ROSE: Reconstructing Objects, Scenes, and Trajectories from Casual Videos for Robotic Manipulation

Anonymous Author(s)
Affiliation
Address
email

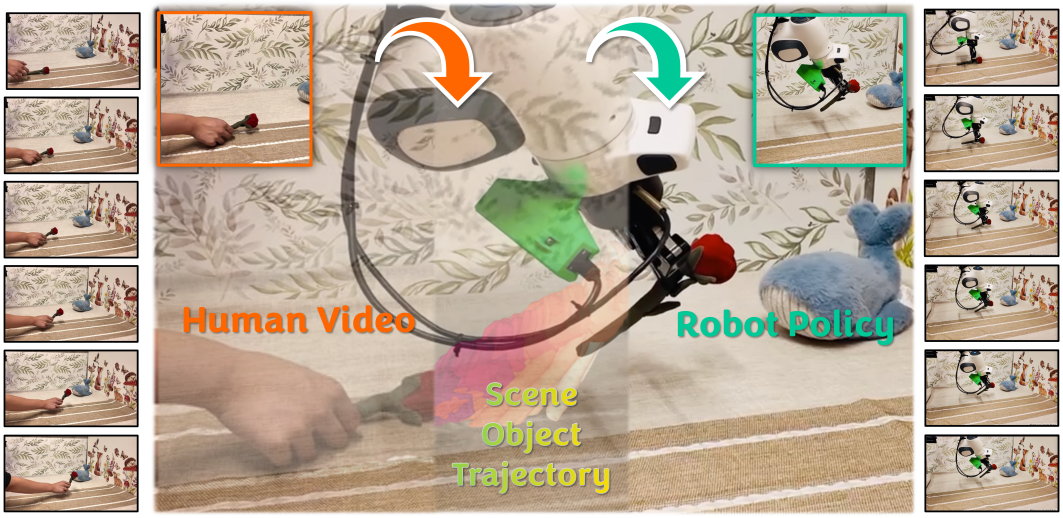

Figure 1: **Demonstration of ROSE: Reconstructing Objects, Scenes, and Trajectories from Casual Videos for Robotic Manipulation.** Left: Human demonstration from a monocular casual video. Middle: Reconstruction of objects, scenes, and trajectories. Right: Robotic manipulation learned from the demonstration.

Abstract: In this paper, we build a real-to-sim-to-real (Real2Sim2Real) system for robot manipulation policy learning from casual human videos. We propose a new framework, ROSE, that directly leverages casual videos to reconstruct simulator-ready assets, including objects, scenes, and object trajectories, for training manipulation policies with reinforcement learning in the simulation. Unlike existing real-to-sim pipelines that rely on specialized equipment or time-consuming and labor-intensive human annotation, our pipeline is equipment-agnostic and fully automated, facilitating data collection scalability. From casual monocular videos, ROSE enables the direct reconstruction of metric-scale scenes, objects, and object trajectories in the same gravity-calibrated coordinate for robotic data collection in the simulator. With ROSE, we curate a dataset of simulator-ready scenes from casual videos from our own capture and the Internet, and create a benchmark for real-to-sim evaluation. Across a diverse suite of manipulation tasks, ROSE outperforms the existing baselines, laying the groundwork for scalable robotic data collection and achieving efficient Real2Sim2Real deployment.

1 Introduction

This paper develops a real-to-sim-to-real system that enables robot manipulation policy learning from casual human videos. Human videos are useful for learning complex robot manipulation skills

Submitted to the 9th Conference on Robot Learning (CoRL 2025). Do not distribute.

Table 1: **Comparison with existing real-to-sim pipelines**. Scene mesh: 3D collision mesh of the scene. Object Mesh: 3D collision mesh of objects. Object Traj.: The 6-DoF pose of objects to be manipulated. Gravity Dir: The gravity direction of the reconstructed scene and objects. Metric Scale: If the reconstructed scene is in metric space (cm). World Coord.: If the reconstructed scene is in the world coordinate. Automation: It is a fully automated pipeline or requires human annotations (*e.g.*, RialTo [1] needs expert human annotation using GUI tools). O: Unknown.

	Characteristics							
Method	Scene Mesh	Object Mesh	Object Traj.	Gravity Dir.	Metric Scale	World Coord.	Automation	**Input / Platform**
RialTo [1]	✓	✓	✓	✓	✓	✓	✗	RGB-Video
Video2Policy [2]	✗	✓	✓	✗	✓	✗	✓	MV-imgs / LiDAR
RL-GSBridge [3]	✗	✓	✗	✗	✗	✓	✗	MV-imgs
SplatSim [4]	✓	✓	✓	✗	✗	✓	✗	RGB-D
ReBot [5]	✗	✗	✓	✗	✓	✓	✓	Video / Mesh
Digital Cousins [6]	O	O	✗	✗	O	✓	✓	Image
Chen et al. [7]	✗	✗	✗	✗	✗	✗	✓	Mesh / Trajectory
URDFormer [8]	✓	✓	✗	✗	✗	✓	✓	MV-imgs
Ditto In the House [9]	✓	✓	✗	✗	✗	✓	O	Image / Interaction
Ours	✓	✓	✓	✓	✓	✓	✓	RGB-Video

efficiently, the following three knowledge must be utilized: **i)** *Objects*: Different objects pose different manipulation strategies according to specific shapes, sizes, textures, *etc* (*e.g.*, grasping a goblet *vs* grasping a box), which is valuable for scaling up object priors. **ii)** *Scenes*: The scene plays a vital role in manipulation that requires scene awareness. *e.g.*, inserting a book into the bookshelf requires an understanding of both the book and the bookshelf context. **iii)** *Trajectories*: The 6-DoF object trajectories encode the most abundant task-solving information, guiding both traditional motion planning and policy training.

To fully leverage the aforementioned knowledge, one could replicate the same scene and objects in the real world and train a manipulation policy via reinforcement learning (RL) [10, 11] or imitation learning [12], as training on extensive robotic data can endow policies (*e.g.*, vision-language-action models [13–15]) with broad generalization across tasks. However, this is typically infeasible and extremely inefficient – experimenting with robots in the real world with RL has safety risks, and replicating the exact same environment as human demonstrations for robotic data collection is difficult, especially when using *casual* human videos from the Internet. Moreover, learning robust behaviors for novel situations often demands either numerous human demonstrations or risky trial-and-error on physical hardware. Consequently, imitation learning alone may struggle to generalize, while direct reinforcement learning in the real world may require impractical amounts of unsafe interaction.

Policy training and testing in simulation followed by sim-to-real (Sim2Real) policy transfer has thus emerged as an effective alternative, allowing robots to practice skills and explore failure with RL without real-world consequences [16, 17]. This motivates a promising solution – transferring the high-fidelity human video that encodes the three key knowledge to the simulation – a Real2Sim2Real approach. Modern 3D scene and object reconstruction techniques can be utilized to turn monocular RGB videos into detailed 3D models of the environment, dramatically simplifying virtual scene creation. For example, neural implicit representations like NeRF [18] and related methods are able to capture high-fidelity object geometry and textures from casual camera scans. Hence, one can rapidly produce photorealistic, physics-ready virtual replicas of real scenes that support *interaction* in a simulator through this real-to-sim (Real2Sim) procedure.

Several recent systems demonstrate the power of this approach. As depicted in Tab. 1, existing methods only transfer partial knowledge [2, 3, 5, 7–9, 19, 20], lack physics alignment like gravity [2, 3, 5, 8, 9, 20], or require significant human annotation efforts [1, 4]. For example, Torne et al. [1]

construct a digital twin of a real manipulation scene on-the-fly from a few camera scans, then use it to fine-tune an imitation-learned policy via reinforcement learning in simulation, but extensive human labor is required for annotating the scene information. Similarly, Fang et al. [5] replay real robot manipulation trajectories in a simulator to diversify object interactions, and then integrate the simulated motions into real image backgrounds to synthesize new realistic training videos. This requires a time-consuming collection of robotic data, and this approach completely overlooks the value in scenes and objects. As a result, these studies have not fully realized the potential value of human videos, leaving an automatic Real2Sim2Real pipeline that satisfies all the features deficient.

In this paper, we propose **R**econstruction of **O**bjects, **S**cene and Traj**E**ctories (ROSE), a system for Real2Sim 3D reconstruction from monocular videos to facilitate robotic manipulation. Given only a single RGB camera moving through a real scene, ROSE automatically builds an interactive 3D simulation of that scene, reconstructing both the geometry and appearance of objects and surfaces. The resulting simulation (a "digital twin" of the scene) can be used to train and evaluate manipulation strategies in a safe and scalable manner. By eliminating much of the manual effort required to create detailed simulated environments, ROSE aims to enable robots to learn and test manipulation policies in faithful virtual replicas of real-world settings, then execute them reliably in the physical world. We collect a large-scale dataset comprising diverse scenes, objects, trajectories, and physically plausible robotic actions for task completion. The dataset includes more than 30 scenes, 50 objects, 600 trajectories, and 3,500 robot action samples.

