WARPED: Wrist-Aligned Rendering for Robot Policy Learning from Egocentric Human Demonstrations

Harry Freeman[1], Chung Hee Kim[1], George Kantor[1]

***Abstract*—Recent advancements in learning from human demonstration have shown promising results in addressing the scalability and high cost of data collection required to train robust visuomotor policies. However, existing approaches are often constrained by a reliance on multiview camera setups, depth sensors, or custom hardware and are typically limited to policy execution from third-person or egocentric cameras. In this paper, we present WARPED, a framework designed to synthesize realistic wrist-view observations and actions from human demonstration videos to facilitate the training of visuomotor policies using only monocular RGB data. With data collected from an egocentric RGB camera, our system leverages vision foundation models to initialize the interactive scene. A hand-object interaction pipeline is then employed to track the hand and manipulated object and retarget the trajectories to a robotic end-effector. Lastly, photo-realistic wrist-view observations are synthesized via Gaussian Splatting to directly train a robotic policy. We demonstrate that WARPED achieves success rates comparable to policies trained on teleoperated demonstration data for five tabletop manipulation tasks, while requiring 5–8x less data collection time.**

I. Introduction

Imitation learning [84, 39, 37, 31, 44, 114, 124, 42] has emerged as a popular approach to train robotic visuomotor policies to perform a variety of manipulation tasks. These tasks range from simple pick and place, pushing, and insertion [41, 108, 93], to more complex long-horizon tasks such as folding laundry, tool use, and washing dishes [75, 12, 16]. However, the performance of these policies is highly dependent on the availability and quality of the demonstration data. Methods often utilize existing large-scale datasets from teleoperated robot demonstrations [19, 28] or internet videos [20, 32], which are expensive and difficult to collect when scaling to new tasks and environments. This challenge is more evident for domain-specific manipulation tasks, such as agriculture [47], where demonstration data is often limited. Alternatively, methods can rely on collecting new teleoperated robot data [40, 85, 22, 77], which is slow, time-consuming, and labor-intensive to acquire.

Recent works on learning directly from a small amount of human demonstrations without teleoperation or large-scale datasets have been proposed to address these limitations and improve the scalability and adaptability of training visuomotor policies, often using small amounts of task-specific data. Because humans manipulate objects quickly and naturally, demonstrations can be collected significantly faster than teleoperation, allowing for rapid adaptation to new tasks. To bridge the observation gap between human demonstrations and robot execution, prior approaches have utilized specialized data collection interfaces [120, 98, 119], data augmentation strategies [70, 67, 30], and simplified intermediate representations such as hand-pose trajectories [113, 71], keypoints [57, 33, 103], and object-centric 3D representations [46, 109, 122].

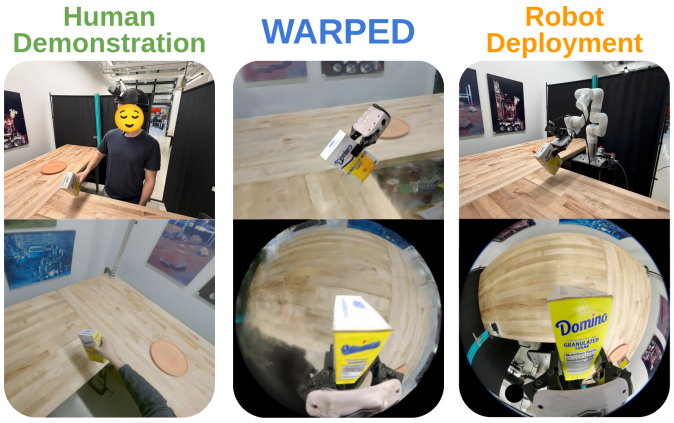

Fig. 1: WARPED: A framework that warps egocentric human demonstrations into wrist-camera observations and trajectories for training robot policies.

However, many of these approaches still depend on additional sensing or setups that limit how easily data can be collected for new tasks. Several works require multi-view images [80, 33, 101] or depth sensors [72, 58, 56] for reconstruction and tracking; rely on custom hardware-based collection interfaces [17, 94], or depend on custom-trained generative models to convert human demonstrations into robot-compatible observations [36, 4]. As a result, if a user wants to train a policy for a new manipulation task, they need one of these components, which may be difficult to use or not readily available. Additionally, many of these methods are limited to rolling out policies on fixed or egocentric camera viewpoints. Wrist-mounted camera viewpoints are often desirable, as they capture more fine-grained details of the interactive scene [1, 18, 48]. Because most human demonstration-based approaches require policies to be executed from camera viewpoints similar to those used during data collection, they cannot readily leverage wrist-view camera observations at deployment.

To address these limitations, we present **W**rist-**A**ligned **R**endering for Robot **P**olicy Learning from **E**gocentric Human **D**emonstrations (WARPED), an approach that enables training visuomotor policies from human demonstrations without requiring multi-view sensing, depth sensors, or custom collection hardware. With only a single monocular RGB camera worn on the head of the user, WARPED takes egocentric videos of human demonstrations and produces robot end-effector-

[1]Carnegie Mellon University Robotics Institute, PA, USA {hfreeman, chunghek, kantor}@cs.cmu.edu

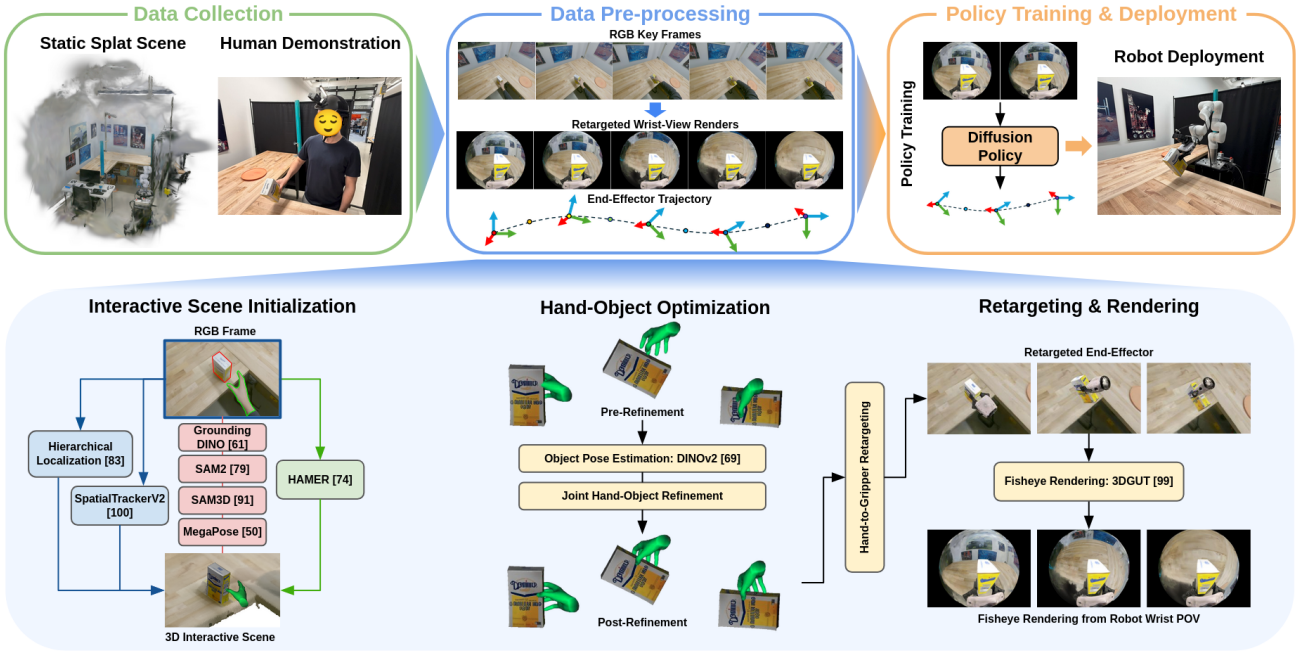

Fig. 2: Overview of WARPED. Images of the scene are captured to build an initial Gaussian Splat representation. The user then performs a tabletop manipulation task, recording an egocentric demonstration from a GoPro attached to a helmet. The interactive scene is aligned with the Gaussian Splat, and the object pose is localized using vision foundation models. The object pose is then tracked over time via hand–object optimization. The resulting hand trajectories are retargeted to a robot gripper, and per-frame wrist-camera views of the scene are rendered and used to train a diffusion policy.

aligned observations and actions from a wrist-camera perspective that can be used directly for policy learning. Our system leverages vision foundation models to initialize the scene and a hand-object interaction pipeline to track and retarget human trajectories to a robotic end-effector. We then use Gaussian Splatting to render photorealistic wrist-view observations to train a robotic policy. We demonstrate that WARPED achieves performance comparable to teleoperation across five tabletop manipulation tasks while requiring 5-8 times less data collection time. Our specific contributions are:

- A pipeline for generating robot-aligned wrist-view observations and trajectories from egocentric human demonstrations to train visuomotor policies.
- A monocular RGB-only approach that integrates vision foundation models and Gaussian Splatting to synthesize photorealistic wrist-view observations without multi-view setups, depth sensors, or custom hardware.
- An evaluation and analysis on five tabletop manipulation tasks demonstrating teleoperation-level performance with significantly reduced data collection time.

II. Related Work

A. Imitation Learning

Recent progress in imitation learning has focused on learning visuomotor policies directly from visual and proprioceptive inputs [7, 49, 90, 8, 9], allowing robots to perform increasingly complex manipulation tasks. Advances in policy architectures, including sequence modeling and diffusion-based formulations [119, 16], have further improved the expressiveness and real-world performance of imitation learning methods. However, despite these advances, most approaches rely on large amounts of robot-collected demonstration data, making them difficult to scale to new tasks, scenes, and object instances.

Expert demonstrations for imitation learning are often collected through teleoperation, where a human controls a robot and records observations and actions. Teleoperation can be performed using a wide range of interfaces, including a keyboard [45, 53], game controller [52], or 3D SpaceMouse [66, 24], which can be limited in fine-grained manipulation. VR and AR-based controllers [113, 25, 15, 117, 115] enable more visually guided interaction but introduce additional hardware requirements and user learning overhead. Custom hardware systems [119, 29, 17, 98] have also been proposed to facilitate data collection, but require specialized equipment.

B. Learning from Human Video

To reduce the cost and effort of robot data collection, prior works have explored learning robot policies from human video demonstrations, which are easier and faster to collect. Several approaches leverage large-scale human video datasets to pretrain visual representations [68, 64, 63, 65], learn object and scene-level affordances [2, 11, 87], or predict robot actions [3, 5, 86]. Other methods extract coarse supervision from human videos, such as keypoints [57, 33, 71, 80], object trajectories [37, 58, 122, 107], correspondence tracks [6, 4], or motion priors [72, 88] to guide robot policy learning. Works have also studied learning from egocentric human video [43, 62, 110], as the demonstrations naturally capture hand-object interactions from a first-person perspective. However, across these approaches, data collection often depends on curated annotations from large-scale datasets, multi-view and RGB-D camera setups, or specialized sensing hardware, which are not always available for new tasks.

C. View and Embodiment Synthesis for Robot Learning

Recent works have explored synthesizing robot-centric wrist-camera views from alternative viewpoints for visuomotor policy learning. WristWorld [76] extends VGGT [95] to generate temporally consistent wrist-view observations from third-person camera inputs. Imagination at Inference [21] fine-tunes ZeroNVS [82] to synthesize auxiliary wrist-camera images during policy execution. RwoR [36] converts wrist-mounted human hand demonstrations into robot end-effector observations using a learned generative model. Other works replace the human embodiment in video demonstrations with a robot using image editing or inpainting [13, 14, 54]. Methods such as Phantom [56] and Masquerade [55] remove the human hand and insert a robot gripper into the scene, producing robot-consistent visual observations directly from human videos.

Neural rendering and view synthesis have also been used to render novel views and augment robot demonstrations from limited real data [92, 102, 109, 38, 104, 70, 118, 121, 112]. These approaches leverage neural representations and diffusion models to synthesize viewpoints, trajectories, or robot executions from a small number of demonstrations, enabling data augmentation without collecting new real-world trajectories.

