and Lerrel Pinto. Dexterous imitation made easy: A learning-based framework for efficient dexterous manipulation. 2022.

[4] Shikhar Bahl, Abhinav Gupta, and Deepak Pathak. Human-to-robot imitation in the wild. 2022.

[5] Homanga Bharadhwaj, Abhi Gupta, Vikash Kumar, and Shubham Tulsiani. Towards generalizable zero-shot manipulation via translating human interaction plans. *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6904–6911, 2023. URL https://api.semanticscholar.org/CorpusID:265551754.

[6] Homanga Bharadhwaj, Abhi Gupta, Shubham Tulsiani, and Vikash Kumar. Zero-shot robot manipulation from passive human videos. volume abs/2302.02011, 2023.

[7] Homanga Bharadhwaj, Debidatta Dwibedi, Abhinav Gupta, Shubham Tulsiani, Carl Doersch, Ted Xiao, Dhruv Shah, Fei Xia, Dorsa Sadigh, and Sean Kirmani. Gen2act: Human video generation in novel scenarios enables generalizable robot manipulation. *ArXiv*, abs/2409.16283, 2024.

[8] Homanga Bharadhwaj, Roozbeh Mottaghi, Abhinav Gupta, and Shubham Tulsiani. Track2act: Predicting point tracks from internet videos enables diverse zero-shot robot manipulation. *arXiv preprint arXiv:2405.01527*, 2024.

[9] Aude G Billard, Sylvain Calinon, and Florent Guenter. Discriminative and adaptive imitation in uni-manual and bi-manual tasks. *Robotics Auton. Syst.*, 54:370–384, 2006.

[10] Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *arXiv preprint arXiv:2303.04137*, 2023.

[11] Cheng Chi, Zhenjia Xu, Chuer Pan, Eric Cousineau, Benjamin Burchfiel, Siyuan Feng, Russ Tedrake, and Shuran Song. Universal manipulation interface: In-the-wild robot teaching without in-the-wild robots. *ArXiv*, abs/2402.10329, 2024.

[12] Tianyuan Dai, Josiah Wong, Yunfan Jiang, Chen Wang, Cem Gokmen, Ruohan Zhang, Jiajun Wu, and Fei-Fei Li. Automated creation of digital cousins for robust policy learning. *ArXiv*, abs/2410.07408, 2024.

[13] Haiwen Feng, Junyi Zhang, Qianqian Wang, Yufei Ye, Pengcheng Yu, Michael J. Black, Trevor Darrell, and Angjoo Kanazawa. St4rtrack: Simultaneous 4d reconstruction and tracking in the world. 2025. URL https://api.semanticscholar.org/CorpusID:277857146.

[14] Tyler Ga, Wei Lum, Olivia Y. Lee, C. Karen Liu, Jeannette Bohg, and Pre-Manip Hand Pose. Crossing the human-robot embodiment gap with sim-to-real rl using one human demonstration. 2025.

[15] Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Q. Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, Miguel Martin, Tushar Nagarajan, Ilija Radosavovic, Santhosh K. Ramakrishnan, Fiona Ryan, Jayant Sharma, Michael Wray, Mengmeng Xu, Eric Z. Xu, Chen Zhao, Siddhant Bansal, Dhruv Batra, Vincent Cartillier, Sean Crane, Tien Do, Morrie Doulaty, Akshay Erapalli, Christoph Feichtenhofer, Adriano Fragomeni, Qichen Fu, Christian Fuegen, Abrham Gebreselasie, Cristina González, James M. Hillis, Xuhua Huang, Yifei Huang, Wenqi Jia, Weslie Khoo, Jáchym Kolár, Satwik Kottur, Anurag Kumar, Federico Landini, Chao Li, Yanghao Li, Zhenqiang Li, Karttikeya Mangalam, Raghava Modhugu, Jonathan Munro, Tullie Murrell, Takumi Nishiyasu, Will Price, Paola Ruiz Puentes, Merey Ramazanova, Leda Sari, Kiran K. Somasundaram, Audrey Southerland, Yusuke Sugano, Ruijie Tao, Minh Vo, Yuchen Wang, Xindi Wu, Takuma Yagi, Yunyi Zhu, Pablo Arbeláez, David J. Crandall, Dima Damen, Giovanni Maria Farinella, Bernard Ghanem, Vamsi Krishna Ithapu, C. V. Jawahar, Hanbyul Joo, Kris Kitani, Haizhou Li, Richard A. Newcombe, Aude Oliva, Hyun Soo Park, James M. Rehg, Yoichi Sato, Jianbo Shi, Mike Zheng Shou, Antonio Torralba, Lorenzo Torresani, Mingfei Yan, and Jitendra Malik. Ego4d: Around the world in 3,000 hours of egocentric video. pages 18973–18990, 2021.