2 Related Work

2.1 Sim-to-Real RL Policy Transfer

Training robot policies with RL in simulation, followed by a sim-to-real (Sim2Real) policy transfer, has become one of the most successful robot learning strategies in wide applications, such as locomotion [21–25], loco-manipulation [26–29], dexterous manipulation [30, 31], *etc*. One advantage of such Sim2Real RL training lies in the low-cost, safe, and more potential in improving generalization through domain / dynamic randosmizations [16, 32], making it a widely adopted alternative to collecting real-world data that is typically time-consuming and labor-intensive. However, such a low-cost and safe simulation training alternative may bring a Sim2Real gap that makes it hard for the Sim2Real policy transfer. To address this issue, a lot of works have been proposed to mitigate the gap, *e.g.*, curriculum learning of Sim2Real constraints [24, 33–35], teacher-student distillation of privileged information like object states or environment extrinsics [23, 35, 36], 3D awareness [37–41], and perception augmentation / randomization [42–47].

2.2 Real-to-Sim Dynamic Scene and Object Transfer

Recently, a lot of efforts in 3D vision have been devoted to creating simulated twins of the real-world scenes / objects from 2D videos [18, 48, 49], which is critical in enriching operating environments when training robot policies in simulation. Generally, transferring real-world scene videos to the 3D simulation that is useful for robot learning involves three key components: i) 3D scene geometry, ii) 3D object geometry, and iii) object dynamics, which requires two key techniques as follows.

Dynamic 3D Scene Reconstruction from 2D focuses on recovering the appearance and geometry of scenes from 2D images or videos. Earlier methods [49–53] typically rely on dense multiview capture and require significant computational resources to reconstruct dynamic scenes, often using NeRF-based [54] or 3D Gaussian Splatting [55] representations that evolve over time. More recently, with advances in deep multiview stereo [56, 57] and monocular depth estimation [58, 59], a new line of work has emerged that better captures the geometry of dynamic scenes from casual inputs. Notably, approaches such as MegaSaM [60], MonST3R [61], and CUT3R [62] demonstrate robust and efficient dynamic 3D reconstruction from casually captured monocular videos. These methods mark a significant step towards scalable, large-scale scene reconstruction and asset creation for downstream applications like robotics.

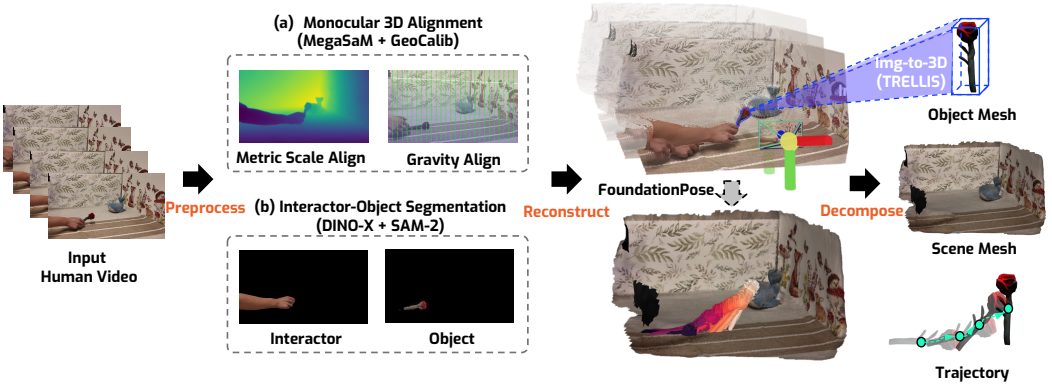

Figure 2: **ROSE Real2Sim pipeline illustration.** (a) We leverage MegaSaM[60] and GeoCalib[68] to reconstruct scene point cloud in the metric-scale and gravity-align world coordinates. (b) We further use SAM-2[69] and DINO-X[70] to detect and track interactor and object mask from videos.

3D Object Dynamics from 2D provides the object-level kinematic dynamics encoded as object spatial translation and orientation in 3D, offering valuable priors that help both traditional motion planning methods and learning-based approaches. To capture such object dynamics, various object representations have been used as the policy tracking goal. For example, Bharadhwaj et al. [46] propose to use object and hand segmentation as proxy information, followed by a segmentation image conditioned policy that achieves better generalization.

Meanwhile, some works utilize the point-level flow map of objects or images as the point tracking objective and achieve great progress [63–66]. Different from relying on such proxy representations, our approach directly collects 6DoF trajectories through pose estimation [67], offering a scalable and efficient solution for acquiring high-quality motion data. Notably, a concurrent work, Video2Policy [2], also proposes to use 6DoF object trajectories as object dynamics. However, Video2Policy only reconstructs the object states and places objects on the same canonical tabletop in a specific robot frame. In contrast, our approach transfers both the dynamic scenes and the objects in the world coordinate, where the world frame reconstruction helps SLAM-based scene reconstruction [60].

3 ROSE – Reconstructing Object, Scene, and Trajectory

3.1 Object Reconstruction

Object Grounding. As shown in the Fig. 2, given the target object label obtained from user input or LLM inference, we scan the video frame-by-frame until the object is first detected by DINO-X [70]. The detected bounding box is then passed to the SAM-2 image predictor [69] to obtain the target object mask. This mask is registered as the initial label in the SAM-2 video predictor, which subsequently propagates the segmentation through the rest of the sequence, yielding per-frame object masks.

Object Mesh Reconstruction. Using the segmented masks, we leverage TRELLIS [71] to reconstruct the 3D mesh, which provides 3D reconstruction pipelines from both single image and multiview images. Since most manipulation videos are filmed from a single viewpoint, we select the first frame mask to reconstruct the 3D mesh. For highly occluded or feature unclear situation, we would use masks from multiple non-occuluded views to reconstruct.

3.2 Scene Reconstruction

Scene Point Cloud Reconstruction. For every video frame I_i, MegaSAM [60] supplies the camera intrinsics $\mathbf{K}_i$, the camera pose $\mathbf{G}_i = [\mathbf{R}_i|\mathbf{t}_i]$, and a relative depth map D_i^{rel}. We feed I_i to UniDepth [58] to obtain an absolute depth estimate D_i^{abs}. A global scale factor $\hat{\alpha}$ and offset $\hat{\beta}$ align D_i^{rel} to metric depth D_i^{align}. Each pixel u is back-projected with $\mathbf{K}_i$, $\mathbf{R}_i$, $\mathbf{t}_i$, and D_i^{align} to yield 3-D points, which we accumulate into a raw scene point cloud $\mathcal{P}$. Then we apply GeoCalib [68] on to the

first frame and obtain a gravity-align transformation. Then we apply this transformation to each of the following frame to ensure the scene is under the gravity-aligned coordinate.

Scene Mesh Reconstruction. The sparse, hole-ridden point cloud yielded by the previous stage is first densified with Neural Kernel Surface Reconstruction (NKSR) [72]; its *detail* hyper-parameter is tuned to close gaps while preserving fine geometry. To satisfy simulator requirements, namely orientability, 2-manifoldness, and self-intersection freedom, we subsequently apply an Alpha Wrapping procedure, [73] producing a watertight, validity-guaranteed surface. Finally, color is restored by a point-to-vertex transfer: each mesh vertex inherits the distance-weighted average RGB of its three nearest neighbours in the processed point cloud, yielding a textured, simulation-ready scene mesh.

3.3 Trajectory Reconstruction

Improved Foundation Pose. Given the segmentation masks, we apply FoundationPose [67] in a model-based setup to estimate the object's 6 DoF Pose Q. The model takes as input the trellis mesh, the camera intrinsic $\mathbb{K}$, and depth map d. We obtain $\mathbb{K}$ and d from MegaSAM [60].

For the masked region M, we compute the maximum pairwise 3D distance:

$$D_{\text{image}} = \max_{(i_1,j_1),(i_2,j_2)\in M} \|\mathbf{p}(i_1,j_1) - \mathbf{p}(i_2,j_2)\| \tag{1}$$

where $p_{i,j}$ denotes the 3D location in camera space. We similarly compute D_{mesh}, the maximum distance between mesh vertices, and define the initial scale ratio as $\rho = D_{\text{image}}/D_{\text{mesh}}$. Due to noise in depth, intrinsics, and occlusions, we refine the scale by searching within $[\frac{1}{\alpha}, \alpha]$ at step size t, selecting the scale that minimizes the IoU loss:

$$L_{\text{IoU}} = 1 - \frac{\sum_{i=1}^{N} \hat{m}_i m_i}{\sum_{i=1}^{N} (\hat{m}_i + m_i) - \hat{m}_i m_i},$$

where N is the number of pixels, $\hat{m}_i$ is the predicted mask value, and m_i is the ground-truth mask value at pixel i.

3.4 Robot Action Collection

Building on the object trajectories reconstructed by the pipeline described above, we further explain how we collect robotic action data to enable the object to follow the trajectory and complete the task. With the reconstructed scene and object, we first load them into the simulator. Given the object's motion, our goal is to control the robot to interact appropriately with the object and guide it along the desired trajectory. We primarily utilize two baseline approaches for diverse robotic action collection: motion planning-based and reinforcement learning-based methods.