III. Methodology

Our goal is to transform RGB egocentric human demonstrations into a wrist-camera robotic observation-action dataset to train visuomotor policies for tabletop manipulation tasks. An overview of our pipeline is shown in Fig. 2. The pipeline consists of five stages: data collection, interactive scene initialization, hand-object optimization, wrist-view retargeting and rendering, and policy training and deployment. The system is designed to be portable and easy to use, enabling demonstration collection using only a single monocular RGB camera without specialized hardware or additional sensing.

A. Assumptions

For the purposes of this work, we assume that all objects are rigid. Objects are manipulated in a tabletop setting, with no significant scene changes beyond the hand-object interaction during demonstration. This setup serves as a foundation to validate our proposed approach, with the intention of extending to more complex tasks and diverse scenes in future work.

B. Data Collection

The user first records a short monocular RGB video of the workspace without the manipulated objects. This process is analogous to the environment scan performed by Universal Manipulation Interface (UMI) [17] prior to demonstration data collection, and provides the necessary visual data to reconstruct the scene geometry. Structure-from-Motion (SfM) with Lightglue [60] feature matching is used to estimate camera poses and recover a sparse 3D reconstruction of the environment. Collecting the scan is quick and typically takes less than one minute. The resulting camera poses are used to initialize a 3D Gaussian Splat representation of the scene and to localize subsequent demonstrations (Sec. III-C1).

During demonstration data collection, the user wears a head-mounted egocentric camera and records multiple demonstrations within the workspace. In this work, a GoPro Hero 9 equipped with a standard linear lens, modeled as a pinhole camera, is attached to a helmet and used to record both the scene and the demonstrations, although other monocular RGB cameras and mounting configurations could be substituted.

C. Interactive Scene Initialization

While a reconstruction of the static scene is available from Sec. III-B, the geometry of the hand and objects must also be initialized for trajectory tracking and wrist-view rendering.

1) Interactive Scene Reconstruction: For each demonstration, the interactive scene is first reconstructed to recover hand and object depth, which cannot be obtained from the static scene alone. Demonstration frames are localized within the static scene using Hierarchical Localization [83] and LightGlue feature matching to estimate per-frame camera poses. SpatialTrackerV2 [100] is used to extract temporally consistent monocular depth maps for each frame. Since SfM reconstructions are ambiguous up to a global scale, a scene-level scale alignment is estimated between the SfM reconstruction and the predicted depth maps, and the scene Gaussian Splat is rescaled accordingly. Additional details are provided in Appendix B.

2) Hand Pose Initialization: For initial hand pose estimation, HAMER [74] is used to obtain per-frame hand shape and pose estimates. The hand parameters are refined using a sequence-level optimization that enforces temporal smoothness and consistency with the monocular depth estimates. A complete description can be found in Appendix C.

3) Object Pose Initialization: For object pose initialization, we provide a text description of the manipulated object and apply Grounding DINO [61] to detect the object in the initial frame. SAM2 [79] then generates segmentation masks and propagates them throughout the sequence. An initial mesh of the object is reconstructed using SAM3D [91]. Rather than using the Gaussian representation additionally produced by SAM3D, we construct our own Gaussian Splat of the object by rendering multi-view images of the mesh. This results in higher-fidelity renderings under the camera trajectories observed in our demonstrations. Finally, the reconstructed mesh along with the first-frame segmentation is used by MegaPose [50] to obtain an initial 6D pose estimate.

To refine the object geometry, the initial contact frame is estimated by thresholding the overlap between the hand and object segmentation masks. For frames prior to contact, the object pose and scale are jointly optimized by enforcing consistency and alignment of the segmentation with the monocular depth maps. Full details of the full object pose initialization process are provided in Appendix D.

D. Hand-Object Optimization

To render realistic object-centric views, we need to track the 3D pose of the object. While prior works rely on depth sensors [71, 57, 72], multi-view cameras [80, 33, 56], smart glasses [62], or full 3D object scans [46, 109], our system

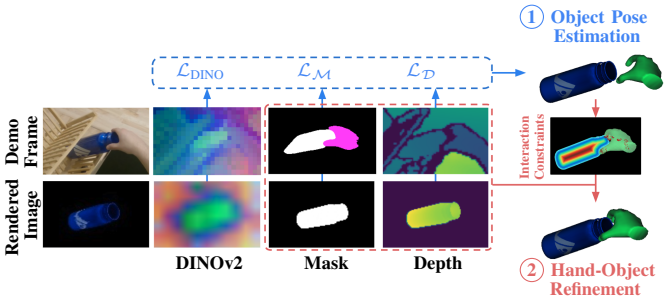

Fig. 3: Overview of hand–object optimization. The object pose is first estimated using supervised mask, depth, and DINOv2 losses. Then the hand and object pose are jointly refined using mask, depth, and interaction constraints.

assumes none of these. As a result, we adopt a hand-object interaction-based approach, leveraging the fact that hand and object motion provide complimentary geometric constraints, particularly when the object is faced with hand-occlusions. Similar to prior works [35, 73, 27, 123], we utilize two stages (Fig. 3), where the object pose is first estimated independently and then jointly refined together with the hand.

1) Object Pose Estimation: Given the object mesh with vertices V^{obj} and faces F^{obj}, object masks and monocular depth maps $\hat{\mathcal{M}}^{obj}$, $\hat{\mathcal{D}}^{obj}$, hand masks $\hat{\mathcal{M}}^{hand}$, and the initialized pose $(\mathbf{R}_0^{obj}, \mathbf{t}_0^{obj})$ from Sec. III-C, the object pose is estimated sequentially for each frame of the demonstration sequence using differentiable rendering. For frame t, a differentiable rasterizer $\mathcal{R}$ [51] is used to render the image, mask, and depth of the object.

$$\mathcal{I}_t^{obj}, \mathcal{M}_t^{obj}, \mathcal{D}_t^{obj} = \mathcal{R}(\mathbf{R}_t^{obj} V^{obj} + \mathbf{t}_t^{obj}, F^{obj}) \tag{1}$$

An occlusion-aware mask loss [116] is first calculated between the rendered and SAM2 predicted masks.

$$\mathcal{L}_{\mathcal{M}_{obj}} = \|(\mathcal{M}_t^{obj} - \hat{\mathcal{M}}_t^{obj}) \odot (1 - \hat{\mathcal{M}}_t^{hand})\| \tag{2}$$

Because similar object masks can be produced by different poses, particularly when faced with hand occlusions, we additionally utilize a depth consistency loss that encourages consistency between the rendered and predicted depths.

$$\mathcal{L}_{\mathcal{D}_{obj}} = \|(\mathcal{D}_t^{obj} - \hat{\mathcal{D}}_t^{obj}) \odot (1 - \hat{\mathcal{M}}_t^{hand})\|_2^2 \tag{3}$$

To account for sparsity and noise in monocular depth predictions, we additionally incorporate image-based feature supervision. We extract DINOv2 [69] features from the rendered image $\mathcal{F}_t$ and the masked original frame $\hat{\mathcal{F}}_t$ and compute a cosine similarity loss.

$$\begin{aligned} \mathcal{F}_t &= \mathcal{G}(\mathcal{I}_t), \\ \hat{\mathcal{F}}_t &= \mathcal{G}(\hat{\mathcal{I}}_t \odot \mathcal{M}_t^{obj}), \\ \mathcal{L}_{\text{DINO}} &= \frac{1}{P} \sum_{p=1}^{P} \left(1 - \frac{\mathcal{F}_{t,p} \cdot \hat{\mathcal{F}}_{t,p}}{\|\mathcal{F}_{t,p}\|_2 \, \|\hat{\mathcal{F}}_{t,p}\|_2}\right) \end{aligned} \tag{4}$$

where $\hat{\mathcal{I}}_t$ is the original image, and $\mathcal{G}$ denotes the DINOv2 network with a ViT-S [23] backbone, using features extracted from the ninth layer. The final optimization formulation is

$$\min_{\mathbf{R}_t^{obj}, \mathbf{t}_t^{obj}} \lambda_{\mathcal{M}_{obj}} \mathcal{L}_{\mathcal{M}_{obj}} + \lambda_{\mathcal{D}_{obj}} \mathcal{L}_{\mathcal{D}_{obj}} + \lambda_{\text{DINO}} \mathcal{L}_{\text{DINO}} \tag{5}$$

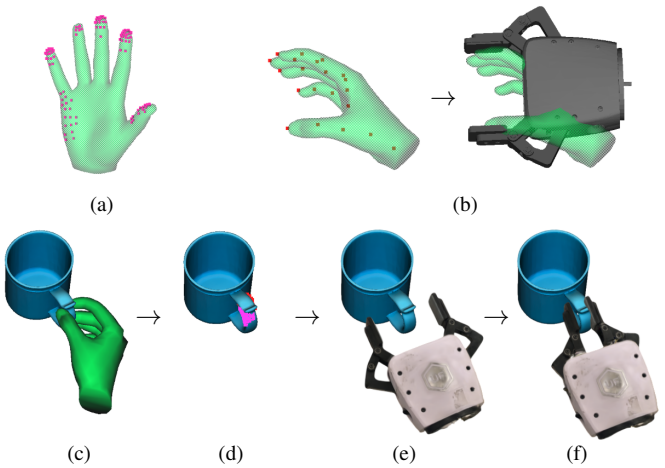

Fig. 4: (a) Frequently contacted hand vertices. (b) Pose mapping from hand to end-effector using thumb and index joints. At the contact start frame (c), the closest 50 object points to the thumb and fingertips are identified (d) and used to refine the initial gripper pose and width (e) into a configuration that establishes contact with the points (f).

Each pose is optimized independently per frame, using the pose from the previous frame as initialization. Once all poses are optimized, the contact start and end frames (t_s, t_e) are identified based on whether the object's translation or rotation exceed predefined thresholds for a fixed number of consecutive frames. A full description can be found in Appendix E.

2) Joint Hand-Object Refinement: Because estimating hand and object poses independently can be inaccurate, we jointly optimize both to exploit hand–object interaction constraints. To jointly refine the hand and object poses, all parameters are optimized simultaneously across all frames. We denote the object parameters as $\mathbf{\Theta}^{obj}$, which consists of the object rotations $\mathbf{R}^{obj}$ and translations $\mathbf{t}^{obj}$ for all frames in addition to a global scaling factor s^{obj}. The object is assumed to remain stationary for all $t <= t_s$ and $t >= t_e$. In addition, we optimize the MANO [81] hand parameters $\mathbf{\Theta}^{hand}$, which include the global hand rotations $\mathbf{R}^{hand}$, global hand translations $\mathbf{t}^{hand}$, hand pose parameters $\boldsymbol{\theta}$, and shared hand shape parameters $\boldsymbol{\beta}$. Our joint hand-object optimization uses the following loss functions to enforce visual and interactive constraints:

Occlusion-Aware Mask Loss. Similar to Sec. III-D1, we use a differentiable rasterizer to render masks and depths for both the object and the hand across all frames.

$$\begin{aligned} M^{obj}, D^{obj} &= \mathcal{R}(\mathbf{R}^{obj} V^{obj} + \mathbf{t}^{obj}, F^{obj}) \\ M^{hand}, D^{hand} &= \mathcal{R}(\mathbf{R}^{hand} V^{hand} + \mathbf{t}^{hand}, F^{hand}) \end{aligned} \tag{6}$$

The hand mesh is obtained from the MANO hand model using the current hand pose and shape parameters.

$$V^{hand}, F^{hand} = \text{MANO}(\boldsymbol{\theta}, \boldsymbol{\beta}) \tag{7}$$

We compute occlusion-aware mask losses that account for mutual hand-object occlusions.

$$\begin{aligned} \mathcal{L}_{\mathcal{M}_{obj}} &= \|(\mathcal{M}^{obj} - \hat{\mathcal{M}}^{obj}) \odot (1 - \hat{\mathcal{M}}^{hand})\| \\ \mathcal{L}_{\mathcal{M}_{hand}} &= \|(\mathcal{M}^{hand} - \hat{\mathcal{M}}^{hand}) \odot (1 - \hat{\mathcal{M}}^{obj})\| \\ \mathcal{L}_{\mathcal{M}} &= \mathcal{L}_{\mathcal{M}_{obj}} + \mathcal{L}_{\mathcal{M}_{hand}} \end{aligned} \tag{8}$$

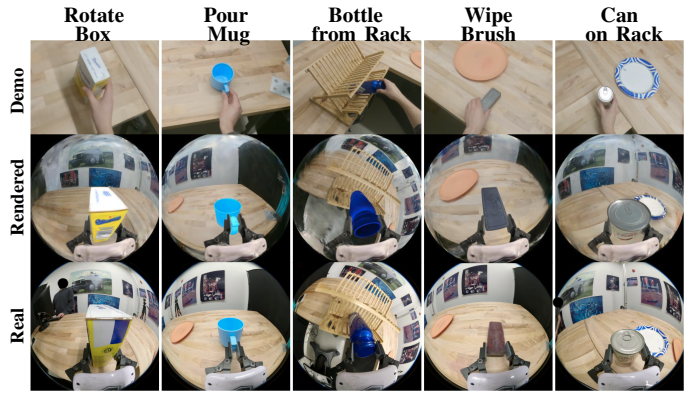

Fig. 5: Example demonstration frames, original wrist-view renders, and real rollout images at initial contact.

