# Real-to-Sim Transfer in Isaac Sim

Jaeryeong Kim  Jihwan Shin  Hugo De la Riva Fernandez
ETH Zürich
{jaerkim, jishin, hudela}@student.ethz.ch

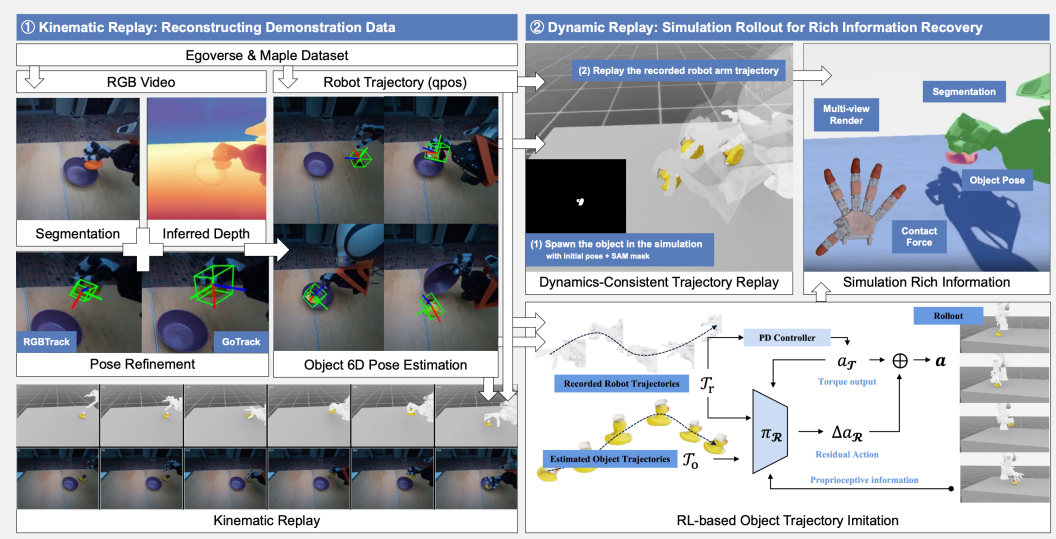


Figure 1. **Overview of Real2Sim.** (a) **Kinematic reconstruction:** Given RGB observations and recorded robot trajectories, we estimate temporally consistent 6-DoF object motion and reconstruct the demonstration in simulation. (b) **Dynamic reconstruction:** A residual policy refines the nominal robot trajectory under simulation dynamics, producing a physically consistent rollout with dense annotations, including multi-view images, segmentation masks, object poses, and contact forces.

## Abstract

*Teleoperation demonstrations are a key source of supervision for imitation learning yet their scalability is limited by two fundamental challenges: costly data collection and partial observability. Typical recordings provide RGB observations and robot trajectories while omitting object motion, contacts, and scene geometry. We present **Real2Sim**, a framework that reconstructs minimally observed teleoperation demonstrations as physically consistent simulation rollouts. From monocular RGB and robot trajectory, we estimate temporally coherent 6-DoF object trajectories and reconstruct the robot, objects, and scene in Isaac Sim. A residual reinforcement learning policy then refines the nominal robot trajectory under contact dynamics, reproducing the demonstrated object motion through physically plausible interaction. Each reconstructed demonstration is paired with dense privileged information—multi-view renders, depth, normals, segmentation, object and robot states, and contact forces—absent from the original recordings.*

## 1. Introduction

Imitation learning from human demonstrations has emerged as an effective paradigm for acquiring complex robotic manipulation skills [6, 20]. However, learning robust and generalizable policies typically requires large and diverse demonstration datasets [2, 8, 20]. Collecting such datasets on physical robots remains costly and time-consuming, due to operator effort [19, 37], specialized hardware [10, 39], and controlled collection setups. Moreover, teleoperation recordings provide only partial observations of the underlying interaction: robot commands and RGB observations may be available, while object state, contact information, and scene geometry are often missing.

Physics simulation offers a compelling means of addressing both limitations. Once a real-world demonstration has been reconstructed in simulation, the simulator provides dense privileged state and enables systematic augmentation across object configurations, robot poses, viewpoints, and scene appearances. Recent methods such as MimicGen and DexMimicGen have demonstrated that simulation-based augmentation can transform a small number of source demonstrations into large-scale manipulation datasets [14, 21]. DexMimicGen, for example, generated over 21,000 demonstrations from 60 source trajectories and substantially improved real-world performance.

Despite this progress, the preceding real-to-sim reconstruction stage remains largely unaddressed. Existing simulation-based augmentation methods, such as MimicGen papers [14, 21], assume access to simulation-compatible source demonstrations rather than providing a general procedure to recover them. Real2Render2Real reconstructs demonstrations through kinematic replay, but does not enforce dynamic feasibility or recover robot–object interaction [36]. To the best of our knowledge, no publicly available framework reconstructs partially observed teleoperation recordings as complete, physically plausible simulation trajectories.