Motion Planning. For the motion planning-based algorithm, we first predict an appropriate grasping pose for the object. Once a stable grasp is achieved, the robot follows the object's trajectory using end-effector control based on cuRobo[74]. If the object remains stable and the trajectory is successfully followed, a data sample is considered successfully collected. For this method, we only consider the parallel-jaw gripper setting. In detail, we use GSNet [75] to predict grasp poses based on the point cloud generated in the simulation. After executing a planned trajectory to successfully grasp the object, the robot then follows the trajectory obtained from our vision pipeline to collect valid data.

Reinforcement Learning. Although the motion planning-based method is efficient and easy to implement, it is not sufficient for all scenarios. For example, when using high-dimensional robotic hands, as opposed to simple parallel-jaw grippers, predicting an appropriate grasping pose becomes significantly more challenging. In such cases, reinforcement learning (RL) allows the robot to explore and learn effective grasping strategies to complete the task.

Our RL baseline consists of two stages: object grasping and object manipulation. In the first stage, we design a reward function composed of three terms: a reaching reward r_{reach}, a grasping reward r_{grasp}. In the second stage, we follow the object trajectory generated by our previous pipeline to

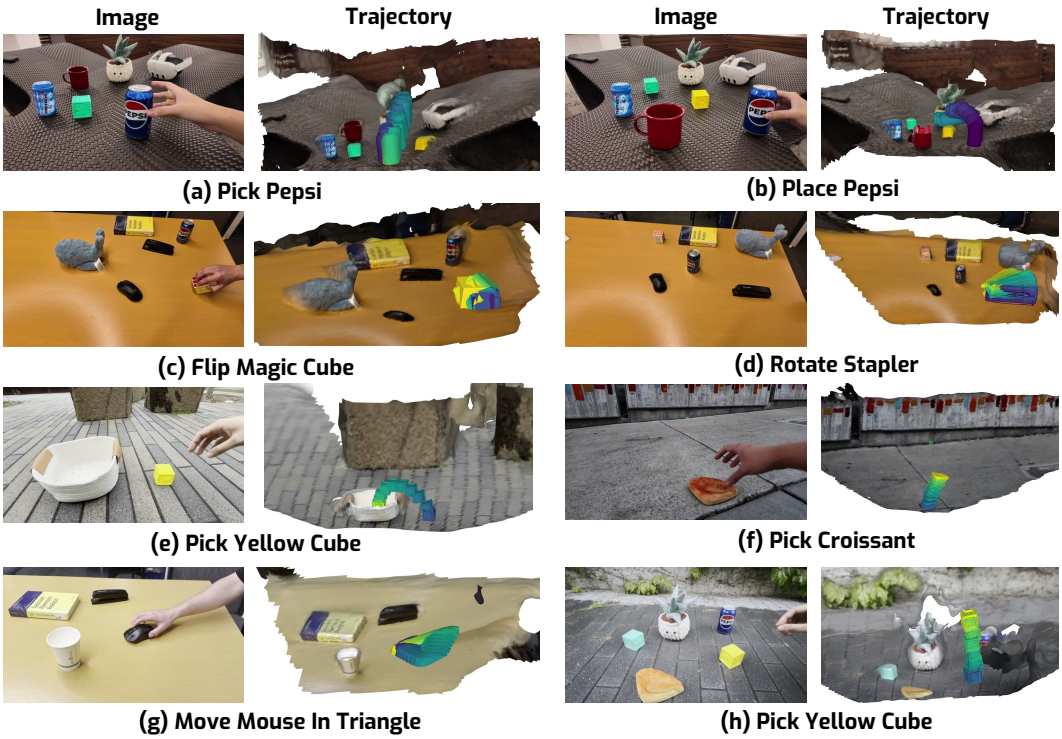

Figure 3: **Qualitative ROSE Real2Sim results.**

complete the task. To achieve this, we use CuRobo to control the end-effector and track the trajectory accurately.

It is also worth noting that we explored an end-to-end RL approach without the two-stage setting. While we carefully designed a reward function for trajectory following, we found that it was difficult for the policy to accurately replicate the generated motion, particularly in cases involving complex rotations. This limitation arises from the inherent nature of RL: since learning relies heavily on exploration, it is challenging for a policy to acquire precise trajectory-following behavior, especially when the robot is simultaneously required to grasp and manipulate the object.

3.5 Sim-to-real Transfer

With action data collected, we further train a model for sim-to-real transfer. A key advantage of our vision pipeline is its ability to generate high-quality, simulation-ready scene and object meshes, along with corresponding object trajectories. This enables fast and accurate robotic data collection in simulation. Using this data, we can leverage a high-quality renderer to produce realistic visual datasets. This allows us to train a vision-based robotic model capable of directly transferring to real-world scenarios.

4 Experiments

4.1 Experiment Setup

Our model is able to collect robotic data from diverse datasets from various sources, including outdoor, indoor environments. We benchmarked our real-to-sim method in RoboVerse[76] simulation environment and validated it in both simulation and real-world settings using the Franka arm and Unitree G1 humanoid robots.

4.2 Benchmark Construction

We construct a new benchmark to evaluate the fidelity of real-to-sim-to-real pipeline scene reconstructions from casual monocular videos as shown in Tab. 2. Because existing metrics treat scene layout, object shape, and motion separately, our benchmark fuses them into one holistic evaluation. It provides five simulated environments with full ground-truth geometry, appearance, and trajectories,

Task	Avg. Scene Chamfer Dist.	Object Chamfer Dist.	Translation APE	Rotation RPE	Translation RPE
Unstack	0.6211	0.02158	0.003242	3.724	0.001649
Place	0.6945	0.01060	0.02629	3.804	0.02269
Lift	0.6696	0.02786	0.02208	9.065	0.004374
Push	0.7513	0.01516	0.01086	4.229	0.002170
Rotate	0.6513	0.01394	0.008418	3.508	0.003301
Average	0.6776	0.01782	0.01418	4.866	0.006837

Table 2: **Benchmark comparison across tasks.** ROSE's performance metrics from our benchmark. Avg. Chamfer distance is computed for scene reconstructions, while object metrics include Chamfer distance, Absolute Pose Error (APE), and Relative Pose Error (RPE).

plus a casually captured video that serves as the pipeline's input. Evaluation uses four metrics: per-frame Chamfer distance between scene point clouds, Chamfer distance for object geometry, and APE/RPE (translation and rotation) for object trajectories. Scores are averaged across frames to yield stable measures. Together, these metrics reveal how well a method recovers both the static environment and the dynamics of the objects within it.

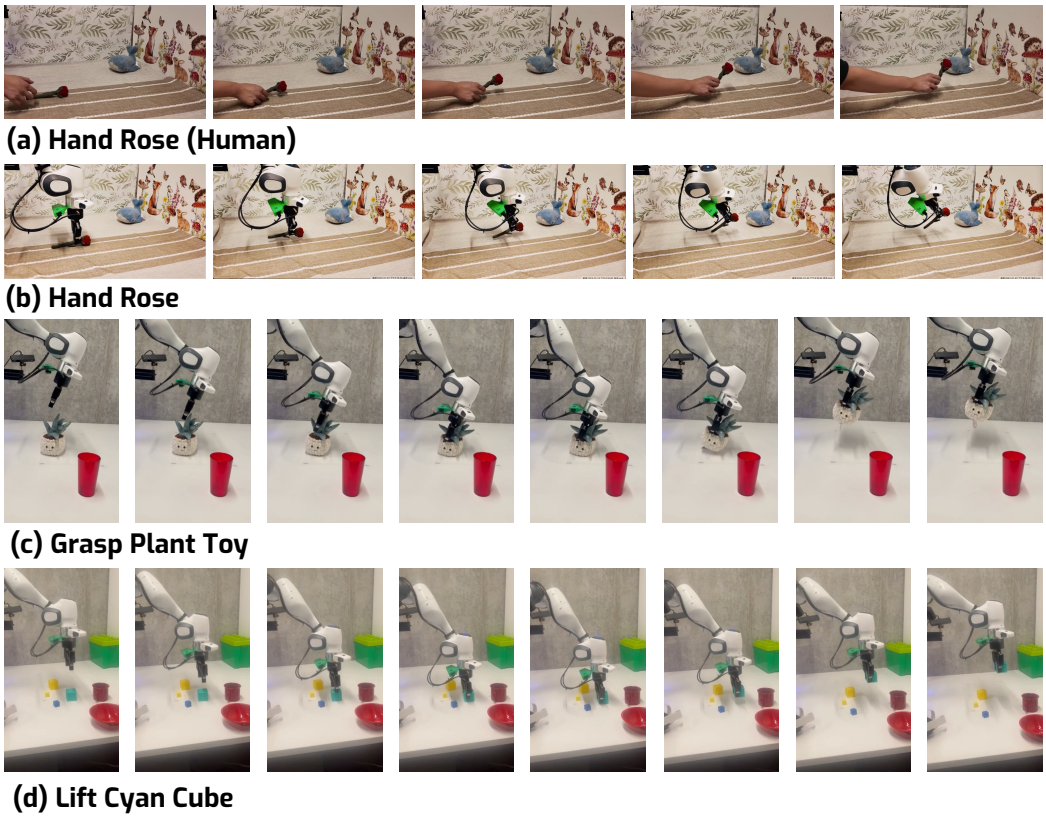

Figure 4: **Qualitative ROSE real-world results.**

4.3 Qualitative Results for Scene, Object and Trajectory Reconstruction

We present qualitative results on Fig. 3, demonstrating how our pipeline reconstructs geometrically accurate scene, object and object trajectory from casual videos to enable policy training.