[16] Irmak Güzey, Yinlong Dai, Georgy Savva, Raunaq M. Bhirangi, and Lerrel Pinto. Bridging the human to robot dexterity gap through object-oriented rewards. *ArXiv*, abs/2410.23289, 2024.

[17] Binbin Huang, Zehao Yu, Anpei Chen, Andreas Geiger, and Shenghua Gao. 2d gaussian splatting for geometrically accurate radiance fields. In *International Conference on Computer Graphics and Interactive Techniques*, 2024.

[18] Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Manuel Y. Galliker, Dibya Ghosh, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Devin LeBlanc, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Allen Z. Ren, Lucy Xiaoyang Shi, Laura Smith, Jost Tobias Springenberg, Kyle Stachowicz, James Tanner, Quan Vuong, Homer Rich Walke, Anna Walling, Haohuan Wang, Lili Yu, and Ury Zhilinsky. π 0.5: A vision-language-action model with open-world generalization. 2025.

[19] Aadhithya Iyer, Zhuoran Peng, Yinlong Dai, Irmak

Güzey, Siddhant Haldar, Soumith Chintala, and Lerrel Pinto. Open teach: A versatile teleoperation system for robotic manipulation. *ArXiv*, abs/2403.07870, 2024.

[20] Vidhi Jain, Maria Attarian, Nikhil Joshi, Ayzaan Wahid, Danny Driess, Quan Vuong, Pannag R. Sanketi, Pierre Sermanet, Stefan Welker, Christine Chan, Igor Gilitschenski, Yonatan Bisk, and Debidatta Dwibedi. Vid2robot: End-to-end video-conditioned policy learning with cross-attention transformers. volume abs/2403.12943, 2024.

[21] Eric Jang, Alex Irpan, Mohi Khansari, Daniel Kappler, Frederik Ebert, Corey Lynch, Sergey Levine, and Chelsea Finn. Bc-z: Zero-shot task generalization with robotic imitation learning. volume abs/2202.02005, 2022.

[22] K. Kedia, Prithwish Dan, and Sanjiban Choudhury. One-shot imitation under mismatched execution. *ArXiv*, abs/2409.06615, 2024.

[23] Justin Kerr, Chung Min Kim, Mingxuan Wu, Brent Yi, Qianqian Wang, Ken Goldberg, and Angjoo Kanazawa. Robot see robot do: Imitating articulated object manipulation with monocular 4d reconstruction. In *Conference on Robot Learning*, 2024. URL https://api.semanticscholar.org/CorpusID:272910721.

[24] Alexander Khazatsky, Karl Pertsch, Suraj Nair, Ashwin Balakrishna, Sudeep Dasari, Siddharth Karamcheti, Soroush Nasiriany, Mohan Kumar Srirama, Lawrence Yunliang Chen, Kirsty Ellis, et al. Droid: A large-scale in-the-wild robot manipulation dataset. *arXiv preprint arXiv:2403.12945*, 2024.

[25] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag R. Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. Openvla: An open-source vision-language-action model. *ArXiv*, abs/2406.09246, 2024.

[26] Sateesh Kumar, Jonathan Zamora, Nicklas Hansen, Rishabh Jangir, and Xiaolong Wang. Graph inverse reinforcement learning from diverse videos. 2022.

[27] Marion Lepert, Ria Doshi, and Jeannette Bohg. Shadow: Leveraging segmentation masks for cross-embodiment policy transfer. *ArXiv*, abs/2503.00774, 2025.

[28] Marion Lepert, Jiaying Fang, and Jeannette Bohg. Phantom: Training robots without robots using only human videos. *ArXiv*, abs/2503.00779, 2025.

[29] Sergey Levine, Chelsea Finn, Trevor Darrell, and P. Abbeel. End-to-end training of deep visuomotor policies. *J. Mach. Learn. Res.*, 17:39:1–39:40, 2015.