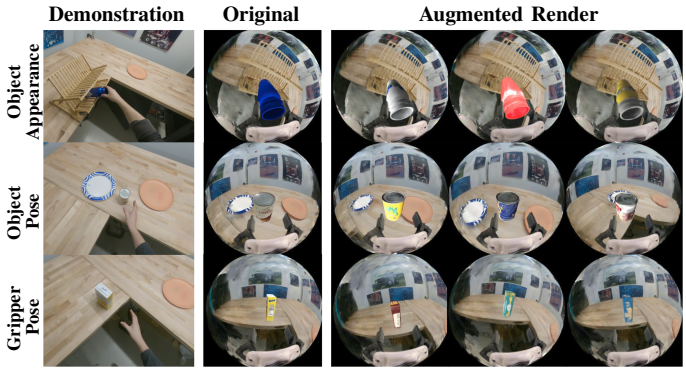

Fig. 6: Example demonstration frames and wrist-view renders illustrating object appearance, object pose, and gripper pose augmentations.

Depth Loss. We apply the same depth loss from Eq. 3 to both the object and the hand for all frames.

$$\begin{aligned}\mathcal{L}_{\mathcal{D}_{obj}} &= \|(\mathcal{D}^{obj} - \hat{\mathcal{D}}^{obj}) \odot (1 - \hat{\mathcal{M}}^{hand})\|_2^2 \\ \mathcal{L}_{\mathcal{D}_{hand}} &= \|(\mathcal{D}^{hand} - \hat{\mathcal{D}}^{hand}) \odot (1 - \hat{\mathcal{M}}^{obj})\|_2^2 \\ \mathcal{L}_{\mathcal{D}} &= \mathcal{L}_{\mathcal{D}_{obj}} + \mathcal{L}_{\mathcal{D}_{hand}}\end{aligned} \tag{9}$$

Contact Loss. Inspired by previous work [34, 73, 27, 59], proximity is encouraged between frequently contacted hand vertices $V^{tip} \subset V^{hand}$ (Fig. 4(a)) and object vertices during contact frames to promote interaction.

$$\mathcal{L}_{\text{contact}}(t) = \begin{cases} \sum_{\mathbf{v}^\tau \in V^{tip}} \min_{\mathbf{v}^o \in V^{obj}} \|\mathbf{v}_t^\tau - \mathbf{v}_t^o\|_2^2, & t_s <= t <= t_e, \\ 0, & \text{otherwise} \end{cases} \tag{10}$$

Collision Loss. To discourage physically implausible interpenetration between the hand and the object during contact, we penalize hand vertices that fall inside the object volume. We precompute a truncated signed distance field (TSDF) for the object mesh and evaluate it at each hand vertex

$$\mathcal{L}_{\text{col}} = \sum_{\mathbf{v}^h \in V^{hand}} \Phi^{obj}(\mathbf{v}^h) \tag{11}$$

where $\Phi^{obj}(\cdot)$ denotes the object TSDF evaluated at hand vertex $\mathbf{v}^h \in V^{hand}$.

Stable Grasp Loss Following Zhu *et al.* [123], we use a stable grasp loss that encourages finger tip vertices $\mathbf{v}^\tau \in V^{tip}$ to maintain consistent distances to the object vertices $\mathbf{v}^o \in V^{obj}$ during contact

$$\begin{aligned}\mathcal{L}_{\text{sg}} &= \sum_{\mathbf{v}^\tau} \sum_{\mathbf{v}^o} \sum_{n=t_s}^{t_e} \sum_{m=t_s}^{t_e} \|d_n^{o\tau} - d_m^{o\tau}\| \\ d_n^{o\tau} &:= \|\mathbf{v}_n^\tau - \mathbf{v}_n^o\|_2^2\end{aligned} \tag{12}$$

The final joint optimization minimizes

$$\begin{aligned}\min_{\boldsymbol{\Theta}} \quad & \lambda_{\mathcal{M}}\mathcal{L}_{\mathcal{M}} + \lambda_{\mathcal{D}}\mathcal{L}_{\mathcal{D}} + \lambda_c\mathcal{L}_{\text{contact}} \\ & + \lambda_{\text{col}}\mathcal{L}_{\text{col}} + \lambda_{\text{sg}}\mathcal{L}_{\text{sg}} + \mathcal{L}_{\text{aux}}\end{aligned} \tag{13}$$

where $\boldsymbol{\Theta} = \{\boldsymbol{\Theta}^{obj}, \boldsymbol{\Theta}^{hand}\}$ and $\mathcal{L}_{\text{aux}}$ represents additional auxiliary loss terms described in Appendix E.

E. Retargeting and Rendering

Once the hand and object poses have been recovered through joint optimization, the human hand motion is retargeted into executable robot end-effector trajectories, and the corresponding wrist-camera observations are rendered.

First, a sparse set of evenly spaced keyframes are extracted from the full hand-object trajectory. For keyframes prior to contact ($t < t_s$), the gripper remains open and the end-effector pose is mapped from the human-hand pose using the thumb and index joints, as shown in Fig. 4(b) and discussed in more detail in Appendix F. Following Pan *et al.* [70], we apply pre-contact trajectory optimization to prevent unintended gripper-object collisions. The optimization is formulated as

$$\min_{\mathbf{T}_{t<t_s}^{ee}} \lambda_{\text{funnel}}\mathcal{L}_{\text{funnel}} + \lambda_{\text{col}}\mathcal{L}_{\text{col}} + \lambda_{\text{smooth}}\mathcal{L}_{\text{smooth}} \tag{14}$$

where $\mathcal{L}_{\text{funnel}}$ constrains trajectories to remain close to the original motion, $\mathcal{L}_{\text{col}}$ prevents gripper-object collision by applying the same TSDF-based loss as Eq. 11, and $\mathcal{L}_{\text{smooth}}$ encourages temporally smooth motion.

At the contact start frame ($t = t_s$), the end-effector pose $\mathbf{T}_{t_s}^{ee}$ and gripper width are refined to ensure a physically plausible grasp (Fig. 4(c-f)). For both the thumb and index finger tips, 50 contact points are identified based on mesh vertex proximity. Using these contact points, the end-effector position and gripper width are optimized to align the grasp with the hand-held object. Additional details are provided in Appendix F.

During contact ($t_s \leq t \leq t_e$), the pose of the gripper and object are assumed to be relatively fixed. For subsequent frames, the end-effector pose is updated by applying the relative object motion with respect to the contact frame,

$$\mathbf{T}_t^{ee} = \mathbf{T}_{t_s}^{ee} \left(\mathbf{T}_{t_s}^{obj}\right)^{-1} \mathbf{T}_t^{obj} \tag{15}$$

ensuring that the end-effector rigidly follows the object motion while preserving the established grasp.

Finally, the end-effector and object poses defined at keyframes are interpolated to produce smooth continuous trajectories. Wrist-camera views of the retargeted end-effector trajectories are rendered by combining the Gaussian Splats of the scene, object, and end-effector. We render fisheye images using Nerfstudio's [89, 106] 3DGUT [99] implementation. Example renderings are shown in Fig. 5.

F. Policy Training

We train a diffusion policy [16, 17] conditioned on wrist-view observations and robot proprioceptive inputs to generate relative pose and gripper action chunks. Similar to prior work [10, 78], we add Gaussian noise to the input images during training to mitigate the sim-to-real gap between rendered wrist-view observations and real robot images. To increase dataset diversity, we augment the demonstrations as shown in Fig. 6. Augmentations include retexturing object meshes, randomly translating the object in the scene, randomizing the initial gripper pose, randomly scaling the scene, and perturbing both the wrist-camera intrinsics and extrinsics.

IV. Experiments and Results

A. Experimental Setup

We evaluate the performance of our framework across five different tabletop manipulation tasks. All experiments are conducted using a UFactory xArm7 robot equipped with an xArm G1 Gripper. For human demonstration data collection, a GoPro Hero9 is attached to a helmet (Fig. 7(a)) worn by the user. For policy rollout, a GoPro Hero 9 with Max Lens Mod 1.0 is mounted above the gripper to capture wrist-view observations during execution, as shown in Fig. 7(b).

For each task, we collect 30 demonstrations within a fixed working space using a single object per task. Each demonstration is captured at 30 Hz and processed through WARPED to generate end-effector trajectories and wrist-camera renderings. The demonstrations are augmented 10 times following Sec. III-F. A diffusion policy is trained using four NVIDIA Tesla V100s. Training details are provided in Appendix G.

Each trained policy is evaluated over 20 trials per task with a rollout frequency of 10Hz. Performance is measured using success rate, defined as the number of trials in which the task is successfully completed. At the start of each evaluation trial, the robot is initialized in a task-dependent configuration such that the manipulated object is within the wrist-camera field of view and in reasonable proximity of the end-effector, matching the human hand conditions observed during demonstration.

B. Task Descriptions

We evaluate our method on five tabletop manipulation tasks (Fig. 8). Demonstrations for each task are collected using the same object, while object placement varies across trials. We briefly describe each task below:

a) **Rotate Box:** The robot must rotate a box 90 degrees onto a new face. The box is placed at different initial positions across trials. This task tests the ability to track and manipulate object pose across rotations.

b) **Pour Mug:** The robot must grasp a mug by its handle and execute a pouring motion. The mug is placed at different starting positions. This task tests tracking object rotation and grasping a specific part of the object.

c) **Take Bottle out of Dish Rack:** The robot must pick up a bottle from the middle shelf of a dish rack. The bottle position within the rack varies between trials. This task tests manipulation in a constrained setting.

Fig. 7: (a) GoPro Hero9 attached to a helmet to record human demonstrations. (b) GoPro Hero9 with Max Lens Mod 1.0 mounted above gripper. (c) Objects used for data collection. (d) Novel objects used for evaluation.

d) **Wipe Plate with Brush:** The robot must pick up a brush and wipe a plate. This task is challenging because the brush is small and lies flat on the table, requiring reasonably accurate pose estimation and grasping.

e) **Pick and Place Can on Plate:** The robot must pick up a can and place it onto a plate. Unlike the Wipe Brush task, the plate is not fixed and varies in position, requiring WARPED to localize multiple objects and render both the can and the plate in the scene.

C. Baselines

For all tasks, we compare our method against a teleoperation baseline, where policies are trained on teleoperated demonstrations using a Meta Quest 2 VR headset and controller. We additionally adopt the Alter baseline presented by Heng *et al.* [36]. Following this setup, the same GoPro is mounted on a human hand to approximate the robot end-effector pose. Demonstrations are recorded, inpainted using InpaintAnything [111], and overlaid with masked gripper images to simulate wrist-view observations. End-effector trajectories are extracted using the localization method in Sec. III-C1, and gripper open/close states are manually annotated. We note that RwoR [36] is a relevant baseline but no public code or trained models were available at the time of testing.