We study this problem using minimally observed RGB recordings of real-world manipulation, including policy-executed robot rollouts from MAPLE and egocentric demonstrations from EgoVerse [11, 27]. Unlike recordings obtained with calibrated multi-view cameras or depth sensors, these datasets lack metric object trajectories. Reconstruction is further complicated by hand occlusion, monocular depth ambiguity, motion blur, and textureless objects. We use this minimally observed setting as a stress test for recovering complete, physically consistent demonstrations from RGB observations.

In this work, we introduce **Real2Sim**, a framework for reconstructing real-world manipulation demonstrations as physically consistent simulation rollouts. Real2Sim estimates temporally coherent 6-DoF object trajectories from RGB observations and reconstructs the robot, objects, and scene in simulation. It then uses residual reinforcement learning to refine the nominal robot trajectory, compensating for perception errors and dynamics mismatch while reproducing the demonstrated object motion through physically plausible interaction.

Our pipeline pairs each reconstructed demonstration with dense simulation state, including depth, segmentation, object and robot poses, velocities, contacts, and interaction forces. These paired rollouts enable controlled trajectory and scene augmentation while providing privileged supervision unavailable in the original recordings. They also establish explicit correspondence between real and simulated observations, supporting downstream real–synthetic training and domain-gap mitigation.

In summary, our contributions are as follows:

- We present **Real2Sim**, an end-to-end framework for reconstructing RGB-based real-world manipulation demonstrations as physically plausible simulation rollouts.
- We develop a robust RGB 6-DoF object-tracking and scene-reconstruction pipeline designed for monocular observations with severe robot-induced occlusion.
- We formulate physically consistent demonstration recovery as a residual reinforcement learning formulation that refines estimated object trajectories and nominal robot trajectories to reproduce demonstrated object motion.

## 2. Related Work

**6D Object Pose Estimation.** Most established model-based 6D pose estimators, including FoundationPose [33] and BundleSDF [32], rely on known object models and RGB-D observations. Although recent RGB-only methods have improved monocular tracking and refinement [3, 12, 22–24], they are primarily evaluated under standard model-based settings with sufficient object visibility. In dexterous hand demonstrations, however, the hand can heavily occlude the object throughout manipulation, while small image footprints and near-symmetric geometry further degrade visual tracking. We address this heavily occluded, depth-free setting by combining monocular RGB with synchronized robot trajectories, a calibrated camera, and a known object mesh. These proprioceptive cues enable our tracker to identify grasp and release events and maintain object pose through in-hand occlusion, where frame-wise RGB estimation alone is unreliable. (Sec. 3.1).

**Real-to-Sim Transfer.** Transferring real-world data into simulation enables physically consistent data generation, policy training, and evaluation without the cost and risk of operating on hardware.

A large body of work focuses on *digital-twin reconstruction*, recovering geometry, appearance, and physical properties of real scenes and objects so that they can be reinstantiated in a simulator [26, 28, 35]. These methods target the fidelity of the reconstructed *environment*; they do not address how an executed demonstration trajectory should be reproduced under contact dynamics.

Another line of work closes the loop as *real-to-sim-to-real*, reconstructing a simulation from human video to train a policy that is then deployed back on hardware [9, 17, 36]. Most closely to our work, Human2Sim2Robot [18] reconstructs a simulated scene from an RGB-D human demonstration and extract the object pose trajectory to train a robot policy from scratch with in-simulation RL before transferring it back to a real robot.

Our setting differs in both input and goal. Rather than learning a deployable policy from human video, we transfer

*teleoperated robot demonstrations*—egocentric video with recorded proprioception—into Isaac Sim. The transferred high quality data can then be used for downstream tasks such as deployable policies.

**Residual Policy Learning.** Residual policy learning augments a nominal controller or reference policy with learned corrective actions, often improving sample efficiency and training stability over learning complex manipulation behaviors from scratch [15, 30, 31]. Prior work has applied this paradigm to grasp refinement and dexterous motion retargeting [34, 38]. Most closely related to our work, ManipTrans [17] first generates a coarse robot trajectory using a pretrained hand-motion imitation policy and then learns a task-specific residual policy conditioned on object states and simulated contacts to satisfy physical interaction constraints. Building on ManipTrans, we adapt this framework to real-to-sim reconstruction, where both robot and object trajectories are imperfect estimates rather than ground-truth references. The residual policy refines the robot motion, while the object trajectory is corrected implicitly through simulated contact dynamics, yielding a physically plausible reconstruction of the demonstration.