[30] Chengshu Li, Ruohan Zhang, Josiah Wong, Cem Gokmen, Sanjana Srivastava, Roberto Martín-Martín, Chen Wang, Gabrael Levine, Michael Lingelbach, Jiankai Sun, Mona Anvari, Minjune Hwang, Manasi Sharma, Arman Aydin, Dhruva Bansal, Samuel Hunter, Kyu-Young Kim, Alan Lou, Caleb R. Matthews, Ivan Villa-Renteria, Jerry Huayang Tang, Claire Tang, Fei Xia, Silvio Savarese, Hyowon Gweon, C. Karen Liu, Jiajun Wu, and Li Fei-Fei. Behavior-1k: A benchmark for embodied ai with 1, 000 everyday activities and realistic simulation. In *Conference on Robot Learning*, 2022.

[31] Jinhan Li, Yifeng Zhu, Yuqi Xie, Zhenyu Jiang, Mingyo Seo, Georgios Pavlakos, and Yuke Zhu. Okami: Teaching humanoid robots manipulation skills through single video imitation. *ArXiv*, abs/2410.11792, 2024.

[32] Tongzhou Mu, Z. Ling, Fanbo Xiang, Derek Yang, Xuanlin Li, Stone Tao, Zhiao Huang, Zhiwei Jia, and Hao Su. Maniskill: Generalizable manipulation skill benchmark with large-scale demonstrations. In *NeurIPS Datasets and Benchmarks*, 2021.

[33] Nvidia, Johan Bjorck, Fernando Castaneda, Nikita Cherniadev, Xingye Da, Runyu Ding, LinxiJimFan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, Joel Jang, Zhenyuan Jiang, Jan Kautz, Kaushil Kundalia, Lawrence Lao, Zhiqi Li, Zongyu Lin, Kevin Lin, Guilin Liu, Edith Llontop, Loic Magne, Ajay Mandlekar, Avnish Narayan, Soroush Nasiriany, Scott Reed, You Liang Tan, Guanzhi Wang, Zu Wang, Jing Wang, Qi Wang, Jiannan Xiang, Yuqi Xie, Yinzhen Xu, Zhen-Teng Xu, Seonghyeon Ye, Zhiding Yu, Ao Zhang, Hao Zhang, Yizhou Zhao, Ruijie Zheng, and Yuke Zhu. Gr00t n1: An open foundation model for generalist humanoid robots. *ArXiv*, abs/2503.14734, 2025.

[34] Abhishek Padalkar, Acorn Pooley, Ajinkya Jain, Alex Bewley, Alex Herzog, Alex Irpan, Alexander Khazatsky, Anant Rai, Anikait Singh, Anthony Brohan, et al. Open x-embodiment: Robotic learning datasets and rt-x models. *arXiv preprint arXiv:2310.08864*, 2023.

[35] Austin Patel, Andrew Wang, Ilija Radosavovic, and Jitendra Malik. Learning to imitate object interactions from internet videos. *ArXiv*, abs/2211.13225, 2022.

[36] Shivansh Patel, Xi Yin, Wenlong Huang, Shubham Garg, Hooshang Nayyeri, Fei-Fei Li, Svetlana Lazebnik, and Yunzhu Li. A real-to-sim-to-real approach to robotic manipulation with vlm-generated iterative keypoint rewards. *ArXiv*, abs/2502.08643, 2025.

[37] Georgios Pavlakos, Dandan Shan, Ilija Radosavovic, Angjoo Kanazawa, David F. Fouhey, and Jitendra Malik. Reconstructing hands in 3d with transformers. *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9826–9836, 2023.

[38] Polycam. Polycam, 2020. URL https://poly.cam. Accessed: 2025-04-30.

[39] Juntao Ren, Priya Sundaresan, Dorsa Sadigh, Sanjiban Choudhury, and Jeannette Bohg. Motion tracks: A unified representation for human-robot transfer in few-shot imitation learning. *ArXiv*, abs/2501.06994, 2025.

[40] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *ArXiv*, abs/1707.06347, 2017.

[41] Lin Shao, Toki Migimatsu, Qiang Zhang, Karen Yang, and Jeannette Bohg. Concept2robot: Learning manipulation concepts from instructions and human demonstrations. volume 40, pages 1419 – 1434, 2020.

[42] Kenneth Shaw, Shikhar Bahl, and Deepak Pathak. Videodex: Learning dexterity from internet videos. In *Conference on Robot Learning*, 2022.

[43] Aravind Sivakumar, Kenneth Shaw, and Deepak Pathak. Robotic telekinesis: Learning a robotic hand imitator by watching humans on youtube. 2022.