D. Results

How well do policies trained using WARPED perform compared to teleoperation and baselines? We evaluate the performance of our framework where training and evaluation are performed in a single scene using the same objects observed during training. The results are shown in Table I. Overall, WARPED achieves performance comparable to teleoperation on the Pour Mug, Bottle from Rack, and Can on Plate tasks, despite relying solely on monocular and egocentric human video demonstrations. Notably, WARPED outperforms teleoperation on the Rotate Box task. This is because precise and consistent rotational control was found to be difficult using teleoperation, highlighting the benefit of using natural hand motion for manipulation. However, WARPED underperforms teleoperation on Wipe Brush. This is because the brush is small and level to the table (Fig. 8), making pose estimation

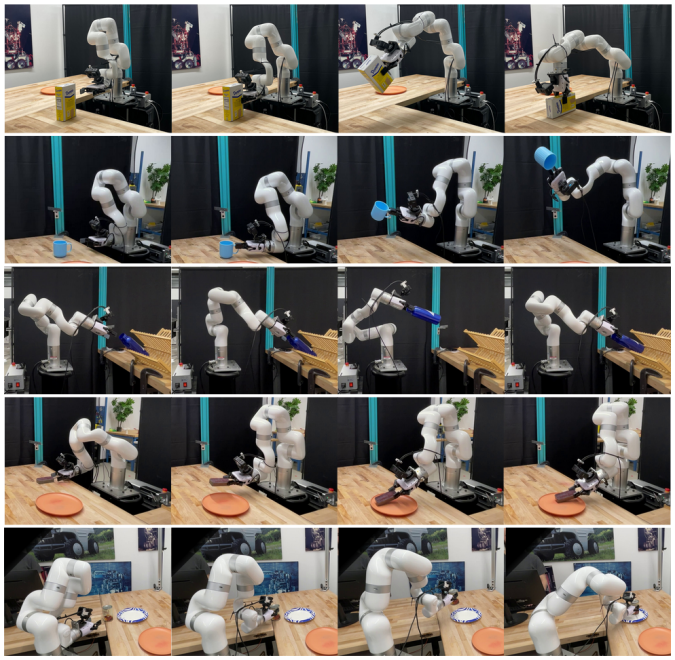

Fig. 8: Real-world rollouts for all evaluated tasks.

and retargeting more difficult. The Alter baseline performs poorly across all tasks, demonstrating that naively rendering wrist-view images without accurate hand-object geometry is insufficient for policy learning.

How well do learned policies generalize to novel objects? We assess WARPED's ability to generalize to novel objects not seen during training. For each task, policies are evaluated on two unseen objects of the same category (Fig. 7(d)). The results are reported in Table II. WARPED consistently outperforms teleoperation on novel objects for all tasks except Wipe Brush. The performance difference is most evident on Obj 2 for the Rotate Box and Bottle from Rack tasks, indicating WARPED's robustness to changes in object geometry and size.

How well do learned policies generalize to out-of-distribution scenes? We examine WARPED's ability to generalize to out-of-distribution scenes. For the Can on Plate task, we collect 50 demonstrations across 20 diverse tabletop settings and evaluate on four unseen scenes for a total of 20 trials. Visualizations of the training and testing scenes can be found in Appendix H. WARPED achieves a success rate of 16/20. The variation across scenes is likely due to subtle differences in the training data, including lighting conditions and scene layout. Overall, these results show that WARPED can generalize to out-of-distribution scenes. Performance could be improved by integrating scene-level augmentation strategies [109, 104, 70].

How robust is WARPED to background distractors? We test the robustness of WARPED when presented with background visual distractors that are not in the initial Gaussian Splat of the scene (Fig. 9). As shown in Table I, background distractors have negligible impact on Can on Plate and Bottle from Rack tasks, and result in a small drop in performance for Rotate Box and Wipe Brush. The largest degradation is observed for Pour Mug, where failures are primarily due to missed grasps.

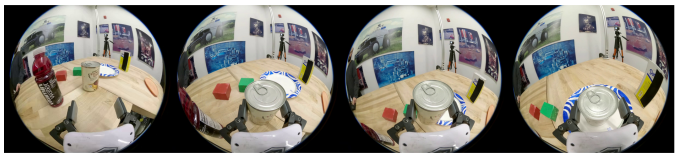

Fig. 9: Example of background distractor rollout for Can on Plate task.

How efficient is data collection using WARPED? We assess the data collection efficiency of WARPED and compare it against the time required to collect teleoperated data. We report the total time needed to collect all demonstrations in Table III. For WARPED, this includes the initial scan of the scene. Similarly, for teleoperation, we report the total time, including environment resets and any additional overhead incurred during operation. Data collection with WARPED is approximately 5–8x faster than teleoperation across all tasks. This highlights one of WARPED's advantages for collecting new datasets, and suggests potential for more scalable data collection as complexity and dataset sizes increase.

Can WARPED complement teleoperated data? We evaluate whether WARPED demonstrations can be co-trained with teleoperated data. For each task, we combine 15 teleoperated and 15 WARPED demonstrations, including all augmentations. To align human and robot trajectories, human demonstrations are slowed down by a factor of three. The results are shown in Table I. For Rotate Box, Pour Mug, Bottle from Rack, and Can on Plate, co-training achieves performance comparable or better than training with teleoperation alone. However, for Wipe Brush, performance decreases under co-training. We observe that this task involves more complex motion, and the rendered trajectories from WARPED are more dissimilar from the teleoperated executions, leading to data inconsistencies.

Qualitatively, co-trained policies exhibit behaviors more similar to teleoperated executions, despite much of the training data coming from WARPED. This shows that by leveraging diffusion policies, we can achieve teleoperation-like performance with fewer demonstrations by supplementing them with faster-to-collect WARPED data.

How does WARPED compare to UMI-based data collection? We compare data collection using WARPED against UMI on the Can on Plate and Rotate Box tasks. Policies are trained on 30 demonstrations and evaluated over 20 trials, with 10 using the demonstration object and 10 using a novel object. UMI is evaluated on a UR5 equipped with its custom gripper.

Results are shown in Table IV. For Can on Plate, the policy trained on UMI data achieves near-perfect success on both training and novel objects. This is likely because of the similar overhead visual appearance of cylindrical aluminum cans, resulting in consistent and representative demonstrations. In contrast, WARPED exhibits a slightly lower success, possibly as a result of the noise in object pose tracking which introduces minor variability in the demonstrations.

For Rotate Box, UMI performs substantially worse, achieving only 2/10 successes on the training object and failing on the novel object, while WARPED achieves near-perfect success. Rotating a box onto a new face requires precise,

TABLE I: Success rates comparing teleoperation, the Alter baseline, and WARPED.

Method	Rotate Box	Pour Mug	Bottle from Rack	Wipe Brush	Can on Plate
Teleoperation	16/20	19/20	16/20	**15/20**	19/20
Alter (Heng *et al.*)	7/20	3/20	0/20	0/20	8/20
WARPED (no augmentation)	0/20	17/20	0/20	0/20	8/20
WARPED (background distractors)	18/20	15/20	**17/20**	9/20	17/20
WARPED	**20/20**	18/20	**17/20**	11/20	17/20
Teleoperation + WARPED	19/20	**20/20**	**17/20**	11/20	**20/20**

TABLE II: Success rates on novel object instances.

Method	Novel Obj	Rotate Box	Bottle from Rack	Wipe Brush	Can on Plate
Teleoperation	Obj 1	8/10	4/10	**7/10**	9/10
	Obj 2	2/10	2/10	4/10	**9/10**
WARPED	Obj 1	**10/10**	**8/10**	**7/10**	**10/10**
	Obj 2	**8/10**	**5/10**	2/10	**9/10**

TABLE III: Demonstration collection time (MM:SS).

Method	Rotate Box	Pour Mug	Bottle from Rack	Wipe Brush	Can on Plate
Teleoperation	22:51	24:59	31:49	30:16	15:20
WARPED	**3:37**	**3:18**	**3:27**	**5.19**	**3.41**

continuous rotational motion and is sensitive to slipping, which is difficult to demonstrate consistently with end-effector tools. In contrast, human hand demonstrations capture smoother rotational trajectories, and the geometric variation among novel boxes further benefits WARPED's augmentation strategy. UMI's framework does not support such augmentation, limiting policy training to the original 30 demonstrations.

How do augmented demonstrations benefit WARPED? We evaluate the impact of data augmentation from Sec. III-F by removing augmented demonstrations from training. As shown in Table I, data augmentation has a significant effect on performance. Without augmentation, WARPED fails entirely on three of five tasks, and performance of Can on Plate drops by more than 50%. This suggests that augmentation helps bridge sim-to-real discrepancies.

How does hand–object optimization compare to object pose tracking? We qualitatively compare the effect of hand-object optimization from Sec. III-D against object pose tracking using FoundationPose [96]. For the Rotate Box and Can on Plate tasks, we run both hand-object optimization and object pose tracking and visually assess whether the resulting object poses are reasonable over the course of the trajectory. Visualizations of the results can be seen in Appendix I.

Using hand-object optimization results in success rates of 17/20 for both the Rotate Box and Can on Plate tasks, whereas FoundationPose achieves success rates of 11/20 and 2/20 respectively. The poor performance of FoundationPose is likely because of noise in monocular depth estimates, inaccuracies in camera intrinsics, and predicted mesh misalignment. We additionally observe that FoundationPose fails predominantly on the Can on Plate task because the object is small and occluded by the hand, leading to unrecoverable tracking failures. In contrast, hand-object optimization exploits interaction constraints, improving pose tracking accuracy through occlusions.

TABLE IV: Success rates comparing UMI and WARPED.

Method	Object	Can on Plate	Rotate Box
UMI	Train	**9/10**	2/10
	Novel	**10/10**	0/10
WARPED	Train	8/10	**10/10**
	Novel	9/10	**10/10**

V. Discussion and Limitations

We demonstrate that visuomotor policies can be trained from human egocentric demonstrations using only a monocular RGB camera, substantially reducing demonstration collection time compared to teleoperation. On four of five tasks, WARPED achieves performance comparable to teleoperation, and outperforms both teleoperation and UMI on Rotate Box, where smooth rotational motion is easier to demonstrate with a human hand. Another advantage of our approach is the ability to render synthetic wrist-view observations, enabling data augmentation and improving robustness on novel objects. Finally, combining teleoperation with WARPED demonstrations achieves comparable performance while reducing data collection time relative to teleoperation alone.

Although our approach shows competitive performance with reduced data collection overhead, we acknowledge several important limitations. First, WARPED currently only supports rigid object motion. Extending the method to articulated objects could leverage recent works that distill DINOv2 features to track articulated motion [46, 109], while deformable objects may be addressed by integrating deformable Gaussian representations [26]. Next, despite robustness to background distractors, the method assumes a quasi-static scene in which only the manipulated object moves. Adapting to moving scenes would likely require integrating dynamic Gaussian Splatting representations [97, 105]. Additionally, the pipeline fails when the manipulated object becomes fully occluded, which is also a limitation of state-of-the-art object tracking methods.

VI. Conclusion

In this paper, we present WARPED, a framework for learning visuomotor manipulation policies from monocular human demonstration videos by synthesizing realistic wrist-view robot observations and retargeted end-effector trajectories. By combining hand–object tracking, scene reconstruction, and photorealistic rendering, WARPED enables policy training using standard RGB data without multiview setups, depth sensors, or custom hardware. Future work will focus on extending WARPED to articulated objects, dynamic scenes, and longer-horizon manipulation tasks.

References

[1] Cihan Acar, Kuluhan Binici, Alp Tekirdağ, and Yan Wu. Visual-policy learning through multi-camera view to single-camera view knowledge distillation for robot manipulation tasks. *IEEE Robotics and Automation Letters*, 9(1):691–698, January 2024. ISSN 2377-3774. doi: 10.1109/lra.2023.3336245. URL http://dx.doi.org/10.1109/LRA.2023.3336245.

[2] Shikhar Bahl, Russell Mendonca, Lili Chen, Unnat Jain, and Deepak Pathak. Affordances from human videos as a versatile representation for robotics, 2023. URL https://arxiv.org/abs/2304.08488.