4.4 Robotic Dataset Collection

Leveraging our scene, object, and trajectory reconstruction results, along with our robotic data collection pipeline, we construct a robotic manipulation dataset from monocular video. In the end, we collect **3.5k** valid robotic datasets with diverse task settings and environment variation.

4.4.1 Simulation Environment Setup

Leveraging the RoboVerse platform [76], we develop a pipeline for generating simulation environments. Specifically, we use a standardized configuration file to process scene layouts and object meshes. After loading the target robot into the simulation, we perform unit tests to ensure proper setup and collision-free initialization. We then follow the data collection pipelines to gather robotic manipulation data.

4.4.2 Manipulation Benchmark in Simulation

We establish a simulation benchmark to evaluate the performance of different robotic data collection methods. Specifically, we compare our proposed motion-planning-based approach and a two-stage reinforcement learning (RL) method against an end-to-end RL baseline. Our results show that the motion planning-based and two-stage RL methods perform differently across various settings—each demonstrating strengths in different scenarios. Comparing with strong baselines, including the concurrent work Video2Policy [2], Our method achieves the best performance on three out of four tasks as well as on the average score as shown in Tab. 3.

Method	PickPepsi	StackBlock	PlaceBowl	MoveTriangle	Average
End-to-End RL	**1.00**	0.00	**1.00**	0.00	0.50
Video2Policy [2]	0.00	0.00	0.40	0.00	0.10
Ours (Motion Planning)	0.80	**1.00**	0.40	0.80	0.75
Ours (Two-stage RL)	**1.00**	0.60	**1.00**	**1.00**	**0.90**

Table 3: **Task completion rate in simulation.**

4.5 Sim-to-Real Transfer

To validate the usefulness of our collected data, we conduct experiments to demonstrate the effectiveness of both the dataset and the trained policy.

4.5.1 Zero-shot Robotic Manipulation and Data Collection

We evaluate our collected robotic data and data collection pipeline in real-world settings. Specifically, we deploy the motion-planning-based method in a physical environment to assess its capability for zero-shot data collection and task execution using only a single demonstration. We test the data collection system across 13 different scenarios, achieving success in 11 of them—resulting in an **84.6%** success. The failure is primarily due to incorrect grasp poses and joint limit violations during motion planning.

4.5.2 Policy Sim-to-Real Transfer

We further train an RGB-based policy in simulation and demonstrate that, using the assets generated by our vision pipeline, the action data collected in simulation, and high-quality rendering based on RoboVerse [76], the resulting policy can zero-shot generalize to the real world.

5 Conclusion

We have introduced a fully automated *real–to–sim* framework that lifts casual monocular videos into simulator-ready assets—metric-scale, gravity-aligned scenes, watertight textured meshes, and object trajectories—without specialised sensors or manual annotation. Leveraging our pipeline, we delivers valid, photorealistic environments that satisfy modern simulators' geometric constraints. Experiments on a newly curated benchmark and diverse manipulation tasks demonstrate consistent improvements over prior Real2Sim baselines in scene fidelity, pose accuracy, and zero-shot policy transfer. By lowering the barrier to scalable data curation, our work lays a foundation for large-scale, task-agnostic robot learning and opens avenues toward richer video-driven Real2Sim2Real research.

6 Limitation

Our current pipeline focuses solely on the reconstruction and manipulation of rigid objects. Extending this approach to more challenging materials and deformable objects is left for future work. While our large-scale dataset, collected from casual videos, holds significant potential for pretraining a foundation model, exploring this direction is beyond the scope of this paper due to resource constraints.

References

[1] 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. 2

[2] W. Ye, F. Liu, Z. Ding, Y. Gao, O. Rybkin, and P. Abbeel. Video2policy: Scaling up manipulation tasks in simulation through internet videos. *arXiv preprint arXiv:2502.09886*, 2025. 2, 4, 8

[3] 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. *arXiv preprint arXiv:2409.20291*, 2024. 2

[4] 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, 2024. 2

[5] Y. Fang, Y. Yang, X. Zhu, K. Zheng, G. Bertasius, D. Szafir, and M. Ding. Rebot: Scaling robot learning with real-to-sim-to-real robotic video synthesis. *arXiv preprint arXiv:2503.14526*, 2025. 2, 3

[6] T. Dai, J. Wong, Y. Jiang, C. Wang, C. Gokmen, R. Zhang, J. Wu, and L. Fei-Fei. Automated creation of digital cousins for robust policy learning. In *Conference on Robot Learning (CoRL)*, 2024. 2

[7] Y. Chen, C. Wang, Y. Yang, and K. Liu. Object-centric dexterous manipulation from human motion data. In *8th Annual Conference on Robot Learning*. 2

[8] Z. Chen, A. Walsman, M. Memmel, K. Mo, A. Fang, K. Vemuri, A. Wu, D. Fox, and A. Gupta. Urdformer: A pipeline for constructing articulated simulation environments from real-world images, 2024. 2

[9] C.-C. Hsu, Z. Jiang, and Y. Zhu. Ditto in the house: Building articulation models of indoor scenes through interactive perception. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2023. 2

[10] S. Levine, P. Pastor, A. Krizhevsky, J. Ibarz, and D. Quillen. Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection. *The International journal of robotics research*, 37(4-5):421–436, 2018. 2

[11] J. Luo, C. Xu, J. Wu, and S. Levine. Precise and dexterous robotic manipulation via human-in-the-loop reinforcement learning. *arXiv preprint arXiv:2410.21845*, 2024. 2

[12] C. Chi, S. Feng, Y. Du, Z. Xu, E. Cousineau, B. Burchfiel, and S. Song. Diffusion policy: Visuomotor policy learning via action diffusion. In *Robotics: Science and Systems XIX, Daegu, Republic of Korea, July 10-14, 2023*, 2023. 2

[13] M. J. Kim, K. Pertsch, S. Karamcheti, T. Xiao, A. Balakrishna, S. Nair, R. Rafailov, E. Foster, G. Lam, P. Sanketi, et al. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024. 2

[14] K. Black, N. Brown, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai, L. Groom, K. Hausman, B. Ichter, et al. π_0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024.

[15] P. Intelligence, K. Black, N. Brown, J. Darpinian, K. Dhabalia, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai, et al. $\pi\_0.5$: a vision-language-action model with open-world generalization. *arXiv preprint arXiv:2504.16054*, 2025. 2

[16] X. B. Peng, M. Andrychowicz, W. Zaremba, and P. Abbeel. Sim-to-real transfer of robotic control with dynamics randomization. In *2018 IEEE International Conference on Robotics and Automation, ICRA 2018, Brisbane, Australia, May 21-25, 2018*, pages 1–8. IEEE, 2018. 2, 3

[17] J. Kober, J. A. Bagnell, and J. Peters. Reinforcement learning in robotics: A survey. *The International Journal of Robotics Research*, 32(11):1238–1274, 2013. 2

[18] 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. *Commun. ACM*, 65(1):99–106, 2022. 2, 3

[19] L. Wang, R. Guo, Q. Vuong, Y. Qin, H. Su, and H. Christensen. A real2sim2real method for robust object grasping with neural surface reconstruction. In *2023 IEEE 19th International Conference on Automation Science and Engineering (CASE)*, pages 1–8, 2023. 2

[20] T. Dai, J. Wong, Y. Jiang, C. Wang, C. Gokmen, R. Zhang, J. Wu, and L. Fei-Fei. Automated creation of digital cousins for robust policy learning. In *Conference on Robot Learning, 6-9 November 2024, Munich, Germany*, volume 270 of *Proceedings of Machine Learning Research*, pages 4912–4943. PMLR, 2024. 2

[21] J. Tan, T. Zhang, E. Coumans, A. Iscen, Y. Bai, D. Hafner, S. Bohez, and V. Vanhoucke. Sim-to-real: Learning agile locomotion for quadruped robots. In *Robotics: Science and Systems XIV, Carnegie Mellon University, Pittsburgh, Pennsylvania, USA, June 26-30, 2018*, 2018. 3

[22] H. Lai, W. Zhang, X. He, C. Yu, Z. Tian, Y. Yu, and J. Wang. Sim-to-real transfer for quadrupedal locomotion via terrain transformer. In *IEEE International Conference on Robotics and Automation, ICRA 2023, London, UK, May 29 - June 2, 2023*, pages 5141–5147. IEEE, 2023.