[44] Mª Mercedes Lopez Torné, Anthony Simeonov, Zechu Li, April Chan, Tao Chen, Abhishek Gupta, and Pulkit Agrawal. Reconciling reality through simulation: A real-to-sim-to-real approach for robust manipulation. *ArXiv*, abs/2403.03949, 2024.

[45] Aäron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *ArXiv*, abs/1807.03748, 2018.

[46] Pietro Vitiello, Kamil Dreczkowski, and Edward Johns. One-shot imitation learning: A pose estimation perspective. In *Conference on Robot Learning*, 2023.

[47] Chen Wang, Linxi Fan, Jiankai Sun, Ruohan Zhang, Li Fei-Fei, Danfei Xu, Yuke Zhu, and Anima Anandkumar. Mimicplay: Long-horizon imitation learning by watching human play. 2023.

[48] Bowen Wen, Wei Yang, Jan Kautz, and Stanley T. Birchfield. Foundationpose: Unified 6d pose estimation and tracking of novel objects. *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17868–17879, 2023.

[49] Philipp Wu, Yide Shentu, Zhongke Yi, Xingyu Lin, and P. Abbeel. Gello: A general, low-cost, and intuitive teleoperation framework for robot manipulators. *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 12156–12163, 2023.

[50] Hongchi Xia, Entong Su, Marius Memmel, Arhan Jain, Raymond Yu, Numfor Mbiziwo-Tiapo, Ali Farhadi, Abhishek Gupta, Shenlong Wang, and Wei-Chiu Ma. Drawer: Digital reconstruction and articulation with environment realism. 2025.

[51] Hongchi Xia, Entong Su, Marius Memmel, Arhan Jain, Raymond Yu, Numfor Mbiziwo-Tiapo, Ali Farhadi, Abhishek Gupta, Shenlong Wang, and Wei-Chiu Ma. Drawer: Digital reconstruction and articulation with environment realism. *arXiv preprint arXiv:2504.15278*, 2025.

[52] Jiale Xu, Weihao Cheng, Yiming Gao, Xintao Wang, Shenghua Gao, and Ying Shan. Instantmesh: Efficient 3d mesh generation from a single image with sparse-view large reconstruction models. *ArXiv*, abs/2404.07191, 2024. URL https://api.semanticscholar.org/CorpusID:269033473.

[53] Mengda Xu, Zhenjia Xu, Cheng Chi, Manuela Veloso, and Shuran Song. XSkill: Cross embodiment skill discovery. In *7th Annual Conference on Robot Learning*, 2023. URL https://openreview.net/forum?id=8L6pHd9aS6w.

[54] Jianglong Ye, Jiashun Wang, Binghao Huang, Yuzhe Qin, and Xiaolong Wang. Learning continuous grasping function with a dexterous hand from human demonstrations. volume 8, pages 2882–2889, 2022.

[55] Weirui Ye, Fangchen Liu, Zheng Ding, Yang Gao, Oleh Rybkin, and Pieter Abbeel. Video2policy: Scaling up manipulation tasks in simulation through internet videos. *ArXiv*, abs/2502.09886, 2025.

[56] Kevin Zakka, Andy Zeng, Peter R. Florence, Jonathan Tompson, Jeannette Bohg, and Debidatta Dwibedi. Xirl: Cross-embodiment inverse reinforcement learning. In *Conference on Robot Learning*, 2021.

[57] Tony Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. *ArXiv*, abs/2304.13705, 2023.

[58] Tony Z. Zhao, Jonathan Tompson, Danny Driess, Pete Florence, Kamyar Ghasemipour, Chelsea Finn, and Ayzaan Wahid. Aloha unleashed: A simple recipe for robot dexterity. In *Conference on Robot Learning*, 2024.

[59] Yifeng Zhu, Arisrei Lim, Peter Stone, and Yuke Zhu. Vision-based manipulation from single human video with open-world object graphs. *arXiv preprint arXiv:2405.20321*, 2024.

IV. APPENDIX

A. Robustness to Test-Time Changes

Image-conditioned policies are particularly sensitive to viewpoint bias, demanding additional data for each perspective. X-SIM overcomes this by leveraging simulation to render trajectories from any desired view, enabling efficient coverage. We evaluate this by collecting simulated rollouts from *Side* and *Frontal* camera views, and training policies with data from each view individually and jointly. As shown in Fig. 7, combining diverse viewpoints in simulation significantly improves generalization, even to unseen camera angles at test time.