## 3. Method

### 3.1. 6D Object Pose Estimation

Transferring a demonstration to simulation requires the manipulated object's full per-frame 6-DoF trajectory. Our recordings provide only a single egocentric RGB stream together with robot proprioception and rigid object meshes. Recovering pose from this signal is hard for three reasons: (i) objects are often near-symmetric, leaving the rotation ambiguous; (ii) robot heavily occludes the object during manipulation; (iii) no depth information is given. We divide the object 6D pose estimation pipeline into multiple stages below. Robot proprioception and object mask are used to deduce when the object grasped and released. The final object 6D pose estimate trajectory is then supplied to Sec. 3.3.

**Preprocessing.** We obtain the object masks with SAM2 video predictor [29] or SAM3 [4]. Depth is inferred using Video Depth Anything [5]; we further calibrate the metric result scale by a binary-search depth solve against the object mask.

**Initial track.** We get an initial pose estimate from preprocessed information with RGBTrack [12]. The first frame's orientation (which corrupts the whole track when wrong) is refined using GoTrack [23].

**Pre-grasp.** Before the object moves, we keep the pose fixed at the clean first-frame estimate while the hand remains open.

**In-hand.** While the object is grasped we estimate the pose using the most trustworthy signal. For *position* we keep RGBTrack's estimate but slide it back onto the mask whenever the rendered silhouette drifts off, since the mask shows where the object really is. For *depth* we use the robot: a firmly-held object moves with the end-effector, so the end-effector forward kinematics (from the joint log) give a clean depth, free of the jitter that plagues video-estimated depth. For *rotation* we use GoTrack as a *noisy teacher*—it occasionally flips the symmetric object, so we discard frames whose orientation jumps from their neighbours and interpolate the rest. Order matters: GoTrack only *refines* a pose, it cannot find one from scratch, so it locks on only when started from this kinematics-corrected in-hand pose—which then serves as the high-quality anchor everything downstream is measured against. Started from the raw tracker, it diverges.

**Post-release.** Once the robot releases the object, the object is no-longer heavily occluded. We run GoTrack over the post-release window then apply anomaly-filtering and smoothing.

### 3.2. Simulation Reconstruction

**Object.** The manipulated objects have been scanned using the Artec Leo [1] 3D scanner and meshed into an OBJ file with UV texture map (Fig. 3). We suspend the objects in air using thin strings to scan them from all directions, applying masking tape on surfaces which are trasparent or reflective to improve scan quality. Objects with thin edges (e.g. a plastic shovel) often caused divergence problems with the scanner, which required us to align them manually using the Artec Studio software.

**Robot.** The Franka Emika Panda [13] and ORCA hand [7] are connected with a 3D-printed 90-degree connector component. We create the URDF of the full robot based on manufacturers' files then import it in Isaac Sim through its URDF-to-USD converter.

**Scene.** The scene setup of EgoVerse and MAPLE are similar with minor differences. Different camera are used for the egocentric-viewpoint of each dataset, and their extrinsics are calibrated using different methods: EgoVerse uses Aria glasses calibrated by aligning the edges of the table, while MAPLE uses OAK-D Pro calibrated by an April tag [25]. Comparing the source egocentric video against simulated render shows high correspondence (Fig. 4).

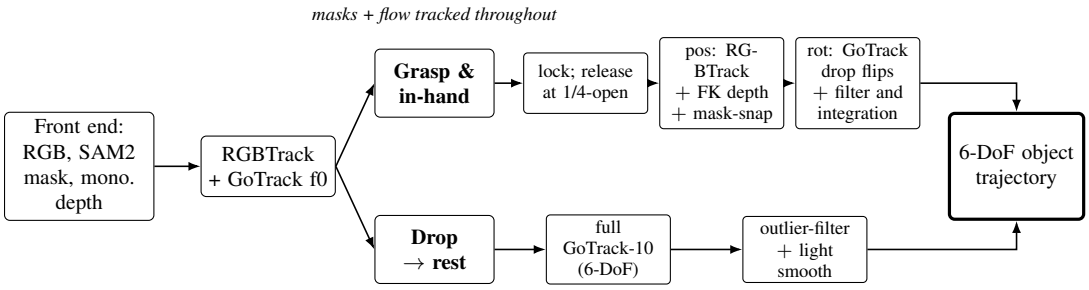


Figure 2. **Object pose estimation pipeline.** We adopt a multi-stage pipeline for a robust pose estimator in egocentric teleoperation demonstrations.

Figure 3. **Objects meshes** (ball, duck, fish, grape, pan, shovel) scanned using Artec Leo 3D scanner.