[3] Homanga Bharadhwaj, Abhinav Gupta, Shubham Tulsiani, and Vikash Kumar. Zero-shot robot manipulation from passive human videos, 2023. URL https://arxiv.org/abs/2302.02011.

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

[5] Homanga Bharadhwaj, Abhinav Gupta, Vikash Kumar, and Shubham Tulsiani. Towards generalizable zero-shot manipulation via translating human interaction plans. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6904–6911, 2024. doi: 10.1109/ICRA57147.2024.10610288.

[6] Homanga Bharadhwaj, Roozbeh Mottaghi, Abhinav Gupta, and Shubham Tulsiani. Track2act: Predicting point tracks from internet videos enables generalizable robot manipulation, 2024. URL https://arxiv.org/abs/2405.01527.

[7] Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Lucy Xiaoyang Shi, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. π_0: A vision-language-action flow model for general robot control, 2026. URL https://arxiv.org/abs/2410.24164.

[8] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, Pete Florence, Chuyuan Fu, Montse Gonzalez Arenas, Keerthana Gopalakrishnan, Kehang Han, Karol Hausman, Alexander Herzog, Jasmine Hsu, Brian Ichter, Alex Irpan, Nikhil Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Lisa Lee, Tsang-Wei Edward Lee, Sergey Levine, Yao Lu, Henryk Michalewski, Igor Mordatch, Karl Pertsch, Kanishka Rao, Krista Reymann, Michael Ryoo, Grecia Salazar, Pannag Sanketi, Pierre Sermanet, Jaspiar Singh, Anikait Singh, Radu Soricut, Huong Tran, Vincent Vanhoucke, Quan Vuong, Ayzaan Wahid, Stefan Welker, Paul Wohlhart, Jialin Wu, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. Rt-2: Vision-language-action models transfer web knowledge to robotic control, 2023. URL https://arxiv.org/abs/2307.15818.

[9] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, Julian Ibarz, Brian Ichter, Alex Irpan, Tomas Jackson, Sally Jesmonth, Nikhil J Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Kuang-Huei Lee, Sergey Levine, Yao Lu, Utsav Malla, Deeksha Manjunath, Igor Mordatch, Ofir Nachum, Carolina Parada, Jodilyn Peralta, Emily Perez, Karl Pertsch, Jornell Quiambao, Kanishka Rao, Michael Ryoo, Grecia Salazar, Pannag Sanketi, Kevin Sayed, Jaspiar Singh, Sumedh Sontakke, Austin Stone, Clayton Tan, Huong Tran, Vincent Vanhoucke, Steve Vega, Quan Vuong, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. Rt-1: Robotics transformer for real-world control at scale, 2023. URL https://arxiv.org/abs/2212.06817.

[10] Arunkumar Byravan, Jan Humplik, Leonard Hasenclever, Arthur Brussee, Francesco Nori, Tuomas Haarnoja, Ben Moran, Steven Bohez, Fereshteh Sadeghi, Bojan Vujatovic, and Nicolas Heess. Nerf2real: Sim2real transfer of vision-guided bipedal motion skills using neural radiance fields, 2022. URL https://arxiv.org/abs/2210.04932.

[11] Hanzhi Chen, Boyang Sun, Anran Zhang, Marc Pollefeys, and Stefan Leutenegger. Vidbot: Learning generalizable 3d actions from in-the-wild 2d human videos for zero-shot robotic manipulation, 2025. URL https://arxiv.org/abs/2503.07135.

[12] Haonan Chen, Cheng Zhu, Shuijing Liu, Yunzhu Li, and Katherine Rose Driggs-Campbell. Tool-as-interface: Learning robot policies from observing human tool use. In *Proceedings of Robotics: Conference on Robot Learning (CoRL)*, 2025.

[13] Lawrence Yunliang Chen, Kush Hari, Karthik Dharmarajan, Chenfeng Xu, Quan Vuong, and Ken Goldberg. Mirage: Cross-embodiment zero-shot policy transfer with cross-painting, 2024. URL https://arxiv.org/abs/2402.19249.

[14] Lawrence Yunliang Chen, Chenfeng Xu, Karthik Dharmarajan, Muhammad Zubair Irshad, Richard Cheng, Kurt Keutzer, Masayoshi Tomizuka, Quan Vuong, and Ken Goldberg. Rovi-aug: Robot and viewpoint augmentation for cross-embodiment robot learning, 2024. URL https://arxiv.org/abs/2409.03403.

[15] Sirui Chen, Chen Wang, Kaden Nguyen, Li Fei-Fei, and C. Karen Liu. Arcap: Collecting high-quality human demonstrations for robot learning with augmented reality feedback, 2024. URL https://arxiv.org/abs/2410.

08464.

[16] Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. In *Proceedings of Robotics: Science and Systems (RSS)*, 2023.

[17] Cheng Chi, Zhenjia Xu, Chuer Pan, Eric Cousineau, Benjamin Burchfiel, Siyuan Feng, Russ Tedrake, and Shuran Song. Universal manipulation interface: In-the-wild robot teaching without in-the-wild robots. In *Proceedings of Robotics: Science and Systems (RSS)*, 2024.

[18] Ian Chuang, Andrew Lee, Dechen Gao, M-Mahdi Naddaf-Sh, and Iman Soltani. Active vision might be all you need: Exploring active vision in bimanual robotic manipulation, 2025. URL https://arxiv.org/abs/2409.17435.

[19] Open X-Embodiment Collaboration, Abby O'Neill, Abdul Rehman, Abhinav Gupta, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, Ajinkya Jain, Albert Tung, Alex Bewley, Alex Herzog, Alex Irpan, Alexander Khazatsky, Anant Rai, Anchit Gupta, Andrew Wang, Andrey Kolobov, Anikait Singh, Animesh Garg, Aniruddha Kembhavi, Annie Xie, Anthony Brohan, Antonin Raffin, Archit Sharma, Arefeh Yavary, Arhan Jain, Ashwin Balakrishna, Ayzaan Wahid, Ben Burgess-Limerick, Beomjoon Kim, Bernhard Schölkopf, Blake Wulfe, Brian Ichter, Cewu Lu, Charles Xu, Charlotte Le, Chelsea Finn, Chen Wang, Chenfeng Xu, Cheng Chi, Chenguang Huang, Christine Chan, Christopher Agia, Chuer Pan, Chuyuan Fu, Coline Devin, Danfei Xu, Daniel Morton, Danny Driess, Daphne Chen, Deepak Pathak, Dhruv Shah, Dieter Büchler, Dinesh Jayaraman, Dmitry Kalashnikov, Dorsa Sadigh, Edward Johns, Ethan Foster, Fangchen Liu, Federico Ceola, Fei Xia, Feiyu Zhao, Felipe Vieira Frujeri, Freek Stulp, Gaoyue Zhou, Gaurav S. Sukhatme, Gautam Salhotra, Ge Yan, Gilbert Feng, Giulio Schiavi, Glen Berseth, Gregory Kahn, Guangwen Yang, Guanzhi Wang, Hao Su, Hao-Shu Fang, Haochen Shi, Henghui Bao, Heni Ben Amor, Henrik I Christensen, Hiroki Furuta, Homanga Bharadhwaj, Homer Walke, Hongjie Fang, Huy Ha, Igor Mordatch, Ilija Radosavovic, Isabel Leal, Jacky Liang, Jad Abou-Chakra, Jaehyung Kim, Jaimyn Drake, Jan Peters, Jan Schneider, Jasmine Hsu, Jay Vakil, Jeannette Bohg, Jeffrey Bingham, Jeffrey Wu, Jensen Gao, Jiaheng Hu, Jiajun Wu, Jialin Wu, Jiankai Sun, Jianlan Luo, Jiayuan Gu, Jie Tan, Jihoon Oh, Jimmy Wu, Jingpei Lu, Jingyun Yang, Jitendra Malik, João Silvério, Joey Hejna, Jonathan Booher, Jonathan Tompson, Jonathan Yang, Jordi Salvador, Joseph J. Lim, Junhyek Han, Kaiyuan Wang, Kanishka Rao, Karl Pertsch, Karol Hausman, Keegan Go, Keerthana Gopalakrishnan, Ken Goldberg, Kendra Byrne, Kenneth Oslund, Kento Kawaharazuka, Kevin Black, Kevin Lin, Kevin Zhang, Kiana Ehsani, Kiran Lekkala, Kirsty Ellis, Krishan Rana, Krishnan Srinivasan, Kuan Fang, Kunal Pratap Singh, Kuo-Hao Zeng, Kyle Hatch, Kyle Hsu, Laurent Itti, Lawrence Yunliang Chen, Lerrel Pinto, Li Fei-Fei, Liam Tan, Linxi "Jim" Fan, Lionel Ott, Lisa Lee, Luca Weihs, Magnum Chen, Marion Lepert, Marius Memmel, Masayoshi Tomizuka, Masha Itkina, Mateo Guaman Castro, Max Spero, Maximilian Du, Michael Ahn, Michael C. Yip, Mingtong Zhang, Mingyu Ding, Minho Heo, Mohan Kumar Srirama, Mohit Sharma, Moo Jin Kim, Muhammad Zubair Irshad, Naoaki Kanazawa, Nicklas Hansen, Nicolas Heess, Nikhil J Joshi, Niko Suenderhauf, Ning Liu, Norman Di Palo, Nur Muhammad Mahi Shafiullah, Oier Mees, Oliver Kroemer, Osbert Bastani, Pannag R Sanketi, Patrick "Tree" Miller, Patrick Yin, Paul Wohlhart, Peng Xu, Peter David Fagan, Peter Mitrano, Pierre Sermanet, Pieter Abbeel, Priya Sundaresan, Qiuyu Chen, Quan Vuong, Rafael Rafailov, Ran Tian, Ria Doshi, Roberto Mart'in-Mart'in, Rohan Baijal, Rosario Scalise, Rose Hendrix, Roy Lin, Runjia Qian, Ruohan Zhang, Russell Mendonca, Rutav Shah, Ryan Hoque, Ryan Julian, Samuel Bustamante, Sean Kirmani, Sergey Levine, Shan Lin, Sherry Moore, Shikhar Bahl, Shivin Dass, Shubham Sonawani, Shubham Tulsiani, Shuran Song, Sichun Xu, Siddhant Haldar, Siddharth Karamcheti, Simeon Adebola, Simon Guist, Soroush Nasiriany, Stefan Schaal, Stefan Welker, Stephen Tian, Subramanian Ramamoorthy, Sudeep Dasari, Suneel Belkhale, Sungjae Park, Suraj Nair, Suvir Mirchandani, Takayuki Osa, Tanmay Gupta, Tatsuya Harada, Tatsuya Matsushima, Ted Xiao, Thomas Kollar, Tianhe Yu, Tianli Ding, Todor Davchev, Tony Z. Zhao, Travis Armstrong, Trevor Darrell, Trinity Chung, Vidhi Jain, Vikash Kumar, Vincent Vanhoucke, Vitor Guizilini, Wei Zhan, Wenxuan Zhou, Wolfram Burgard, Xi Chen, Xiangyu Chen, Xiaolong Wang, Xinghao Zhu, Xinyang Geng, Xiyuan Liu, Xu Liangwei, Xuanlin Li, Yansong Pang, Yao Lu, Yecheng Jason Ma, Yejin Kim, Yevgen Chebotar, Yifan Zhou, Yifeng Zhu, Yilin Wu, Ying Xu, Yixuan Wang, Yonatan Bisk, Yongqiang Dou, Yoonyoung Cho, Youngwoon Lee, Yuchen Cui, Yue Cao, Yueh-Hua Wu, Yujin Tang, Yuke Zhu, Yunchu Zhang, Yunfan Jiang, Yunshuang Li, Yunzhu Li, Yusuke Iwasawa, Yutaka Matsuo, Zehan Ma, Zhuo Xu, Zichen Jeff Cui, Zichen Zhang, Zipeng Fu, and Zipeng Lin. Open X-Embodiment: Robotic learning datasets and RT-X models. https://arxiv.org/abs/2310.08864, 2023.