Train → / Test ↓	Side	Frontal	Side & Frontal
Side	83.3%	23.3%	**96.7%**
Frontal	23.3%	76.7%	**80.0%**
Novel	33.3%	30.0%	**53.5%**

Fig. 7. We show that we can flexibly collect image-action data in simulation from multiple viewpoints (Side and Frontal) with X-SIM and train robust policies that generalize to novel viewpoints.

B. Related Work

Imitation Learning. Imitation learning, particularly behavior cloning (BC), is the dominant paradigm for training visuomotor robot policies. Recent algorithms like Diffusion Policy [10] and ACT [57] achieve state-of-the-art results by learning from expert demonstrations consisting of image-action pairs. However, these methods typically require collecting data via human teleoperation of the specific target robot, using kinesthetic teaching [9], wearable devices [19], or specialized control interfaces [29, 49, 58]. Recent efforts have attempted to build large robotic dataset across different robot embodiments [34, 24] leading to the development of foundation models [25, 18] for robotic control. Still, scaling up such datasets remains a significant challenge because of the heavy reliance on robot teleoperation. While UMI [11] proposes hand-held grippers to collect data without direct robot involvement, these demonstrations can be dynamically infeasible for robots and still require active collection in lab settings. In contrast, our approach bypasses the need for robot action data entirely by leveraging human videos to generate synthetic robot data.

Learning from Human Videos. The ease of collecting human videos has motivated interest in learning robot motion directly from them. Common strategies include retargeting hand motion [43, 42, 3, 54, 6], reducing the visual gap via inpainting [4, 27, 28], or using pretrained open-world vision models for constructing object-relative hand trajectories [59, 46, 31]. All of these methods rely on the robot's capability to match its end-effector with the human's hand positions, which often falls down in practice due to large embodiment differences. Hierarchical frameworks [47, 8, 5] learn high-level plans instead, while one-shot imitation methods [21, 20, 7] learn from prompt videos. These methods typically require human-robot paired data or self-supervised alignment from unpaired data [22, 53]. In either case, a common limitation among these methods is the need for robot teleoperation data to guide low-level control [39]. RL provides an alternative, using video [56] similarity, language matching [41] or object tracking [35] for rewards, but suffers from the sim-to-real gap. Cross-embodiment RL [26, 16] methods that have been deployed on real robots require object tracking at test-time which can be brittle to noisy observations. Instead, we leverage a real-to-sim-to-real pipeline to directly transfer image-based policies from simulation.

Real-to-Sim-to-Real. Advances in 3D computer vision have enabled the development of photorealistic, physically accurate simulations from real-world data. Recent works increasingly use real-to-sim methods to learn robot behaviors in simulation. For instance, RialTo [44] trains RL policies in simulation to improve policy robustness, using point cloud inputs for real-world deployment. ResiP [2] learns residual actions in simulation starting from an image-based policy trained in the real world. However, both these approaches still require real-world robot data collection. To directly learn actions, motion planners are used in simulation but deployed open-loop in the real world [36, 23]. More recently, real-to-sim-to-real has been applied to learn from human videos [55, 14]. However, Video2Policy [55] only extracts the initial and final object states from human videos, and relies on object segmentation masks at test time for policy transfer. Human2Sim2Robot [14] defines rewards for RL using object state tracking from videos, but does not use a photo-realistic simulation. However, real-world deployment additionally requires object tracking at test time. RL training also requires tracking human hand trajectories for guiding the policy, and is applied only to dexterous hands with minimal embodiment gap. Our work offers distinct advantages over these methods: (a) we bypass the need for robot teleoperation data and human hand tracking for RL training, and (b) we transfer image-based policies from simulation to the real world using environment randomization and domain adaptation methods.

C. Limitations

In this paper, we chose to maximize the ability of the real-to-sim pipeline by making simplifying assumptions, while still maintaining the input/output contract (images to actions) that is most practical to deploy in unstructured environments. This is because the focus of the paper is to show the effectiveness of image-based policy transfer given ideal real-to-sim transfer. However, we acknowledge that while X-SIM provides an effective approach for learning robot policies from human videos, its application to unstructured, in-the-wild internet videos remains an open challenge. Below, we outline key assumptions that limit X-SIM's current ability to move towards this broader vision and suggest pathways towards their solutions in the near future:

Requiring Object Meshes for Tracking. Our pipeline currently uses FoundationPose, which requires a 3D object mesh for tracking, limiting applicability to videos where we

either don't know or don't have the object mesh manipulated by the human. One way to extend this to internet videos is by estimating approximate meshes directly from using tools like InstantMesh [52]. Alternatively, object meshes can be retrieved from large 3D asset libraries [30], as shown in prior work on digital cousin generation [12], which suffices since simulation is only used for synthetic data generation.