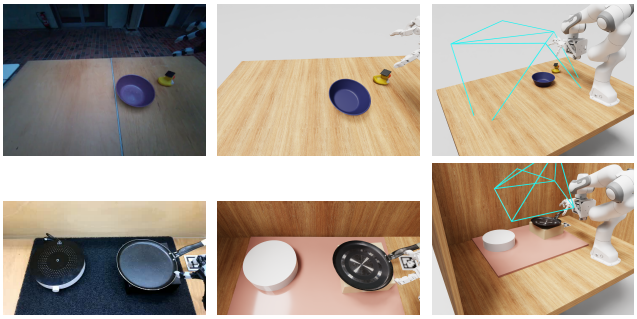

Figure 4. **Source egocentric video, simulation egocentric and third-person render** of EgoVerse (Top) and MAPLE (Bottom) demonstrations in Isaac Sim.

### 3.3. Residual RL Object Trajectory Imitation

Given a monocular RGB recording, the perception stage (Sec. 3.1) produces an *estimated* 6-DoF object trajectory $\tilde{\boldsymbol{p}}_o^{1:T}$. A *nominal* robot trajectory $\bar{\boldsymbol{q}}^{1:T}$ are provided from dataset in joint commands and wrist position. Neither is physically valid in isolation: the object estimate is corrupted by occlusion and depth ambiguity, and open-loop replay of $\bar{\boldsymbol{q}}$ does not reproduce the object motion under contact, due to the latency and noise in proprioceptive sensors. We recover a single physically consistent rollout that reconciles the two, jointly refining the robot trajectory *explicitly* and the object trajectory *implicitly*, through simulated contact dynamics. Unlike ManipTrans [17], which corrects a learned imitator, we correct a deterministic recorded trajectory to reconstruct partially observed teleoperation data.

**Residual Policy.** At each step, the policy $\pi_\theta$ emits a bounded residual on the nominal configuration, applied as a PD joint-position target:

$$\boldsymbol{a}^t = \mathrm{clip}\big(\bar{\boldsymbol{q}}^{\,t} + \alpha\, \pi_\theta(\boldsymbol{o}^t),\ \boldsymbol{\ell}, \boldsymbol{u}\big), \qquad \alpha = 0.1. \tag{1}$$

The base $\bar{\boldsymbol{q}}^{\,t}$ is a deterministic lookup of the *nominal* robot trajectory recorded. The reward drives it onto the estimate $\tilde{\boldsymbol{p}}_o^{1:T}$. The object trajectory is thus de-noised by projection onto the manifold of dynamically feasible motions, whereas the robot trajectory is corrected to produce it.

**Observations and Reward.** We use an asymmetric actor–critic. The actor observes proprioception, the object position in the robot frame, the previous action, and a one-step look-ahead of the reference; the critic additionally receives privileged object velocities and orientation. The reward drives the simulated object onto the estimated reference trajectory, with a small action-magnitude penalty:

$$r^t = w_p\, r^t_{\text{obj-p}} + w_r\, r^t_{\text{obj-r}} - \lambda_a\, \|\boldsymbol{\delta}^t\|_2^2, \tag{2}$$

where $r_{\text{obj-p}}$ and $r_{\text{obj-r}}$ are object position- and orientation-tracking terms and $\boldsymbol{\delta}^t = \pi_\theta(\boldsymbol{o}^t)$ is the policy residual.

### 3.4. Simulation Rich Information

Once a demonstration has been transferred to simulation, we use the Isaac Sim API to extract rich supervision from the rollouts. This yields modalities that are absent from the original recordings, letting us make greater use of the scarce human demonstration data.

**Multi-viewpoint Render.** Because the scene is fully reconstructed, we can freely set the camera intrinsics and extrinsics to render novel viewpoints in addition to the egocentric one. This lets us obtain views that are free of occlusions or impractical to capture in a real-world setup, increasing the diversity of the visual data.

**Depth, Normal and Segmentation.** On real RGB video, recovering these modalities would require estimation models such as Video Depth Anything [5] for depth or SAM 3 [4] for segmentation. Such predictions are inevitably imperfect, suffering from noise, temporal inconsistency, or outright errors. In simulation, we instead read ground-truth depth, surface normals, and segmentation masks directly from the renderer, free of these limitations.

**Contact Force Measurement.** Contact forces are a valuable cue for dexterous manipulation, informing how the hand should make and maintain contact with an object [16]. We attach Isaac Sim contact sensors to the 11 skin meshes of the ORCA hand and record scalar contact-force measurements throughout each manipulation task.

# 4. Result

## 4.1. 6D Object Pose Estimation

We have no metric ground truth for the real recordings, so we report a *vector* of GT-free proxies—rendered-silhouette IoU (mean), temporal jerk, and fall-window occlusion survival—plus a near-ground-truth *in-hand rigidity* signal: while the object is rigidly grasped, the object-in-hand transform from the robot kinematics should be constant, so its variance over the grasp window measures pose error directly. Silhouette IoU is our most informative proxy, but it is a *relative*, mask-dependent comparison against a heavy occulsion mask, so a pose can score well by over-fitting the mask rather than being correct; we therefore pair it with direct visual-overlay inspection, and the downstream simulation-transfer result, which a mask-overfit pose would fail. We report every metric over the in-scope phases {static, grasp, fall, settle} across three duck sequences. The ablation compares three configurations: **(a)** raw RGBTrack on our inputs (scale-calibrated monocular depth and SAM masks), **(b)** our adapted RGBTrack (`C2`), and **(c)** the final composed track (`C5`).