[23] A. Kumar, Z. Fu, D. Pathak, and J. Malik. RMA: rapid motor adaptation for legged robots. In *Robotics: Science and Systems XVII, Virtual Event, July 12-16, 2021*, 2021. 3

[24] X. He, R. Dong, Z. Chen, and S. Gupta. Learning getting-up policies for real-world humanoid robots. In *Robotics: Science and Systems XXI, Los Angeles, California, June 21-25, 2025*, 2025. 3

[25] Z. Li, X. B. Peng, P. Abbeel, S. Levine, G. Berseth, and K. Sreenath. Reinforcement learning for versatile, dynamic, and robust bipedal locomotion control. *The International Journal of Robotics Research*, page 02783649241285161, 2024. 3

[26] J. Siekmann, Y. Godse, A. Fern, and J. W. Hurst. Sim-to-real learning of all common bipedal gaits via periodic reward composition. In *IEEE International Conference on Robotics and Automation, ICRA 2021, Xi'an, China, May 30 - June 5, 2021*, pages 7309–7315. IEEE, 2021. 3

[27] T. He, Z. Luo, X. He, W. Xiao, C. Zhang, W. Zhang, K. M. Kitani, C. Liu, and G. Shi. Omnih2o: Universal and dexterous human-to-humanoid whole-body teleoperation and learning. In *8th Annual Conference on Robot Learning*, 2024.

[28] Z. Fu, Q. Zhao, Q. Wu, G. Wetzstein, and C. Finn. Humanplus: Humanoid shadowing and imitation from humans. In *8th Annual Conference on Robot Learning*, 2024.

[29] Z. Fu, X. Cheng, and D. Pathak. Deep whole-body control: Learning a unified policy for manipulation and locomotion. In K. Liu, D. Kulic, and J. Ichnowski, editors, *Conference on Robot Learning, CoRL 2022, 14-18 December 2022, Auckland, New Zealand*, volume 205 of *Proceedings of Machine Learning Research*, pages 138–149. PMLR, 2022. 3

[30] H. Qi, A. Kumar, R. Calandra, Y. Ma, and J. Malik. In-hand object rotation via rapid motor adaptation. In *Conference on Robot Learning, CoRL 2022, 14-18 December 2022, Auckland, New Zealand*, volume 205 of *Proceedings of Machine Learning Research*, pages 1722–1732. PMLR, 2022. 3

[31] A. Handa, A. Allshire, V. Makoviychuk, A. Petrenko, R. Singh, J. Liu, D. Makoviichuk, K. V. Wyk, A. Zhurkevich, B. Sundaralingam, and Y. S. Narang. Dextreme: Transfer of agile in-hand manipulation from simulation to reality. In *IEEE International Conference on Robotics and Automation, ICRA 2023, London, UK, May 29 - June 2, 2023*, pages 5977–5984. IEEE, 2023. 3

[32] 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. In *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 23–30. IEEE, 2017. 3

[33] W. Liang, S. Wang, H.-J. Wang, O. Bastani, D. Jayaraman, and Y. J. Ma. Eurekaverse: Environment curriculum generation via large language models. *arXiv preprint arXiv:2411.01775*, 2024. 3

[34] J. Siekmann, Y. Godse, A. Fern, and J. W. Hurst. Sim-to-real learning of all common bipedal gaits via periodic reward composition. In *IEEE International Conference on Robotics and Automation, ICRA 2021, Xi'an, China, May 30 - June 5, 2021*, pages 7309–7315. IEEE, 2021.

[35] J. Lee, A. Dosovitskiy, D. Bellicoso, V. Tsounis, V. Koltun, and M. Hutter. Learning agile and dynamic motor skills for legged robots. *Sci. Robotics*, 4(26), 2019. 3

[36] L. Pinto, M. Andrychowicz, P. Welinder, W. Zaremba, and P. Abbeel. Asymmetric actor critic for image-based robot learning. In H. Kress-Gazit, S. S. Srinivasa, T. Howard, and N. Atanasov, editors, *Robotics: Science and Systems XIV, Carnegie Mellon University, Pittsburgh, Pennsylvania, USA, June 26-30, 2018*, 2018. 3

[37] T. Ke, N. Gkanatsios, and K. Fragkiadaki. 3d diffuser actor: Policy diffusion with 3d scene representations. In P. Agrawal, O. Kroemer, and W. Burgard, editors, *Conference on Robot Learning, 6-9 November 2024, Munich, Germany*, volume 270 of *Proceedings of Machine Learning Research*, pages 1949–1974. PMLR, 2024. 3

[38] Y. Ze, G. Yan, Y. Wu, A. Macaluso, Y. Ge, J. Ye, N. Hansen, L. E. Li, and X. Wang. Gnfactor: Multi-task real robot learning with generalizable neural feature fields. In J. Tan, M. Toussaint, and K. Darvish, editors, *Conference on Robot Learning, CoRL 2023, 6-9 November 2023, Atlanta, GA, USA*, volume 229 of *Proceedings of Machine Learning Research*, pages 284–301. PMLR, 2023.

[39] T. Gervet, Z. Xian, N. Gkanatsios, and K. Fragkiadaki. Act3d: 3d feature field transformers for multi-task robotic manipulation. In J. Tan, M. Toussaint, and K. Darvish, editors, *Conference on Robot Learning, CoRL 2023, 6-9 November 2023, Atlanta, GA, USA*, volume 229 of *Proceedings of Machine Learning Research*, pages 3949–3965. PMLR, 2023.

[40] S. Suresh, H. Qi, T. Wu, T. Fan, L. Pineda, M. Lambeta, J. Malik, M. Kalakrishnan, R. Calandra, M. Kaess, et al. Neuralfeels with neural fields: Visuotactile perception for in-hand manipulation. *Science Robotics*, 9(96):eadl0628, 2024.

[41] N. Mishra, M. Sieb, P. Abbeel, and X. Chen. Closing the visual sim-to-real gap with object-composable nerfs. In *IEEE International Conference on Robotics and Automation, ICRA 2024, Yokohama, Japan, May 13-17, 2024*, pages 11202–11208. IEEE, 2024. 3

[42] J. Shi, Y. A, Y. Jin, D. Li, H. Niu, Z. Jin, and H. Wang. Asgrasp: Generalizable transparent object reconstruction and 6-dof grasp detection from RGB-D active stereo camera. In *IEEE International Conference on Robotics and Automation, ICRA 2024, Yokohama, Japan, May 13-17, 2024*, pages 5441–5447. IEEE, 2024. 3

[43] H. Fang, H.-S. Fang, S. Xu, and C. Lu. Transcg: A large-scale real-world dataset for transparent object depth completion and a grasping baseline. *IEEE Robotics and Automation Letters*, 7(3): 7383–7390, 2022.

[44] A. Agarwal, A. Kumar, J. Malik, and D. Pathak. Legged locomotion in challenging terrains using egocentric vision. In K. Liu, D. Kulic, and J. Ichnowski, editors, *Conference on Robot Learning, CoRL 2022, 14-18 December 2022, Auckland, New Zealand*, volume 205 of *Proceedings of Machine Learning Research*, pages 403–415. PMLR, 2022.

[45] A. Mandlekar, S. Nasiriany, B. Wen, I. Akinola, Y. S. Narang, L. Fan, Y. Zhu, and D. Fox. Mimicgen: A data generation system for scalable robot learning using human demonstrations. In *Conference on Robot Learning, CoRL 2023, 6-9 November 2023, Atlanta, GA, USA*, volume 229 of *Proceedings of Machine Learning Research*, pages 1820–1864. PMLR, 2023.

[46] H. Bharadhwaj, A. Gupta, V. Kumar, and S. Tulsiani. Towards generalizable zero-shot manipulation via translating human interaction plans. In *IEEE International Conference on Robotics and Automation, ICRA 2024, Yokohama, Japan, May 13-17, 2024*, pages 6904–6911. IEEE, 2024. 4

[47] Z. Q. Chen, S. C. Kiami, A. Gupta, and V. Kumar. Genaug: Retargeting behaviors to unseen situations via generative augmentation. In K. E. Bekris, K. Hauser, S. L. Herbert, and J. Yu, editors, *Robotics: Science and Systems XIX, Daegu, Republic of Korea, July 10-14, 2023*, 2023. 3

[48] Z. Jiang, C. Hsu, and Y. Zhu. Ditto: Building digital twins of articulated objects from interaction. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 5606–5616. IEEE, 2022. 3

[49] T. Li, M. Slavcheva, M. Zollhöfer, S. Green, C. Lassner, C. Kim, T. Schmidt, S. Lovegrove, M. Goesele, R. A. Newcombe, and Z. Lv. Neural 3d video synthesis from multi-view video. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 5511–5521. IEEE, 2022. 3

[50] G. Wu, T. Yi, J. Fang, L. Xie, X. Zhang, W. Wei, W. Liu, Q. Tian, and X. Wang. 4d gaussian splatting for real-time dynamic scene rendering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 20310–20320, 2024.

[51] Z. Yang, X. Gao, W. Zhou, S. Jiao, Y. Zhang, and X. Jin. Deformable 3d gaussians for high-fidelity monocular dynamic scene reconstruction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 20331–20341, 2024.

[52] J. Lei, Y. Weng, A. Harley, L. Guibas, and K. Daniilidis. Mosca: Dynamic gaussian fusion from casual videos via 4d motion scaffolds. *arXiv preprint arXiv:2405.17421*, 2024.