Restricted to Rigid Object Manipulation. Our current pipeline relies on tracking object states through 6D poses, which limits it to rigid objects and excludes articulated or deformable items commonly seen in real-world tasks. For articulated objects like drawers or doors, recent vision research [50] has shown that visual priors and foundation models can be used to identify and track articulation parameters from RGB input. For deformable objects, emerging representations like particle-based models [1] offer promising avenues for capturing non-rigid dynamics. While these approaches are still maturing, our framework can continue to improve rigid manipulation skills, and its image-conditioned policies may complement existing models trained on separate data to handle deformables and articulations more effectively.

Environment Scan for Generating Simulation. X-SIM currently requires an explicit 3D scan of the environment to reconstruct the simulation scene, which limits its applicability to scenarios where such scans are unavailable. Recent works like St4RTrack [13] have shown that it is possible to generate both geometric and visual reconstructions directly from monocular human videos. While these methods typically rely on dynamic camera motion, many human video datasets—such as Ego4D [15]—naturally satisfy this condition, offering a viable path toward removing the explicit environment scanning requirement.

Estimating Physics Parameters from Vision Alone. In this work, we use default physics parameters—such as mass, friction, and stiffness—for simulation, rather than estimating them from the human video. However, recent approaches suggest viable paths forward: vision language models (e.g., GPT) can provide plausible physics guesses given object categories or visual context [51], and domain randomization can be applied around these estimates to build robustness. Additionally, while our proposed online sim-to-real calibration targets visual alignment, the same framework could be extended to iteratively adapt physical parameters by comparing real and simulated rollouts—enabling self-supervised refinement of both perception and dynamics.

D. Task Descriptions

We provide descriptions and visualizations (Fig. 8) of tasks we report results for in Fig. 2.

- `Mustard Place`: Pick up the Mustard bottle from the right side of the kitchen countertop and place it on the left side.
- `Corn in Basket`: Pick up the corn from the left side of the kitchen countertop and put it inside of the basket.
- `Shoe on Rack`: Pick up the left shoe and place it on top of the shoe rack, next to the right shoe.
- `Letter Arrange`: Move the letter *'I'* next to the letter *'A'* so that they are aligned.
- `Mug Insert`: Insert the mug's handle onto the holder.

E. Real-to-Sim Scans

In order to transfer our environments and objects into simulation, we employ 2D Gaussian Splatting [17]. We take videos (multi-view images) of the environment for < 2 minutes, which are supplied as input to the module to get a photo-realistic 3D reconstruction of the scene. Individual objects to be tracked are scanned with Polycam [38], a phone app, with a similar procedure in < 1 minute per object. The environment and objects are scaled manually to the correct size before being transferred into simulation, though alternate calibration methods could be used to automate this process.

F. RL Training Details

1) PPO Hyperparameters: We provide details of hyperparameters (Table I) used for training privileged-state PPO [40] policies in simulation.

TABLE I
PPO HYPERPARAMETERS

Hyperparameter	Value
Learning rate	3×10^{-4}
Discount factor (γ)	0.8
GAE parameter (λ)	0.9
Clipping parameter (ϵ)	0.2
Value function coefficient	0.5
Entropy coefficient	0.0
Target KL divergence	0.1
Maximum gradient norm	0.5
Minibatch size	9,600
Number of parallel environments	1,024
Actor network	MLP (state dim $\rightarrow$ 256 $\rightarrow$ 256 $\rightarrow$ 256 $\rightarrow$ action dim)
Critic network	MLP (state dim $\rightarrow$ 256 $\rightarrow$ 256 $\rightarrow$ 256 $\rightarrow$ 1)
Activation function	Tanh
Optimizer	Adam
Adam epsilon	1×10^{-5}

2) Simulation State Space: The state-based observation space in simulation consists of the following components:

TABLE II
OBSERVATION SPACE COMPONENTS

Component	Description
`ee_pose`	End-effector pose (position and orientation)
`gripper_width`	Gripper opening width
`achieved_goal`	Current object poses
`desired_goal`	Target waypoint poses for objects
`goal_position_diff`	Position difference between current and target poses
`goal_rotation_diff`	Angular difference between current and target orientations
`is_grasped`	Binary object grasp status (if applicable)