**Ablation.** Across the three duck sequences the final composed track beats both the adapted base and the raw rung on *every* metric of the vector in the three-sequence mean (Tab. 1): the four-phase mean IoU rises to 0.65 (from raw 0.51 / adapted 0.43), the in-hand grasp jerk collapses to 5 mm and the rotational rigidity $\sigma_R$ to 46°, and fall occlusion-survival improves to 0.98—the largest gains being in-hand (the near-ground-truth signal that the held pose is genuinely correct rather than merely mask-consistent) and through the occluded fall. The main finding is that GoTrack pays off only from the proprioceptive (`h5`) initialization—seeded from it the catastrophic flips vanish (grasp-window jerk median $459 \rightarrow 0.4$ mm, angular velocity $90° \rightarrow 1°$, mask-verified usable frames $22 \rightarrow 63\%$)—and diverges when applied to the bare RGBTrack track; GoTrack is a refiner, not a detector.

Table 1. **GT-free metric vector averaged over the duck sequences**.

| Config. | IoU↑ mean | $\text{jerk}_{rms}$↓ grasp, mm | $\sigma_R$↓ grasp, ° | occ. surv.↑ fall |
|---|---|---|---|---|
| Raw + our inputs | 0.51 | 50 | 96 | 0.56 |
| Adapted (`C2`) | 0.43 | 63 | 85 | 0.64 |
| Final (`C5`) | **0.65** | **5** | **46** | **0.98** |

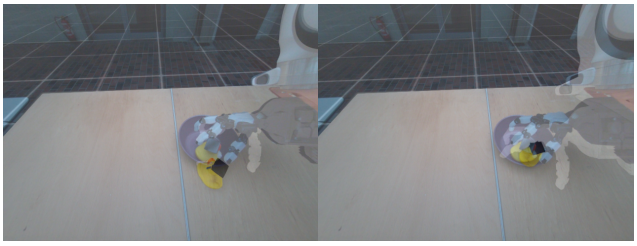

Figure 5. **Terminal-frame alignment of the object with the real recording. (Left)** Open-loop dynamic replay drifts from the recorded duck. **(Right)** Our residual policy stays aligned.

## 4.2. Simulated Replay

**Dynamic replay (no residual).** As a comparison, we replay the nominal robot trajectory under full contact dynamics with the residual disabled ($\alpha$=0 in Eq. 1): the recorded arm and hand joint angles are tracked open-loop by the PD controllers. Unlike kinematic replay, the simulator resolves contact, so hard interpenetration has been resolved (Tab. 2); but the open-loop trajectory deviates from the real recording without corrections. In dynamic replay, robot over-squeezes objects and lets it drift 76.6 mm and 86.6° from the reference object pose by the end of the episode.

**RL result.** The residual policy turns this into a physically consistent reconstruction. It attains the lowest interpenetration of all modes (Tab. 2), and reproduces the demonstrated object trajectory far better. For instance, the terminal object error 25.6 mm and 44.3° (Fig. 5), while lifting the object 16.9 cm. The grasp is also gentler: the fingertip contact force during the hold is $8.3 \pm 3.3$ N . Fig. 6 overlays the reconstruction against the source Aria RGB, showing the grasp and lift faithfully reproduced.

**Simulation rich information.** Once the demonstration is transferred to a physically consistent rollout, we can extract simulation rich information. Fig. 7 shows a montage of rendered modalities from a trained simulation rollout. The corresponding skin meshes in contact force measurement turn

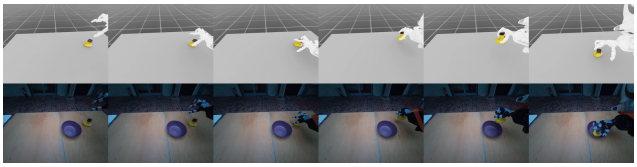

Figure 6. **Comparison of simulation rollout against demonstration video from EgoVerse.**

Table 2. **Duck–robot interpenetration** with peak penetration number and the fraction of interpenetration over rollout.

| Replay mode | Peak penet. (mm) ↓ | Frames >1 mm ↓ |
|---|---|---|
| Kinematic | 17.5 | 48.9% |
| Dynamic | 4.75 | 20.0% |
| Residual RL (ours) | **3.21** | **14.9%** |