[53] Q. Wang, V. Ye, H. Gao, J. Austin, Z. Li, and A. Kanazawa. Shape of motion: 4d reconstruction from a single video. *arXiv preprint arXiv:2407.13764*, 2024. 3

[54] 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. *Communications of the ACM*, 65(1):99–106, 2021. 3

[55] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023. 3

[56] S. Wang, V. Leroy, Y. Cabon, B. Chidlovskii, and J. Revaud. Dust3r: Geometric 3d vision made easy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20697–20709, 2024. 3

[57] V. Leroy, Y. Cabon, and J. Revaud. Grounding image matching in 3d with mast3r. In *European Conference on Computer Vision*, pages 71–91. Springer, 2024. 3

[58] L. Piccinelli, C. Sakaridis, Y.-H. Yang, M. Segu, S. Li, W. Abbeloos, and L. Van Gool. Unidepthv2: Universal monocular metric depth estimation made simpler. *arXiv preprint arXiv:2502.20110*, 2025. 3, 4, 17

[59] L. Yang, B. Kang, Z. Huang, Z. Zhao, X. Xu, J. Feng, and H. Zhao. Depth anything v2. *Advances in Neural Information Processing Systems*, 37:21875–21911, 2024. 3

[60] Z. Li, R. Tucker, F. Cole, Q. Wang, L. Jin, V. Ye, A. Kanazawa, A. Holynski, and N. Snavely. Megasam: Accurate, fast and robust structure and motion from casual dynamic videos. In *arxiv*, 2024. 3, 4, 5, 17

[61] J. Zhang, C. Herrmann, J. Hur, V. Jampani, T. Darrell, F. Cole, D. Sun, and M.-H. Yang. Monst3r: A simple approach for estimating geometry in the presence of motion. *arXiv preprint arxiv:2410.03825*, 2024. 3

[62] Q. Wang, Y. Zhang, A. Holynski, A. A. Efros, and A. Kanazawa. Continuous 3d perception model with persistent state. In *CVPR*, 2025. 3

[63] H. Bharadhwaj, R. Mottaghi, A. Gupta, and S. Tulsiani. Track2act: Predicting point tracks from internet videos enables generalizable robot manipulation. In *European Conference on Computer Vision*, pages 306–324. Springer, 2024. 4

[64] C. Gao, H. Zhang, Z. Xu, Z. Cai, and L. Shao. Flip: Flow-centric generative planning for general-purpose manipulation tasks. *arXiv preprint arXiv:2412.08261*, 2024.

[65] M. Xu, Z. Xu, Y. Xu, C. Chi, G. Wetzstein, M. Veloso, and S. Song. Flow as the cross-domain manipulation interface. In P. Agrawal, O. Kroemer, and W. Burgard, editors, *Conference on Robot Learning, 6-9 November 2024, Munich, Germany*, volume 270 of *Proceedings of Machine Learning Research*, pages 2475–2499. PMLR, 2024.

[66] B. P. Duisterhof, Z. Mandi, Y. Yao, J.-W. Liu, J. Seidenschwarz, M. Z. Shou, D. Ramanan, S. Song, S. Birchfield, B. Wen, et al. Deformgs: Scene flow in highly deformable scenes for deformable object manipulation. *arXiv preprint arXiv:2312.00583*, 2023. 4

[67] B. Wen, W. Yang, J. Kautz, and S. Birchfield. FoundationPose: Unified 6d pose estimation and tracking of novel objects. In *CVPR*, 2024. 4, 5, 17

[68] A. Veicht, P.-E. Sarlin, P. Lindenberger, and M. Pollefeys. GeoCalib: Single-image Calibration with Geometric Optimization. In *ECCV*, 2024. 4, 17

[69] 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. 4, 17

[70] T. Ren, Y. Chen, Q. Jiang, Z. Zeng, Y. Xiong, W. Liu, Z. Ma, J. Shen, Y. Gao, X. Jiang, X. Chen, Z. Song, Y. Zhang, H. Huang, H. Gao, S. Liu, H. Zhang, F. Li, K. Yu, and L. Zhang. Dino-x: A unified vision model for open-world object detection and understanding, 2024. 4, 17

[71] J. Xiang, Z. Lv, S. Xu, Y. Deng, R. Wang, B. Zhang, D. Chen, X. Tong, and J. Yang. Structured 3d latents for scalable and versatile 3d generation. *arXiv preprint arXiv:2412.01506*, 2024. 4, 17

[72] J. Huang, Z. Gojcic, M. Atzmon, O. Litany, S. Fidler, and F. Williams. Neural kernel surface reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4369–4379, 2023. 5, 17

[73] C. Portaneri, M. Rouxel-Labbé, M. Hemmer, D. Cohen-Steiner, and P. Alliez. Alpha Wrapping with an Offset. *ACM Transactions on Graphics*, 41(4):1–22, June 2022. 5, 17

[74] B. Sundaralingam, S. K. S. Hari, A. Fishman, C. Garrett, K. V. Wyk, V. Blukis, A. Millane, H. Oleynikova, A. Handa, F. Ramos, N. Ratliff, and D. Fox. curobo: Parallelized collision-free minimum-jerk robot motion generation, 2023. 5, 18

[75] A. Mousavian, C. Eppner, and D. Fox. 6-dof graspnet: Variational grasp generation for object manipulation. In *International Conference on Computer Vision (ICCV)*, 2019. 5, 18

[76] H. Geng, F. Wang, S. Wei, Y. Li, B. Wang, B. An, C. T. Cheng, H. Lou, P. Li, Y.-J. Wang, Y. Liang, D. Goetting, C. Xu, H. Chen, Y. Qian, Y. Geng, J. Mao, W. Wan, M. Zhang, J. Lyu, S. Zhao, J. Zhang, J. Zhang, C. Zhao, H. Lu, Y. Ding, R. Gong, Y. Wang, Y. Kuang, R. Wu, B. Jia, C. Sferrazza, H. Dong, S. Huang, Y. Wang, J. Malik, and P. Abbeel. Roboverse: Towards a unified platform, dataset and benchmark for scalable and generalizable robot learning, 2025. URL https://arxiv.org/abs/2504.18904. 6, 8, 18, 20

[77] L. Yang, B. Kang, Z. Huang, Z. Zhao, X. Xu, J. Feng, and H. Zhao. Depth anything v2. *arXiv:2406.09414*, 2024. 17

[78] O. Mees, L. Hermann, E. Rosete-Beas, and W. Burgard. Calvin: A benchmark for language-conditioned policy learning for long-horizon robot manipulation tasks. 2022. 20

Supplementary Material

Contents

A Implementation Details

In this section, we introduce the implementation details of ROSE. To be specific, we present the details of the Real2Sim transfer of human videos, Sim2Real policy training, and real-world setup in Appendix A.1, Appendix A.2, and Appendix A.3, respectively.

A.1 Real-to-Sim Transfer

We use the videos with a resolution of 512×288 containing 60$\sim$300 frames in 30 FPS (spanning 2s to 10s). We first reconstruct the 3D point clouds by running MegaSaM [60], detailed as follows.

Point Cloud Reconstruction For every frame, we follow MegaSaM [60] to get the affine-invariant monocular disparity map with Depth-Anything V2 [77], a camera pose in the world coordinate, and the focal length estimation obtained with UniDepth V2 [58]. With this information, we align the disparity to the metric scale, which is converted to the pixel-aligned 3D point clouds coordinated in the world frame. Afterwards, we use GeoCalib [68] on the first frame to estimate the scene's gravity direction. We then rotate the camera rig so that the estimated gravity is to the negative of the z-axis in a right-hand coordinate system, resulting in gravity-aligned point clouds in the world frame. To further avoid residual artifacts, we apply edge dilation to remove colors at boundaries and depth-gradient pruning to remove point clouds with corresponding gravity exceeding a certain threshold of 0.8 for reducing depth discontinuities.

For every image, we use Ground-SAM-2 [69] with DINO-X-Track[70] to obtain binary masks of objects in the scene, used for object removal and reconstruction. For the scene point cloud, we fuse the point clouds by random sampling over the video to leverage complementary information to reduce the inaccurate point clouds caused by occlusions and partial scenes after object removal. The random sampling is used for balancing every frame's contribution. On average, the point cloud reconstruction would take 3 minutes to process a 5-second casual video.

3D Mesh Reconstruction For the object mesh, we use TRELLIS [71] to reconstruct the 3D mesh, and we set the α channel's threshold to 0.2 to help reconstruct dark objects. For the scene mesh, we use a three-stage method. In the first stage, we run Neural Kernel Surface Reconstruction (NKSR) [72] for surface fitting. To close the small gaps created by mask removal while preserving high-frequency geometry, we set the detail level to 0.4 and use a single MISE iteration. Afterwards, to make the mesh orientable, two-manifold, and self-intersection-free for simulator usage, we wrap the resulting mesh with CGAL alpha-wrapping algorithm [73]. The α value is set to 400. Finally, each mesh vertex v of $\mathcal{M}_{\text{wrap}}$ inherits the RGB value $c(\cdot)$:

$$c(v) = \sum_{p \in \mathcal{N}_k(v)} w(p, v)\, c(p), \quad w(p, v) \propto \frac{1}{\|p - v\|_2}.$$

This is the inverse-distance-weighted average of its k-nearest neighbours $\mathcal{N}_k(v)$ in the point cloud. In this work, we use $k = 3$ with a maximum distance of 5cm.