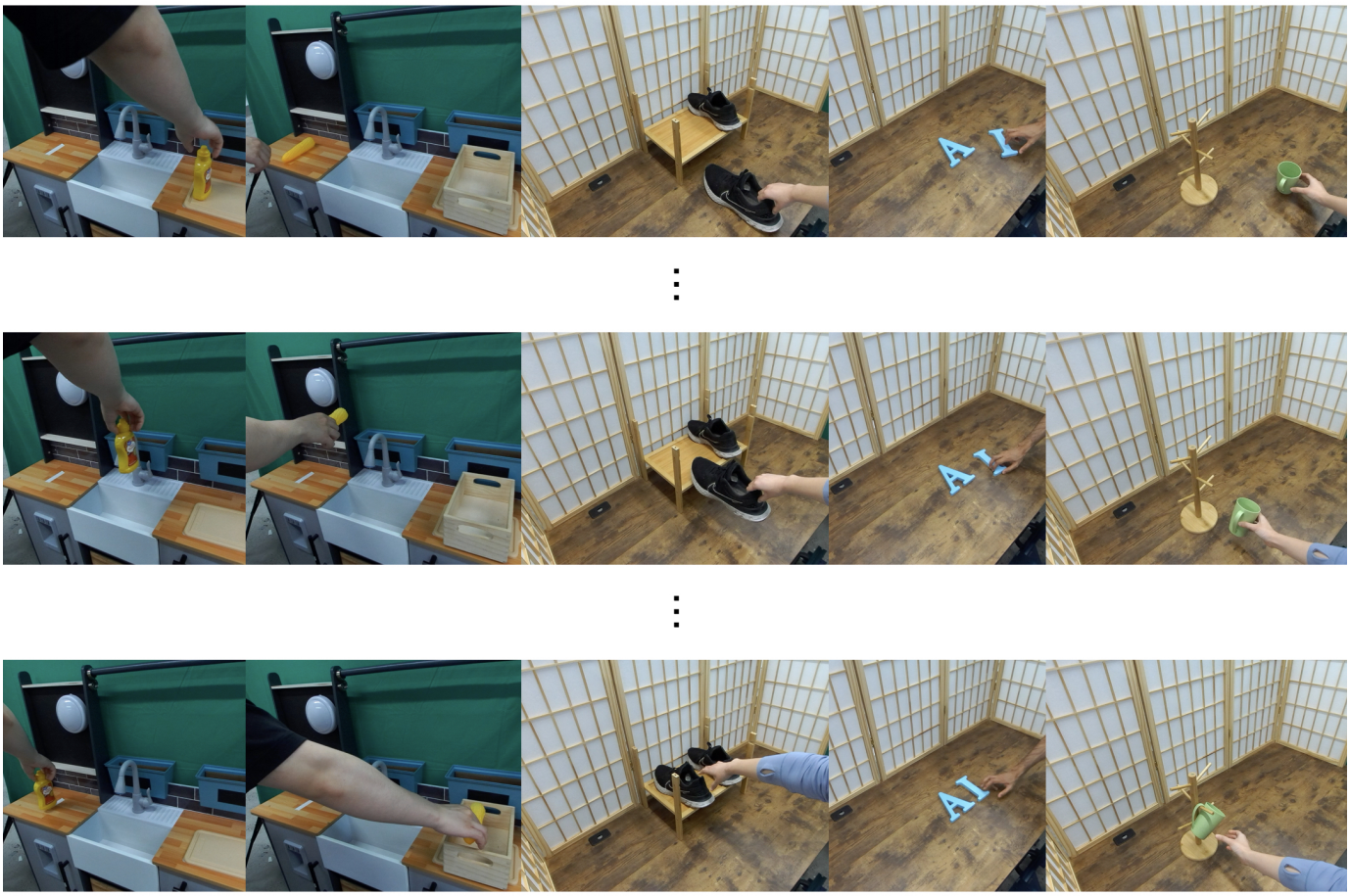


Fig. 8. Visualization of tasks that we report results for in Fig. 2

3) Reward Function Formulation: We provide the complete object-centric reward function implementation proposed in Sec. II:

a) Approach Reward.: The r_{approach} component encourages the agent to approach the target object with:

$$r_{\text{approach}} = (1 - \tanh(k d_{\text{obj}})) \tag{3}$$

where d_{obj} is the distance between the end-effector and the current target object and k is a constant scaling factor.