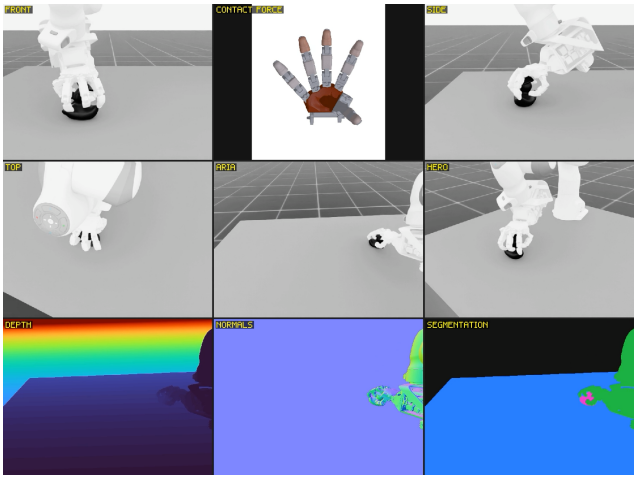


Figure 7. **Simulation-rich information extracted from a single transferred demonstration**: multi-viewpoint render, metric depth, surface normals, instance segmentation, contact force measurement.

red as contact force increases.

## 5. Discussions

**Limitations.** Our work has only been tested and verified on a small subset of teleoperation datasets. To verify the robustness, we may extend the experiment to varying setups, objects and manipulation tasks. The robot and object collisions are approximated as convex hulls for reduced computational demand. Usage of more precise approximation methods (convex decomposition, SDF, triangle mesh, etc.) can be considered for more detailed manipulation tasks.

**Future Works.** We can consider the following options for future works beyond our paper:

- **Augmentation**: The simulation-transferred demonstrations can be augmented to new data by varying visual features (scene background, texture, lighting), object mesh and trajectories.
- **Bimanual dexterous tasks**: We can extend the framework to bimanual dexterous tasks covered in the EgoVerse dataset.
- **Downstream training**: The refined demonstrations, along with the simulation rich information, can be used to train downstream policies with behaviour cloning.

## 6. Conclusion

We presented **Real2Sim**, a framework that reconstructs minimally observed teleoperation demonstrations as physically consistent simulation rollouts. From monocular RGB we estimate 6-DoF object trajectories and rebuild the robot, objects, and scene in Isaac Sim, then use residual reinforcement learning to refine the nominal robot trajectory under contact dynamics. The recovered rollouts attain lower interpenetration and more faithful object tracking than kinematic or open-loop dynamic replay, and each is paired with dense privileged modalities unavailable in the source recordings. Our experiments cover a subset of simple pick-and-place tasks; extending the pipeline to richer, non-quasi-static manipulation and validating the augmented data on downstream policy learning remain promising directions for future work.

**Acknowledgements.** This project was supervised by Jiaqi (Julia) Chen and Alexey Gavryushin of the Computer Vision and Geometry Group (CVG). Ball and duck meshes (Fig. 3) have been provided by Tristan K., Maxence L., Heeyoun L., Mathieu P. of ETH Zürich.

**Work Distribution.** This project is equally well distributed in workload. Hugo made the robust object 6D pose estimation pipeline. Jihwan made the Isaac Sim reconstruction of the EgoVerse and MAPLE setups. Jaeryeong implemented the dynamic-consistent trajectory replay and the residual RL pipeline for recovering the dynamically feasible rollout.

**Codebase.** We share the following repositories of our pipeline for use:

- **pandaorca_description**: URDF and USD files of the robot used. https://github.com/3dv-fs26-real2sim/pandaorca_description
- **dataset_replay**: Scene reconstruction of the EgoVerse and MAPLE datasets in Isaac Sim and residual reinforcement learning training pipeline. https://github.com/3dv-fs26-real2sim/dataset_replay

## References


[1] Artec 3D. Artec leo: Wireless AI-driven 3D scanner. https://www.artec3d.com/portable-3d-scanners/artec-leo, 2022. Accessed: 2026-03-06. 3

[2] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, et al. Rt-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*, 2022. 1

[3] Andrea Caraffa, Davide Boscaini, Amir Hamza, and Fabio Poiesi. Accurate and efficient zero-shot 6d pose estimation with frozen foundation models, 2025. 2

[4] Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, Jie Lei, Tengyu Ma, Baishan Guo, Arpit Kalla, Markus Marks, Joseph Greer, Meng Wang, Peize Sun, Roman Rädle, Triantafyllos Afouras, Effrosyni Mavroudi, Katherine Xu, Tsung-Han Wu, Yu Zhou, Liliane Momeni, Rishi Hazra, Shuangrui Ding, Sagar Vaze, Francois Porcher, Feng Li, Siyuan Li, Aishwarya Kamath, Ho Kei Cheng, Piotr Dollár, Nikhila Ravi, Kate Saenko, Pengchuan Zhang, and Christoph Feichtenhofer. Sam 3: Segment anything with concepts, 2025. 3, 5