Improved Foundation Pose Foundation pose [67] requires that the scale of the mesh and the scale, unprotected from the depth map, be the same in order to ensure accurate pose estimation. Therefore, we align the trellis mesh $\mathbf{M}^{\text{trellis}}$ with the scene mesh $\mathbf{M}^{\text{scene}}$ by the following transformation:

$$\mathbf{M}^{\text{trellis-align}} = \mathbf{G}_0 \mathcal{Q} \mathbf{s} \mathbf{M}^{\text{trellis}} f, \tag{2}$$

where s is the scale factor, $\mathbf{G}_0$ is the camera pose, $\mathcal{Q}$ is the object pose, and f is the focal length.

For pose tracking, we begin by using the scale $\mathbf{s}$ estimated in the first frame to adjust the scale of $\mathbf{M}^{\text{trellis}}$. We then follow the approach outlined in FoundationPose [67]. The object pose $\mathcal{Q}_i$ is initialized using the previously estimated pose $\mathcal{Q}_{i-1}$, and the refinement network is applied to further refine $\mathcal{Q}_i$, yielding the final estimation of the object pose for the current frame.

A.2 Sim-to-Real Policy

Simulation Environment Setup We use RoboVerse [76] as our simulation platform to establish the data collection pipeline. Specifically, we adopt the IsaacGym branch for mesh loading, policy execution, and reinforcement learning, while leveraging the IsaacLab branch for high-fidelity rendering and vision-based policy training.

Simulation and control parameters follow the default settings provided by RoboVerse. For both objects and scenes, we set the friction coefficient to 0.5. To ensure accurate collision detection and reliable physics simulation, we apply convex decomposition to the scene geometry.

Robotic Data Collection based on Motion Planning We utilize GSNet [75] and cuRobo [74] as the core components of our motion planning pipeline, and conduct all simulations within the Isaac Gym environment. After reconstructing the scene (as described in Appendix A.1), we load the reconstructed environment and the target object mesh into the simulator. The object is represented using a high-resolution mesh, preserving geometric detail necessary for accurate grasp prediction and motion planning.

Using GSNet, we extract both the surface point cloud and a predicted grasp pose for the target object. These predictions are then used to initialize an inverse kinematics (IK) solver provided by cuRobo, which computes a feasible trajectory for the robot's end effector to reach the designated grasping configuration. During this process, we account for kinematic constraints, joint limits, and potential collisions in the environment.

Upon reaching the grasp pose, the robot executes a grasp based on a set of hand-crafted heuristics, which evaluate grasp stability using factors such as contact normals and finger placement. Once the grasp is completed, the robot follows a trajectory generated by a previously introduced motion prediction model, which guides the object to a specified goal position or task-specific location.

Robotic Data Collection based on Reinforcement Learning Our method adopts a two-stage policy for robotic manipulation. In the first stage, we employ a reinforcement learning (RL) approach to train a policy that guides the robot's end effector toward the object and performs a grasp once proximity is sufficiently close. To facilitate generalization across different grippers or robotic hands, we design a simple yet broadly applicable reward function. This reward encourages the end effector to reduce its distance to the target object and penalizes undesired motions, without relying on gripper-specific parameters, making it adaptable to a wide range of hardware configurations.

During deployment, we execute the learned grasping policy for a fixed horizon of 50 steps, under the assumption that a successful grasp is achieved by the end of this phase. In the second stage, we switch to a motion planning phase using cuRobo [74]. The robot follows a precomputed trajectory that guides the grasped object to its goal location or completes the assigned task. This two-stage setup decouples grasp acquisition from subsequent manipulation, allowing each component to be optimized independently while ensuring end-to-end effectiveness.

A.3 Real-World Experimental Setups

Franka Setting For our real-world experiments, we use a Franka Emika Panda robotic arm equipped with a Robotiq 2F-85 adaptive gripper. This setup provides a reliable and widely used platform for evaluating grasping and manipulation policies in physical environments. The robot is controlled via a high-level interface that integrates seamlessly with our planning and control pipeline.

To reconstruct the environment, we capture RGB-D data using both an iPhone 16 Pro and a DJI Osmo Pocket 3. These consumer-grade devices offer high-resolution color and depth sensing capabilities, allowing for efficient and accessible scene scanning.

B Real-to-sim Benchmark

B.1 benchmark evaluation metric details

Our proposed benchmark is based on three primary evaluations: scene reconstruction similarity, object reconstruction similarity, and object trajectory reconstruction.

Uniform Sampling. Uniform sampling extracts a point cloud from a mesh by taking N points drawn *i.i.d.* over the surface of the mesh $\mathbf{M}$. Unless stated otherwise, $N = 10{,}000$ in all our experiments.

Symmetric Chamfer Distance. Given two point clouds A and B, symmetric chamfer distance is defined as:

$$\text{ChamferDist}(A, B) = \frac{1}{|A|}\sum_{a\in A}\min_{b\in B}\|a-b\|_2^2 \; + \; \frac{1}{|B|}\sum_{b\in B}\min_{a\in A}\|b-a\|_2^2.$$

Scene Reconstruction Similarity. To evaluate scene reconstruction similarity, we first construct a point cloud $\mathbf{P}^{\text{scene}}$ through uniform sampling the reconstructed mesh. We then align $\mathbf{P}^{\text{scene}}$ with each frame's ground-truth point cloud $\mathbf{P}_i^{\text{gt-scene}} (i = 1, \ldots, F)$ by optimizing an SE(3) transformation to yield $\hat{\mathbf{P}}^{\text{scene}}$. Finally, we compute the average chamfer distance across the aligned frame point cloud $\hat{\mathbf{P}}^{\text{scene}}$ and the ground-truth scene point clouds.

$$\mathbf{E}_{\text{scene}} = \frac{1}{F}\sum_{i=1}^{F}\text{ChamferDist}(\mathbf{P}_i^{\text{gt-scene}}, \hat{\mathbf{P}}^{\text{scene}})$$

Object Reconstruction Similarity. Similarly to scene reconstruction similarity, we construct point clouds $\mathbf{P}^{\text{gt-obj}}$ and $\mathbf{P}^{\text{obj}}$ by uniformly sampling the ground-truth and reconstructed object meshes respectively. We then align the point clouds by optimizing an SE(3) transformation to yield $\hat{\mathbf{P}}^{\text{obj}}$. Finally, we evaluate object reconstruction similarity as the chamfer distance between the aligned point clouds.

$$\mathbf{E}_{\text{obj}} = \text{ChamferDist}(\mathbf{P}^{\text{gt-obj}}, \hat{\mathbf{P}}^{\text{obj}})$$

Trajectory Reconstruction. We evaluate object trajectory reconstruction using three metrics: absolute pose error translation (APE_{trans}), relative pose error translation (RPE_{trans}) and relative pose error rotation (RPE_{rot}). We compute these between the ground truth trajectory $\mathbf{Q}^{\text{gt-obj}}$ and the reconstructed trajectory $\mathbf{Q}^{\text{obj}}$, after aligning the reconstructed trajectories scale and SE(3) transformations.

The absolute pose error (APE) measures the deviation between corresponding poses and is defined as:

$$e_{\text{ape}}(\mathbf{Q}^{\text{gt-obj}}, \mathbf{Q}^{\text{obj}}) = \frac{1}{N}\sum_{i=1}^{N}\left\|\mathbf{Q}_{xyz}^{\text{gt-obj}} - \hat{\mathbf{Q}}_{xyz}^{\text{obj}}\right\|_2$$

where $\mathbf{Q}_{xyz}$ denotes the translation component of pose $\mathbf{Q}_i$ and $\hat{\mathbf{Q}}$ denotes the aligned trajectory.

The relative pose error (RPE) measures the local consistency of motion between consecutive poses. For an interval $\Delta = 1$, we define the relative transformations as:

$$\mathbf{T}_i^{\text{gt-rel}} = (\mathbf{Q}_i^{\text{gt-obj}})^{-1}\mathbf{Q}_{i+\Delta}^{\text{gt-obj}}$$
$$\mathbf{T}_i^{\text{obj-rel}} = (\hat{\mathbf{Q}}_i^{\text{obj}})^{-1}\hat{\mathbf{Q}}_{i+\Delta}^{\text{obj}}$$

The translation component of the RPE at time i is given by:

$$e_{\text{rpe,trans}}(i) = \left\|\text{trans}\left((\mathbf{T}_i^{\text{gt-rel}})^{-1}\mathbf{T}_i^{\text{obj-rel}}\right)\right\|_2$$

where $\text{trans}(\cdot)$ extracts the translation part of a transformation.

The rotation component of the RPE is given by:

$$e_{\text{rpe,rot}}(i) = \arccos\left(\frac{\text{trace}\left(\text{rot}\left((\mathbf{T}_i^{\text{gt-rel}})^{-1}\mathbf{T}_i^{\text{obj-rel}}\right)\right) - 1}{2}\right)$$

where $\text{rot}(\cdot)$ extracts the rotation matrix component of a transformation.