b) Goal Reward.: r_{goal} penalizes positional and rotational deviations from the target state:

$$\begin{aligned} r_{\text{goal}} = &\left(1 - \tanh(\alpha_d \cdot d_{\text{pos}}(s_H^B, s_R^t))\right) \\ &+ \left(1 - \tanh(\alpha_\theta \cdot d_{\text{rot}}(s_H^B, s_R^t))\right) \\ &+ 2 i_{\text{waypoint}} \end{aligned} \tag{4}$$

where $d_{\text{pos}}(\cdot)$ measures the Euclidean distance, and $d_{\text{rot}}(\cdot)$ computes the quaternion angular difference, α_d and α_θ are scaling factors for each waypoint automatically computed from the demonstration, and i_{waypoint} is the current waypoint index to serve as a bonus for progressing through the task. Note that the `desired_goal` in the observation is updated when the current goal is reached within an ϵ threshold, and in practice we sample N object waypoints from the human video to summarize the demonstration.

The goal reward also has additional terms: r_{static} encourages stability of the robot when objects are correctly positioned, r_{success} provides a $+1$ bonus upon task completion (objects are placed in their goal configuration), and r_{grasp} is an optional binary reward to encourage grasps for non-prehensile tasks.

c) Complete Reward.: The final reward is $r_{\text{obj}} = r_{\text{approach}} + r_{\text{goal}}$.

G. Image-Conditioned Policy Training Details

1) Synthetic Data Collection: We provide details on randomization parameters (Table III) used when collecting synthetic data for $D_{\text{synthetic}}$ (Sec. II).

For each task, we collect 500 visuomotor demonstrations in simulation by applying these randomization parameters.

2) Image-Conditioned Diffusion Policy Training: We provide hyperparameters (Table IV) for training Diffusion Policies [10], where input is simply an image of the current state and output is 7-dimensional delta actions in end-effector space (3 position actions, 3 rotation actions, 1 gripper action).

H. Ablation Visualizations

1) Data Efficiency: We provide visualizations (Fig. 10) of the initial state distribution of the Mustard bottle as training

TABLE III
ENVIRONMENT RANDOMIZATION PARAMETERS

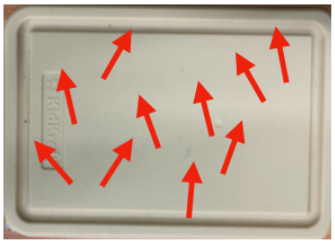
Parameter	**Value**
Object Randomization	
Initial pose position noise (XY)	± 0.025 m
Initial pose rotation noise	$\pm \pi/8$ rad
Robot Randomization	
Initial robot joint angle noise	± 0.02 rad
Camera and Lighting (Evaluation)	
Camera position variation	± 0.03 m
Camera target position variation	± 0.03 m
Lighting configurations	4 presets

TABLE IV
DIFFUSION POLICY TRAINING HYPERPARAMETERS

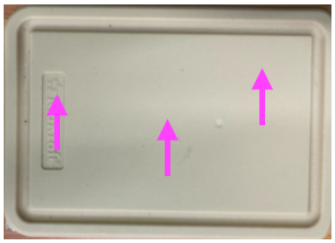
Parameter	**Value**
Diffusion timesteps (training)	100
Diffusion timesteps (inference)	10
Backbone CNN	ResNet18
Image size	$960 \times 720 \rightarrow 96 \times 96$
Image feature dimension	512
Diffusion step embedding dimension	128
Kernel size	5
Normalization layer	GNN
Action horizon	2
Prediction horizon	8
Shift padding	6
Batch size	64
Learning rate	1×10^{-4}
Weight decay	1×10^{-6}
Gradient clipping	5.0
EMA decay rate	0.01
Action prediction loss weight	1
Auto-calibration loss weight	0.1 (if applicable)

input for the data efficiency ablation in Sec. III-C. The robot teleoperation data takes 10 minutes to collect, while the human videos take 1 minute to collect. In simulation, the starting poses of the object are perturbed to enable robustness of the RL policy and diversity in during synthetic data collection. The evaluation distribution is across the cutting board.

2) Viewpoint Changes: We provide visualizations (Fig. 10) of the three different viewpoints that we study at train/test time in Sec. IV-A.



Robot Teleoperation Data - 10min



Human Video Data - 1min
(perturbed in sim)

Fig. 9. Visualization of training states for results in Fig. 6

Side View

Frontal View

Novel View

Fig. 10. Visualization of viewpoints for results in Fig. 7