[20] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Antonino Furnari, Jian Ma, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. Rescaling egocentric vision: Collection, pipeline and challenges for epic-kitchens-100. *International Journal of Computer Vision (IJCV)*, 130:33–55, 2022. URL https://doi.org/10.1007/s11263-021-01531-2.

[21] Haoran Ding, Anqing Duan, Zezhou Sun, Dezhen Song, and Yoshihiko Nakamura. Imagination at inference: Synthesizing in-hand views for robust visuomotor policy inference, 2025. URL https://arxiv.org/abs/2509.15717.

[22] Runyu Ding, Yuzhe Qin, Jiyue Zhu, Chengzhe Jia, Shiqi Yang, Ruihan Yang, Xiaojuan Qi, and Xiaolong Wang. Bunny-visionpro: Real-time bimanual dexterous teleoperation for imitation learning. 2024.

[23] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2021. URL https://arxiv.org/abs/2010.11929.

[24] Anca D. Dragan and Siddhartha S. Srinivasa. Online customization of teleoperation interfaces. In *2012 IEEE RO-MAN: The 21st IEEE International Symposium on Robot and Human Interactive Communication*, pages 919–924, 2012. doi: 10.1109/ROMAN.2012.6343868.

[25] Jiafei Duan, Yi Ru Wang, Mohit Shridhar, Dieter Fox, and Ranjay Krishna. Ar2-d2:training a robot without a robot, 2023. URL https://arxiv.org/abs/2306.13818.

[26] Bardienus P. Duisterhof, Zhao Mandi, Yunchao Yao, Jia-Wei Liu, Jenny Seidenschwarz, Mike Zheng Shou, Deva Ramanan, Shuran Song, Stan Birchfield, Bowen Wen, and Jeffrey Ichnowski. Deformgs: Scene flow in highly deformable scenes for deformable object manipulation, 2024. URL https://arxiv.org/abs/2312.00583.

[27] Zicong Fan, Maria Parelli, Maria Eleni Kadoglou, Muhammed Kocabas, Xu Chen, Michael J Black, and Otmar Hilliges. HOLD: Category-agnostic 3d reconstruction of interacting hands and objects from video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 494–504, 2024.

[28] Hao-Shu Fang, Hongjie Fang, Zhenyu Tang, Jirong Liu, Chenxi Wang, Junbo Wang, Haoyi Zhu, and Cewu Lu. Rh20t: A comprehensive robotic dataset for learning diverse skills in one-shot. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 653–660. IEEE, 2024.

[29] Zipeng Fu, Tony Z. Zhao, and Chelsea Finn. Mobile aloha: Learning bimanual mobile manipulation with low-cost whole-body teleoperation, 2024. URL https://arxiv.org/abs/2401.02117.

[30] Caelan Garrett, Ajay Mandlekar, Bowen Wen, and Dieter Fox. Skillmimicgen: Automated demonstration generation for efficient skill learning and deployment, 2024. URL https://arxiv.org/abs/2410.18907.

[31] Ankit Goyal, Valts Blukis, Jie Xu, Yijie Guo, Yu-Wei Chao, and Dieter Fox. Rvt2: Learning precise manipulation from few demonstrations. *RSS*, 2024.

[32] Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, Miguel Martin, Tushar Nagarajan, Ilija Radosavovic, Santhosh Kumar Ramakrishnan, Fiona Ryan, Jayant Sharma, Michael Wray, Mengmeng Xu, Eric Zhongcong Xu, Chen Zhao, Siddhant Bansal, Dhruv Batra, Vincent Cartillier, Sean Crane, Tien Do, Morrie Doulaty, Akshay Erapalli, Christoph Feichtenhofer, Adriano Fragomeni, Qichen Fu, Abrham Gebreselasie, Cristina Gonzalez, James Hillis, Xuhua Huang, Yifei Huang, Wenqi Jia, Weslie Khoo, Jachym Kolar, Satwik Kottur, Anurag Kumar, Federico Landini, Chao Li, Yanghao Li, Zhenqiang Li, Karttikeya Mangalam, Raghava Modhugu, Jonathan Munro, Tullie Murrell, Takumi Nishiyasu, Will Price, Paola Ruiz Puentes, Merey Ramazanova, Leda Sari, Kiran Somasundaram, Audrey Southerland, Yusuke Sugano, Ruijie Tao, Minh Vo, Yuchen Wang, Xindi Wu, Takuma Yagi, Ziwei Zhao, Yunyi Zhu, Pablo Arbelaez, David Crandall, Dima Damen, Giovanni Maria Farinella, Christian Fuegen, Bernard Ghanem, Vamsi Krishna Ithapu, C. V. Jawahar, Hanbyul Joo, Kris Kitani, Haizhou Li, Richard Newcombe, Aude Oliva, Hyun Soo Park, James M. Rehg, Yoichi Sato, Jianbo Shi, Mike Zheng Shou, Antonio Torralba, Lorenzo Torresani, Mingfei Yan, and Jitendra Malik. Ego4d: Around the world in 3,000 hours of egocentric video, 2022. URL https://arxiv.org/abs/2110.07058.

[33] Siddhant Haldar and Lerrel Pinto. Point policy: Unifying observations and actions with key points for robot manipulation. *arXiv preprint arXiv:2502.20391*, 2025.

[34] Yana Hasson, Gül Varol, Dimitrios Tzionas, Igor Kalevatykh, Michael J. Black, Ivan Laptev, and Cordelia Schmid. Learning joint reconstruction of hands and manipulated objects, 2019. URL https://arxiv.org/abs/1904.05767.

[35] Yana Hasson, Gül Varol, Ivan Laptev, and Cordelia Schmid. Towards unconstrained joint hand-object reconstruction from rgb videos, 2022. URL https://arxiv.org/abs/2108.07044.

[36] Liang Heng, Xiaoqi Li, Shangqing Mao, Jiaming Liu, Ruolin Liu, Jingli Wei, Yu-Kai Wang, Yueru Jia, Chenyang Gu, Rui Zhao, Shanghang Zhang, and Hao Dong. Rwor: Generating robot demonstrations from human hand collection for policy learning without robot, 2025. URL https://arxiv.org/abs/2507.03930.

[37] Nick Heppert, Max Argus, Tim Welschehold, Thomas Brox, and Abhinav Valada. Ditto: Demonstration imitation by trajectory transformation. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, page 7565–7572. IEEE, October 2024. doi: 10.1109/iros58592.2024.10801982. URL http://dx.doi.org/10.1109/IROS58592.2024.10801982.

[38] Nick Heppert, Minh Quang Nguyen, and Abhinav Valada. Real2gen: Imitation learning from a single human demonstration with generative foundational models. In *ICRA 2025 Workshop on Foundation Models and Neuro-Symbolic AI for Robotics*, 2025. URL

https://openreview.net/forum?id=TYtYTHTlel.

[39] Ahmed Hussein, Mohamed Medhat Gaber, Eyad Elyan, and Chrisina Jayne. Imitation learning: A survey of learning methods. *ACM Comput. Surv.*, 50(2), April 2017. ISSN 0360-0300. doi: 10.1145/3054912. URL https://doi.org/10.1145/3054912.

[40] Aadhithya Iyer, Zhuoran Peng, Yinlong Dai, Irmak Guzey, Siddhant Haldar, Soumith Chintala, and Lerrel Pinto. Open teach: A versatile teleoperation system for robotic manipulation, 2024. URL https://arxiv.org/abs/2403.07870.

[41] Eric Jang, Alex Irpan, Mohi Khansari, Daniel Kappler, Frederik Ebert, Corey Lynch, Sergey Levine, and Chelsea Finn. BC-z: Zero-shot task generalization with robotic imitation learning. In *5th Annual Conference on Robot Learning*, 2021. URL https://openreview.net/forum?id=8kbp23tSGYv.

[42] Yueru Jia, Jiaming Liu, Sixiang Chen, Chenyang Gu, Zhilue Wang, Longzan Luo, Lily Lee, Pengwei Wang, Zhongyuan Wang, Renrui Zhang, and Shanghang Zhang. Lift3d foundation policy: Lifting 2d large-scale pretrained models for robust 3d robotic manipulation, 2024. URL https://arxiv.org/abs/2411.18623.

[43] Simar Kareer, Dhruv Patel, Ryan Punamiya, Pranay Mathur, Shuo Cheng, Chen Wang, Judy Hoffman, and Danfei Xu. Egomimic: Scaling imitation learning via egocentric video, 2024. URL https://arxiv.org/abs/2410.24221.

[44] Tsung-Wei Ke, Nikolaos Gkanatsios, and Katerina Fragkiadaki. 3d diffuser actor: Policy diffusion with 3d scene representations. *Arxiv*, 2024.

[45] David Kent, Carl Saldanha, and Sonia Chernova. A comparison of remote robot teleoperation interfaces for general object manipulation. In *2017 12th ACM/IEEE International Conference on Human-Robot Interaction (HRI*, pages 371–379, 2017.

[46] Justin Kerr, Chung Min Kim, Mingxuan Wu, Brent Yi, Qianqian Wang, Ken Goldberg, and Angjoo Kanazawa. Robot see robot do: Imitating articulated object manipulation with monocular 4d reconstruction. In *8th Annual Conference on Robot Learning*, 2024. URL https://openreview.net/forum?id=2LLu3gavF1.

[47] Chung Hee Kim, Abhisesh Silwal, and George Kantor. Autonomous robotic pepper harvesting: Imitation learning in unstructured agricultural environments, 2024. URL https://arxiv.org/abs/2411.09929.

[48] Moo Jin Kim, Jiajun Wu, and Chelsea Finn. Giving robots a hand: Learning generalizable manipulation with eye-in-hand human video demonstrations, 2023. URL https://arxiv.org/abs/2307.05959.

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

[50] Yann Labbé, Lucas Manuelli, Arsalan Mousavian, Stephen Tyree, Stan Birchfield, Jonathan Tremblay, Justin Carpentier, Mathieu Aubry, Dieter Fox, and Josef Sivic. Megapose: 6d pose estimation of novel objects via render & compare. In *Proceedings of the 6th Conference on Robot Learning (CoRL)*, 2022.

[51] Samuli Laine, Janne Hellsten, Tero Karras, Yeongho Seol, Jaakko Lehtinen, and Timo Aila. Modular primitives for high-performance differentiable rendering. *ACM Transactions on Graphics*, 39(6), 2020.

[52] Michael Laskey, Caleb Chuck, Jonathan Lee, Jeffrey Mahler, Sanjay Krishnan, Kevin Jamieson, Anca Dragan, and Ken Goldberg. Comparing human-centric and robot-centric sampling for robot deep learning from demonstrations, 2017. URL https://arxiv.org/abs/1610.00850.

[53] Adam Leeper, Kaijen Hsiao, Matei Ciocarlie, Leila Takayama, and David Gossow. Strategies for human-in-the-loop robotic grasping. In *2012 7th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 1–8, 2012. doi: 10.1145/2157689.2157691.

[54] Marion Lepert, Ria Doshi, and Jeannette Bohg. Shadow: Leveraging segmentation masks for cross-embodiment policy transfer, 2025. URL https://arxiv.org/abs/2503.00774.

[55] Marion Lepert, Jiaying Fang, and Jeannette Bohg. Masquerade: Learning from in-the-wild human videos using data-editing. *arXiv preprint arXiv:2508.09976*, 2025.

[56] Marion Lepert, Jiaying Fang, and Jeannette Bohg. Phantom: Training robots without robots using only human videos, 2025. URL https://arxiv.org/abs/2503.00779.

[57] Mara Levy, Siddhant Haldar, Lerrel Pinto, and Abhinav Shirivastava. P3-po: Prescriptive point priors for visuo-spatial generalization of robot policies, 2024. URL https://arxiv.org/abs/2412.06784.

[58] Jinhan Li, Yifeng Zhu, Yuqi Xie, Zhenyu Jiang, Mingyo Seo, Georgios Pavlakos, and Yuke Zhu. Okami: Teaching humanoid robots manipulation skills through single video imitation. In *8th Annual Conference on Robot Learning (CoRL)*, 2024.