[5] Sili Chen, Hengkai Guo, Shengnan Zhu, Feihu Zhang, Zilong Huang, Jiashi Feng, and Bingyi Kang. Video depth anything: Consistent depth estimation for super-long videos, 2025. 3, 5

[6] Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, 44 (10-11):1684–1704, 2025. 1

[7] Clemens C. Christoph, Maximilian Eberlein, Filippos Katsimalis, Arturo Roberti, Aristotelis Sympetheros, Michel R. Vogt, Davide Liconti, Chenyu Yang, Barnabas Gavin Cangan, Ronan J. Hinchet, and Robert K. Katzschmann. Orca: An open-source, reliable, cost-effective, anthropomorphic robotic hand for uninterrupted dexterous task learning, 2025. 3

[8] Open X-Embodiment Collaboration. Open x-embodiment: Robotic learning datasets and rt-x models, 2025. 1

[9] Prithwish Dan, Kushal Kedia, Angela Chao, Edward Weiyi Duan, Maximus Adrian Pace, Wei-Chiu Ma, and Sanjiban Choudhury. X-sim: Cross-embodiment learning via real-to-sim-to-real, 2025. 2

[10] Zipeng Fu, Tony Z Zhao, and Chelsea Finn. Mobile aloha: Learning bimanual mobile manipulation with low-cost whole-body teleoperation. *arXiv preprint arXiv:2401.02117*, 2024. 1

[11] Alexey Gavryushin, Xi Wang, Robert J. S. Malate, Chenyu Yang, Davide Liconti, René Zurbrügg, Robert K. Katzschmann, and Marc Pollefeys. Maple: Encoding dexterous robotic manipulation priors learned from egocentric videos, 2025. 2

[12] Teng Guo and Jingjin Yu. Rgbtrack: Fast, robust depth-free 6D pose estimation and tracking, 2025. 2, 3

[13] Sami Haddadin. The franka emika robot: A standard platform in robotics research [survey]. *IEEE Robotics & Automation Magazine*, 31(4):136–148, 2024. 3

[14] Zhenyu Jiang, Yuqi Xie, Kevin Lin, Zhenjia Xu, Weikang Wan, Ajay Mandlekar, Linxi Fan, and Yuke Zhu. Dexmimicgen: Automated data generation for bimanual dexterous manipulation via imitation learning, 2025. 2

[15] Tobias Johannink, Shikhar Bahl, Ashvin Nair, Jianlan Luo, Avinash Kumar, Matthias Loskyll, Juan Aparicio Ojea, Eugen Solowjow, and Sergey Levine. Residual reinforcement learning for robot control. In *2019 international conference on robotics and automation (ICRA)*, pages 6023–6029. IEEE, 2019. 3

[16] Zhanat Kappassov, Juan-Antonio Corrales, and Véronique Perdereau. Tactile sensing in dexterous robot hands — review. *Robotics and Autonomous Systems*, 74:195–220, 2015. 5

[17] Kailin Li, Puhao Li, Tengyu Liu, Yuyang Li, and Siyuan Huang. Maniptrans: Efficient dexterous bimanual manipulation transfer via residual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6991–7003, 2025. 2, 3, 4

[18] Tyler Ga Wei Lum, Olivia Y. Lee, C. Karen Liu, and Jeannette Bohg. Crossing the human-robot embodiment gap with sim-to-real rl using one human demonstration, 2025. 2

[19] Ajay Mandlekar, Yuke Zhu, Animesh Garg, Jonathan Booher, Max Spero, Albert Tung, Julian Gao, John Emmons, Anchit Gupta, Emre Orbay, et al. Roboturk: A crowdsourcing platform for robotic skill learning through imitation. In *Conference on Robot Learning*, pages 879–893. PMLR, 2018. 1

[20] Ajay Mandlekar, Danfei Xu, Josiah Wong, Soroush Nasiriany, Chen Wang, Rohun Kulkarni, Li Fei-Fei, Silvio Savarese, Yuke Zhu, and Roberto Martín-Martín. What matters in learning from offline human demonstrations for robot manipulation. *arXiv preprint arXiv:2108.03298*, 2021. 1

[21] Ajay Mandlekar, Soroush Nasiriany, Bowen Wen, Iretiayo Akinola, Yashraj Narang, Linxi Fan, Yuke Zhu, and Dieter Fox. Mimicgen: A data generation system for scalable robot learning using human demonstrations, 2023. 2

[22] Sungphill Moon, Hyeontae Son, Dongcheol Hur, and Sangwook Kim. Co-op: Correspondence-based novel object pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. arXiv:2503.17731. 2

[23] Van Nguyen Nguyen, Christian Forster, Sindi Shkodrani, Vincent Lepetit, Bugra Tekin, Cem Keskin, and Tomas Hodan. Gotrack: Generic 6DoF object pose refinement and tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops (CV4MR)*, 2025. arXiv:2506.07155. 3