B.2 Data Details

We select 5 representative tasks from CALVIN [78] implemented in RoboVerse [76], covering the basic manipulation tasks on rigid objects: lift, push, rotate, "pick and place", and unstack blocks. These tasks are performed on top of a delicate desk, allowing for evaluating the proposed pipeline by reconstructing both the scene (desk) and the objects in interest (blocks).

B.3 Qualitative Results

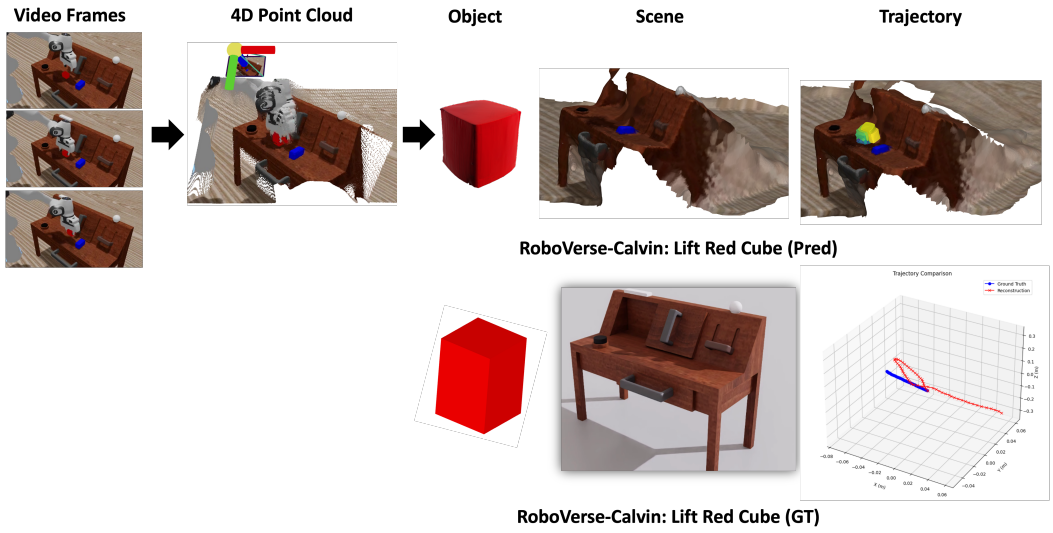

Figure 5: **Qualitative examples of our real-to-sim benchmark.**

C Additional Results

C.1 Additional Results on Real-to-Sim Pipeline

We present additional qualitative results for our real-to-sim pipeline in Fig. 6. Our method reconstructs the scene mesh, object mesh, and object trajectory in the same world coordinate reliably, across diverse platforms and interaction objects.

C.2 Additional Results on Real-World Deployment

For real-world deployment, we evaluate our method across a diverse set of task scenarios drawn from our dataset. In each test case, objects are placed in the same position and orientation as in the initial frame of the corresponding video. Using our vision pipeline, we extract the 3D scene mesh, object mesh, and object motion trajectory from the recorded demonstrations. Leveraging a real-to-sim-to-real pipeline, we train control policies in simulation with the motion planning approach and deploy them directly on the physical robot. In total, we conduct **18** real-world trials, of which **12** are successfully completed. These results in Fig. 7. highlight the robustness and practical effectiveness of our proposed method in transferring from simulation to real-world execution.

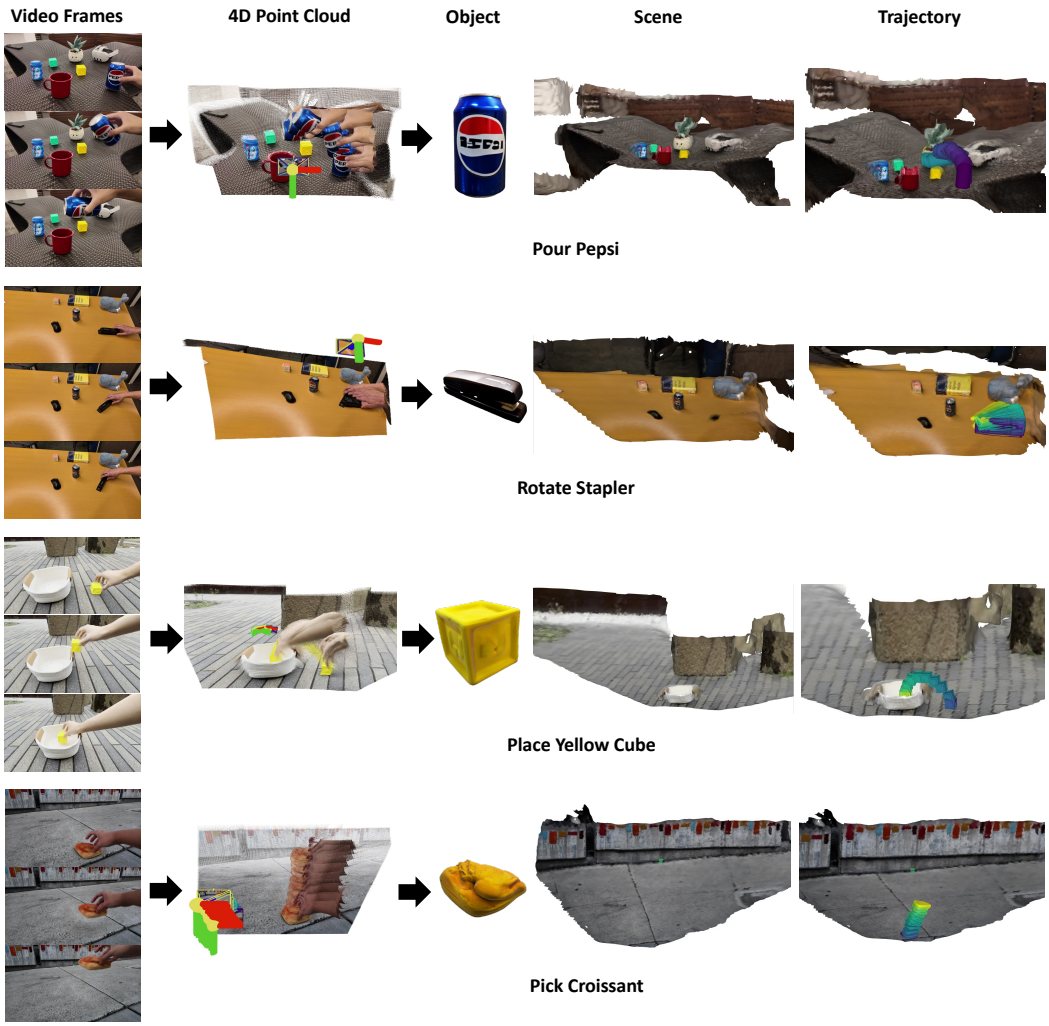

Figure 6: **Additional results on real-to-sim pipeline.**

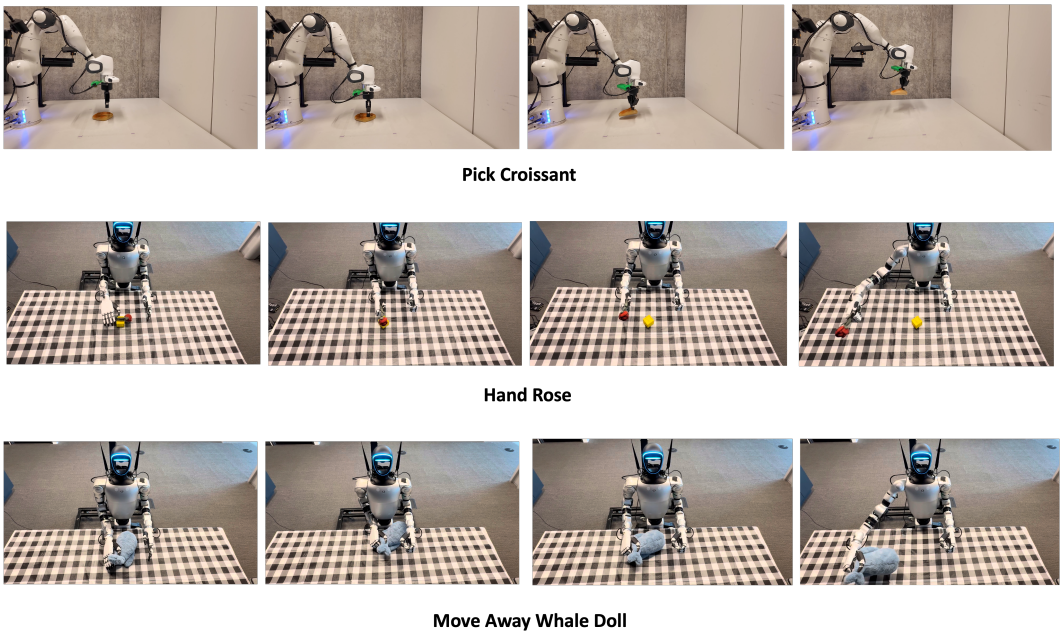

Figure 7: **Additional results on real-world deployment.**