[59] Dixuan Lin, Tianyou Wang, Zhuoyang Pan, Yufu Wang, Lingjie Liu, and Kostas Daniilidis. Zero-shot reconstruction of in-scene object manipulation from video, 2025. URL https://arxiv.org/abs/2512.19684.

[60] Philipp Lindenberger, Paul-Edouard Sarlin, and Marc Pollefeys. LightGlue: Local Feature Matching at Light Speed. In *ICCV*, 2023.

[61] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. *arXiv preprint arXiv:2303.05499*, 2023.

[62] Vincent Liu, Ademi Adeniji, Haotian Zhan, Siddhant Haldar, Raunaq Bhirangi, Pieter Abbeel, and Lerrel

Pinto. Egozero: Robot learning from smart glasses, 2025. URL https://arxiv.org/abs/2505.20290.

[63] Yecheng Jason Ma, William Liang, Vaidehi Som, Vikash Kumar, Amy Zhang, Osbert Bastani, and Dinesh Jayaraman. Liv: Language-image representations and rewards for robotic control, 2023. URL https://arxiv.org/abs/2306.00958.

[64] Yecheng Jason Ma, Shagun Sodhani, Dinesh Jayaraman, Osbert Bastani, Vikash Kumar, and Amy Zhang. Vip: Towards universal visual reward and representation via value-implicit pre-training, 2023. URL https://arxiv.org/abs/2210.00030.

[65] Arjun Majumdar, Karmesh Yadav, Sergio Arnaud, Yecheng Jason Ma, Claire Chen, Sneha Silwal, Aryan Jain, Vincent-Pierre Berges, Pieter Abbeel, Jitendra Malik, Dhruv Batra, Yixin Lin, Oleksandr Maksymets, Aravind Rajeswaran, and Franziska Meier. Where are we in the search for an artificial visual cortex for embodied intelligence?, 2024. URL https://arxiv.org/abs/2303.18240.

[66] 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. In *arXiv preprint arXiv:2108.03298*, 2021.

[67] 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. URL https://arxiv.org/abs/2310.17596.

[68] Suraj Nair, Aravind Rajeswaran, Vikash Kumar, Chelsea Finn, and Abhinav Gupta. R3m: A universal visual representation for robot manipulation, 2022. URL https://arxiv.org/abs/2203.12601.

[69] Maxime Oquab, Timothée Darcet, Theo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Russell Howes, Po-Yao Huang, Hu Xu, Vasu Sharma, Shang-Wen Li, Wojciech Galuba, Mike Rabbat, Mido Assran, Nicolas Ballas, Gabriel Synnaeve, Ishan Misra, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision, 2023.

[70] Chuer Pan, Litian Liang, Dominik Bauer, Eric Cousineau, Benjamin Burchfiel, Siyuan Feng, and Shuran Song. One demo is worth a thousand trajectories: Action-view augmentation for visuomotor policies. In *Conference on Robot Learning (CoRL)*, 2025.

[71] Georgios Papagiannis, Norman Di Palo, Pietro Vitiello, and Edward Johns. R+x: Retrieval and execution from everyday human videos, 2025. URL https://arxiv.org/abs/2407.12957.

[72] Sungjae Park, Homanga Bharadhwaj, and Shubham Tulsiani. Demodiffusion: One-shot human imitation using pre-trained diffusion policy, 2025. URL https://arxiv.org/abs/2506.20668.

[73] Austin Patel, Andrew Wang, Ilija Radosavovic, and Jitendra Malik. Learning to imitate object interactions from internet videos, 2022. URL https://arxiv.org/abs/2211.13225.

[74] Georgios Pavlakos, Dandan Shan, Ilija Radosavovic, Angjoo Kanazawa, David Fouhey, and Jitendra Malik. Reconstructing hands in 3D with transformers. In *CVPR*, 2024.

[75] Karl Pertsch, Kyle Stachowicz, Brian Ichter, Danny Driess, Suraj Nair, Quan Vuong, Oier Mees, Chelsea Finn, and Sergey Levine. Fast: Efficient action tokenization for vision-language-action models, 2025. URL https://arxiv.org/abs/2501.09747.

[76] Zezhong Qian, Xiaowei Chi, Yuming Li, Shizun Wang, Zhiyuan Qin, Xiaozhu Ju, Sirui Han, and Shanghang Zhang. Wristworld: Generating wrist-views via 4d world models for robotic manipulation, 2025. URL https://arxiv.org/abs/2510.07313.

[77] Yuzhe Qin, Wei Yang, Binghao Huang, Karl Van Wyk, Hao Su, Xiaolong Wang, Yu-Wei Chao, and Dieter Fox. Anyteleop: A general vision-based dexterous robot arm-hand teleoperation system, 2024. URL https://arxiv.org/abs/2307.04577.

[78] Mohammad Nomaan Qureshi, Sparsh Garg, Francisco Yandun, David Held, George Kantor, and Abhisesh Silwal. Splatsim: Zero-shot sim2real transfer of rgb manipulation policies using gaussian splatting, 2024. URL https://arxiv.org/abs/2409.10161.

[79] 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. *arXiv preprint arXiv:2408.00714*, 2024. URL https://arxiv.org/abs/2408.00714.

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

[81] Javier Romero, Dimitrios Tzionas, and Michael J. Black. Embodied hands: Modeling and capturing hands and bodies together. *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)*, 36(6), November 2017.

[82] Kyle Sargent, Zizhang Li, Tanmay Shah, Charles Herrmann, Hong-Xing Yu, Yunzhi Zhang, Eric Ryan Chan, Dmitry Lagun, Li Fei-Fei, Deqing Sun, and Jiajun Wu. Zeronvs: Zero-shot 360-degree view synthesis from a single image, 2024. URL https://arxiv.org/abs/2310.17994.

[83] Paul-Edouard Sarlin, Cesar Cadena, Roland Siegwart, and Marcin Dymczyk. From coarse to fine: Robust hierarchical localization at large scale. In *CVPR*, 2019.

[84] Stefan Schaal. Learning from demonstration. In

Proceedings of the 10th International Conference on Neural Information Processing Systems, NIPS'96, page 1040–1046, Cambridge, MA, USA, 1996. MIT Press.

[85] Nur Muhammad Mahi Shafiullah, Anant Rai, Haritheja Etukuru, Yiqian Liu, Ishan Misra, Soumith Chintala, and Lerrel Pinto. On bringing robots home, 2023. URL https://arxiv.org/abs/2311.16098.

[86] Kenneth Shaw, Shikhar Bahl, and Deepak Pathak. Videodex: Learning dexterity from internet videos, 2022. URL https://arxiv.org/abs/2212.04498.

[87] Junyao Shi, Zhuolun Zhao, Tianyou Wang, Ian Pedroza, Amy Luo, Jie Wang, Jason Ma, and Dinesh Jayaraman. Zeromimic: Distilling robotic manipulation skills from web videos. *ICRA*, 2025. URL https://zeromimic.github.io/.

[88] Himanshu Gaurav Singh, Antonio Loquercio, Carmelo Sferrazza, Jane Wu, Haozhi Qi, Pieter Abbeel, and Jitendra Malik. Hand-object interaction pretraining from videos. In *International Conference on Robotics and Automation (ICRA)*. IEEE, 2025.

[89] Matthew Tancik, Ethan Weber, Evonne Ng, Ruilong Li, Brent Yi, Justin Kerr, Terrance Wang, Alexander Kristoffersen, Jake Austin, Kamyar Salahi, Abhik Ahuja, David McAllister, and Angjoo Kanazawa. Nerfstudio: A modular framework for neural radiance field development. In *ACM SIGGRAPH 2023 Conference Proceedings*, SIGGRAPH '23, 2023.

[90] Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, Jianlan Luo, You Liang Tan, Lawrence Yunliang Chen, Pannag Sanketi, Quan Vuong, Ted Xiao, Dorsa Sadigh, Chelsea Finn, and Sergey Levine. Octo: An open-source generalist robot policy, 2024. URL https://arxiv.org/abs/2405.12213.

[91] SAM 3D Team, Xingyu Chen, Fu-Jen Chu, Pierre Gleize, Kevin J Liang, Alexander Sax, Hao Tang, Weiyao Wang, Michelle Guo, Thibaut Hardin, Xiang Li, Aohan Lin, Jiawei Liu, Ziqi Ma, Anushka Sagar, Bowen Song, Xiaodong Wang, Jianing Yang, Bowen Zhang, Piotr Dollár, Georgia Gkioxari, Matt Feiszli, and Jitendra Malik. Sam 3d: 3dfy anything in images. 2025. URL https://arxiv.org/abs/2511.16624.

[92] Stephen Tian, Blake Wulfe, Kyle Sargent, Katherine Liu, Sergey Zakharov, Vitor Guizilini, and Jiajun Wu. View-invariant policy learning via zero-shot novel view synthesis, 2025. URL https://arxiv.org/abs/2409.03685.

[93] Mel Vecerik, Carl Doersch, Yi Yang, Todor Davchev, Yusuf Aytar, Guangyao Zhou, Raia Hadsell, Lourdes Agapito, and Jon Scholz. Robotap: Tracking arbitrary points for few-shot visual imitation, 2023. URL https://arxiv.org/abs/2308.15975.

[94] Chen Wang, Haochen Shi, Weizhuo Wang, Ruohan Zhang, Li Fei-Fei, and C. Karen Liu. Dexcap: Scalable and portable mocap data collection system for dexterous manipulation, 2024. URL https://arxiv.org/abs/2403.07788.

[95] Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer, 2025. URL https://arxiv.org/abs/2503.11651.

[96] Bowen Wen, Wei Yang, Jan Kautz, and Stan Birchfield. Foundationpose: Unified 6d pose estimation and tracking of novel objects, 2024. URL https://arxiv.org/abs/2312.08344.

[97] Guanjun Wu, Taoran Yi, Jiemin Fang, Lingxi Xie, Xiaopeng Zhang, Wei Wei, Wenyu Liu, Qi Tian, and Xinggang Wang. 4d gaussian splatting for real-time dynamic scene rendering, 2024. URL https://arxiv.org/abs/2310.08528.

[98] Philipp Wu, Yide Shentu, Zhongke Yi, Xingyu Lin, and Pieter Abbeel. Gello: A general, low-cost, and intuitive teleoperation framework for robot manipulators, 2024. URL https://arxiv.org/abs/2309.13037.

[99] Qi Wu, Janick Martinez Esturo, Ashkan Mirzaei, Nicolas Moenne-Loccoz, and Zan Gojcic. 3dgut: Enabling distorted cameras and secondary rays in gaussian splatting. *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

[100] Yuxi Xiao, Jianyuan Wang, Nan Xue, Nikita Karaev, Iurii Makarov, Bingyi Kang, Xin Zhu, Hujun Bao, Yujun Shen, and Xiaowei Zhou. Spatialtrackerv2: 3d point tracking made easy. In *ICCV*, 2025.

[101] Haoyu Xiong, Haoyuan Fu, Jieyi Zhang, Chen Bao, Qiang Zhang, Yongxi Huang, Wenqiang Xu, Animesh Garg, and Cewu Lu. Robotube: Learning household manipulation from human videos with simulated twin environments. In Karen Liu, Dana Kulic, and Jeff Ichnowski, editors, *Proceedings of The 6th Conference on Robot Learning*, volume 205 of *Proceedings of Machine Learning Research*, pages 1–10. PMLR, 14–18 Dec 2023. URL https://proceedings.mlr.press/v205/xiong23a.html.

[102] Yuan Xu, Jiabing Yang, Xiaofeng Wang, Yixiang Chen, Zheng Zhu, Bowen Fang, Guan Huang, Xinze Chen, Yun Ye, Qiang Zhang, Peiyan Li, Xiangnan Wu, Kai Wang, Bing Zhan, Shuo Lu, Jing Liu, Nianfeng Liu, Yan Huang, and Liang Wang. Egodemogen: Novel egocentric demonstration generation enables viewpoint-robust manipulation, 2025. URL https://arxiv.org/abs/2509.22578.

[103] Jingyun Yang, Junwu Zhang, Connor Settle, Akshara Rai, Rika Antonova, and Jeannette Bohg. Learning periodic tasks from human demonstrations, 2022. URL https://arxiv.org/abs/2109.14078.