[24] Van Nguyen Nguyen, Stephen Tyree, Andrew Guo, Mederic Fourmy, Anas Gouda, Taeyeop Lee, Sungphill Moon, Hyeontae Son, Lukas Ranftl, Jonathan Tremblay, Eric Brachmann, Bertram Drost, Vincent Lepetit, Carsten Rother, Stan Birchfield, Jiri Matas, Yann Labbe, Martin Sundermeyer, and Tomas Hodan. Bop challenge 2024 on model-based and model-free 6d object pose estimation, 2025. 2

[25] Edwin Olson. AprilTag: A robust and flexible visual fiducial system. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 3400–3407. IEEE, 2011. 3

[26] Nicholas Pfaff, Evelyn Fu, Jeremy Binagia, Phillip Isola, and Russ Tedrake. Scalable real2sim: Physics-aware asset generation via robotic pick-and-place setups, 2025. 2

[27] Ryan Punamiya, Simar Kareer, Zeyi Liu, Josh Citron, Ri-Zhao Qiu, Xiongyi Cai, Alexey Gavryushin, Jiaqi Chen, Davide Liconti, Lawrence Y. Zhu, Patcharapong Aphiwetsa, Baoyu Li, Aniketh Cheluva, Pranav Kuppili, Yangcen Liu, Dhruv Patel, Aidan Gao, Hye-Young Chung, Ryan Co, Renee Zbizika, Jeff Liu, Xiaomeng Xu, Haoyu Xiong, Geng Chen, Sebastiano Oliani, Chenyu Yang, Xi Wang, James Fort, Richard Newcombe, Josh Gao, Jason Chong, Garrett Matsuda, Aseem Doriwala, Marc Pollefeys, Robert Katzschmann, Xiaolong Wang, Shuran Song, Judy Hoffman, and Danfei Xu. Egoverse: An egocentric human dataset for robot learning from around the world, 2026. 2

[28] Tianshuang Qiu, Zehan Ma, Karim El-Refai, Hiya Shah, Chung Min Kim, Justin Kerr, and Ken Goldberg. Omni-scan: Creating visually-accurate digital twin object models using a bimanual robot with handover and gaussian splat merging, 2025. 2

[29] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollár, and Christoph Feichtenhofer. SAM 2: Segment anything in images and videos, 2024. 3

[30] Gerrit Schoettler, Ashvin Nair, Jianlan Luo, Shikhar Bahl, Juan Aparicio Ojea, Eugen Solowjow, and Sergey Levine. Deep reinforcement learning for industrial insertion tasks with visual inputs and natural rewards. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5548–5555. IEEE, 2020. 3

[31] Tom Silver, Kelsey Allen, Josh Tenenbaum, and Leslie Kaelbling. Residual policy learning. *arXiv preprint arXiv:1812.06298*, 2018. 3

[32] Bowen Wen, Jonathan Tremblay, Valts Blukis, Stephen Tyree, Thomas Muller, Alex Evans, Dieter Fox, Jan Kautz, and Stan Birchfield. Bundlesdf: Neural 6-dof tracking and 3d reconstruction of unknown objects, 2023. 2

[33] Bowen Wen, Wei Yang, Jan Kautz, and Stan Birchfield. Foundationpose: Unified 6d pose estimation and tracking of novel objects, 2024. 2

[34] Tianhao Wu, Mingdong Wu, Jiyao Zhang, Yunchong Gan, and Hao Dong. Graspgf: Learning score-based grasping primitive for human-assisting dexterous grasping. *arXiv preprint arXiv:2309.06038*, 2023. 3

[35] 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. 2

[36] Justin Yu, Letian Fu, Huang Huang, Karim El-Refai, Rares Andrei Ambrus, Richard Cheng, Muhammad Zubair Irshad, and Ken Goldberg. Real2render2real: Scaling robot data without dynamics simulation or robot hardware, 2025. 2

[37] Tianhao Zhang, Zoe McCarthy, Owen Jow, Dennis Lee, Xi Chen, Ken Goldberg, and Pieter Abbeel. Deep imitation learning for complex manipulation tasks from virtual reality teleoperation. In *2018 IEEE international conference on robotics and automation (ICRA)*, pages 5628–5635. Ieee, 2018. 1

[38] Shuqi Zhao, Xinghao Zhu, Yuxin Chen, Chenran Li, Yichen Xie, Xiang Zhang, Mingyu Ding, and Masayoshi Tomizuka. Dexh2r: Task-oriented dexterous manipulation from human to robots. *IEEE/ASME Transactions on Mechatronics*, 2025. 3

[39] Tony Z. Zhao, Jonathan Tompson, Danny Driess, Pete Florence, Kamyar Ghasemipour, Chelsea Finn, and Ayzaan Wahid. Aloha unleashed: A simple recipe for robot dexterity, 2024. 1