[104] Sizhe Yang, Wenye Yu, Jia Zeng, Jun Lv, Kerui Ren, Cewu Lu, Dahua Lin, and Jiangmiao Pang. Novel demonstration generation with gaussian splatting enables robust one-shot manipulation, 2025. URL https://arxiv.org/abs/2504.13175.

[105] Ziyi Yang, Xinyu Gao, Wen Zhou, Shaohui Jiao, Yuqing Zhang, and Xiaogang Jin. Deformable 3d gaussians for

high-fidelity monocular dynamic scene reconstruction, 2023. URL https://arxiv.org/abs/2309.13101.

[106] Vickie Ye, Ruilong Li, Justin Kerr, Matias Turkulainen, Brent Yi, Zhuoyang Pan, Otto Seiskari, Jianbo Ye, Jeffrey Hu, Matthew Tancik, and Angjoo Kanazawa. gsplat: An open-source library for gaussian splatting. *Journal of Machine Learning Research*, 26(34):1–17, 2025.

[107] Zhao-Heng Yin, Sherry Yang, and Pieter Abbeel. Object-centric 3d motion field for robot learning from human videos, 2025. URL https://arxiv.org/abs/2506.04227.

[108] Takahiro Yonemaru, Weiwei Wan, Tatsuki Nishimura, and Kensuke Harada. Learning to push, group, and grasp: A diffusion policy approach for multi-object delivery, 2025. URL https://arxiv.org/abs/2502.08452.

[109] 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. URL https://arxiv.org/abs/2505.09601.

[110] Justin Yu, Yide Shentu, Di Wu, Pieter Abbeel, Ken Goldberg, and Philipp Wu. Egomi: Learning active vision and whole-body manipulation from egocentric human demonstrations, 2025. URL https://arxiv.org/abs/2511.00153.

[111] Tao Yu, Runseng Feng, Ruoyu Feng, Jinming Liu, Xin Jin, Wenjun Zeng, and Zhibo Chen. Inpaint anything: Segment anything meets image inpainting. *arXiv preprint arXiv:2304.06790*, 2023.

[112] Tianhe Yu, Ted Xiao, Austin Stone, Jonathan Tompson, Anthony Brohan, Su Wang, Jaspiar Singh, Clayton Tan, Dee M, Jodilyn Peralta, Brian Ichter, Karol Hausman, and Fei Xia. Scaling robot learning with semantically imagined experience, 2023. URL https://arxiv.org/abs/2302.11550.

[113] Chengbo Yuan, Rui Zhou, Mengzhen Liu, Yingdong Hu, Shengjie Wang, Li Yi, Chuan Wen, Shanghang Zhang, and Yang Gao. Motiontrans: Human vr data enable motion-level learning for robotic manipulation policies. *arXiv preprint arXiv:2509.17759*, 2025.

[114] Yanjie Ze, Gu Zhang, Kangning Zhang, Chenyuan Hu, Muhan Wang, and Huazhe Xu. 3d diffusion policy: Generalizable visuomotor policy learning via simple 3d representations. In *Proceedings of Robotics: Science and Systems (RSS)*, 2024.

[115] Qiyuan Zeng, Chengmeng Li, Jude St. John, Zhongyi Zhou, Junjie Wen, Guorui Feng, Yichen Zhu, and Yi Xu. Activeumi: Robotic manipulation with active perception from robot-free human demonstrations, 2025. URL https://arxiv.org/abs/2510.01607.

[116] Jason Y. Zhang, Sam Pepose, Hanbyul Joo, Deva Ramanan, Jitendra Malik, and Angjoo Kanazawa. Perceiving 3d human-object spatial arrangements from a single image in the wild, 2020. URL https://arxiv.org/abs/2007.15649.

[117] 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, 2018. URL https://arxiv.org/abs/1710.04615.

[118] Xiaoyu Zhang, Matthew Chang, Pranav Kumar, and Saurabh Gupta. Diffusion meets dagger: Supercharging eye-in-hand imitation learning, 2024. URL https://arxiv.org/abs/2402.17768.

[119] Tony Z. Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware, 2023. URL https://arxiv.org/abs/2304.13705.

[120] Zhaxizhuoma, Kehui Liu, Chuyue Guan, Zhongjie Jia, Ziniu Wu, Xin Liu, Tianyu Wang, Shuai Liang, Pengan Chen, Pingrui Zhang, Haoming Song, Delin Qu, Dong Wang, Zhigang Wang, Nieqing Cao, Yan Ding, Bin Zhao, and Xuelong Li. Fastumi: A scalable and hardware-independent universal manipulation interface with dataset. 2025. URL https://arxiv.org/abs/2409.19499.

[121] Eric Zhu, Mara Levy, Matthew Gwilliam, and Abhinav Shrivastava. Nerf-aug: Data augmentation for robotics with neural radiance fields, 2025. URL https://arxiv.org/abs/2411.02482.

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

[123] Zhifan Zhu, Siddhant Bansal, Shashank Tripathi, and Dima Damen. Reconstructing objects along hand interaction timelines in egocentric video, 2025. URL https://arxiv.org/abs/2512.07394.

[124] Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, Quan Vuong, Vincent Vanhoucke, Huong Tran, Radu Soricut, Anikait Singh, Jaspiar Singh, Pierre Sermanet, Pannag R. Sanketi, Grecia Salazar, Michael S. Ryoo, Krista Reymann, Kanishka Rao, Karl Pertsch, Igor Mordatch, Henryk Michalewski, Yao Lu, Sergey Levine, Lisa Lee, Tsang-Wei Edward Lee, Isabel Leal, Yuheng Kuang, Dmitry Kalashnikov, Ryan Julian, Nikhil J. Joshi, Alex Irpan, Brian Ichter, Jasmine Hsu, Alexander Herzog, Karol Hausman, Keerthana Gopalakrishnan, Chuyuan Fu, Pete Florence, Chelsea Finn, Kumar Avinava Dubey, Danny Driess, Tianli Ding, Krzysztof Marcin Choromanski, Xi Chen, Yevgen Chebotar, Justice Carbajal, Noah Brown, Anthony Brohan, Montserrat Gonzalez Arenas, and Kehang Han. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In Jie Tan, Marc Toussaint, and Kourosh Darvish, editors, *Proceedings of The 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pages 2165–2183. PMLR, 06–09 Nov 2023. URL

https://proceedings.mlr.press/v229/zitkovich23a.html.

APPENDIX

A. Data Collection

1) Fiducial Marker: Similar to Universal Manipulation Interface (UMI) [17], a fiducial marker of known size is optionally placed in the scene during the initial scan. If present, the marker is used to aid the scene-level scale alignment discussed in Sec. III-C1, and is further detailed in Appendix B. Note that the fiducial marker does not contribute to the Structure-from-Motion (SfM) and does not appear in demonstration videos.

2) Camera Calibration: The head-mounted egocentric camera used to record the demonstrations is modeled as a pinhole camera and may optionally be calibrated prior to data collection. If calibrated intrinsics are available, they are used directly during localization. Otherwise, the intrinsics are first approximated and then refined during localization by additionally registering each demonstration frame into the existing SfM model and performing bundle adjustment.

B. Scene-Level Scale Alignment

Because SfM reconstructions are ambiguous up to a global scale, the reconstruction from Sec. III-B needs to be aligned with the predicted monocular depth maps from Sec. III-C1. We estimate the affine mapping between the predicted depth z^{pred} and SfM depth z^{sfm}

$$z^{\text{sfm}} \approx A + B z^{\text{pred}} \tag{16}$$

and use the resulting scale and offset to align the scene reconstruction and Gaussian Splat with the depth maps.

For each demonstration frame, we use the 2D-3D SfM correspondences from Hierarchical Location [83]. Each correspondence gives a 2D keypoint in the image and a matched 3D SfM point, whose camera-frame depth is z_i^{sfm}. The n correspondences whose 2D keypoints are closest to but lie outside the object mask are selected and used to sample the corresponding monocular depth values z_i^{pred}, where n varies by task but is typically between 10-50. A and B are then estimated by solving the nonlinear least squares equation

$$\min_{A,B} \frac{1}{2} \sum_i \rho\left(\left(A + B\, z_i^{\text{pred}} - z_i^{\text{sfm}} \right)^2 \right) \tag{17}$$

where ρ is the Huber loss function. If the fiducial marker from Appendix A2 is present, the reconstruction is further rescaled so that the recovered marker geometry matches its known size. This is done by triangulating detected tag corner points across views and scaling the reconstruction such that the reconstructed tag length aligns with its known dimensions.

C. Hand Pose Initialization

The 3D hand pose estimates (Sec. III-C2) produced by HAMER [74] are often noisy and temporally inconsistent, as HAMER does not use temporal information between frames. To refine and temporally smooth the hand poses, we utilize a two-stage optimization approach. In the first stage, we fix the hand pose parameters $\boldsymbol{\theta}$ and optimize

$$\begin{aligned} \boldsymbol{\Theta}^{\text{coarse}} &= \{\mathbf{R}^{hand}, \mathbf{t}^{hand}, \boldsymbol{\beta}\} \\ \min_{\boldsymbol{\Theta}^{\text{coarse}}} & \lambda_{\text{2D}} \mathcal{L}_{\text{2D}} + \lambda_{\text{smooth}} \mathcal{L}_{\text{smooth}} \end{aligned} \tag{18}$$

where

$$\begin{aligned} V^{hand} &= \mathbf{R}^{hand} \cdot \text{MANO}(\boldsymbol{\theta}, \boldsymbol{\beta}) + \mathbf{t}^{hand} \\ \mathcal{L}_{\text{2D}} &= \| \Pi(V^{hand}) - \mathbf{v}_{2d}^{hand} \| \\ \mathcal{L}_{\text{smooth}} &= \sum_t \sum_{\mathbf{v}^h \in V^{hand}} \| \mathbf{v}_t^h - \mathbf{v}_{t-1}^h \|_2^2 \end{aligned} \tag{19}$$

Here, $\Pi(\cdot)$ denotes the camera projection operator, and $\mathbf{v}_{2D}^{hand}$ are the 2D hand keypoints predicted by HAMER.

In the second stage, we additionally optimize the hand pose parameters $\boldsymbol{\theta}$ and add a depth supervision loss to align the hand vertices with the predicted monocular depth maps

$$\begin{aligned} \boldsymbol{\Theta}^{\text{fine}} &= \{\mathbf{R}^{hand}, \mathbf{t}^{hand}, \boldsymbol{\theta}, \boldsymbol{\beta}\} \\ \min_{\boldsymbol{\Theta}^{\text{fine}}} & \lambda_{\text{2D}} \mathcal{L}_{\text{2D}} + \lambda_{\text{smooth}} \mathcal{L}_{\text{smooth}} + \lambda_{\mathcal{D}_{hand}} \mathcal{L}_{\mathcal{D}_{hand}} \end{aligned} \tag{20}$$

where $\mathcal{L}_{\text{2D}}$ and $\mathcal{L}_{\text{smooth}}$ are the same as Eq. 19, and $\mathcal{L}_{\mathcal{D}_{hand}}$ is the same hand depth loss defined in Eq. 9. We found this two-stage approach to be more stable than initially optimizing all hand pose parameters with depth supervision.

D. Object Pose Initialization

As discussed in Sec. III-C3, MegaPose [50] provides an initial 6D pose estimate $(\mathbf{R}_0^{obj*}, \mathbf{t}_0^{obj*})$. This estimate is obtained using the canonical object mesh reconstructed by SAM3D [91], with vertices V^{obj*} which are defined up to an arbitrary scale. To refine the object mesh, an initial contact frame $\tilde{t}_s$ is first estimated by thresholding the overlap between the hand and object segmentation masks. Then, we optimize

$$\min_{\mathbf{R}_0^{obj}, \mathbf{t}_0^{obj}, s^{obj}} \lambda_{\mathcal{M}_{obj}} \mathcal{L}_{\mathcal{M}_{obj}} + \lambda_{\mathcal{D}_{obj}} \mathcal{L}_{\mathcal{D}_{obj}} \tag{21}$$

where $\mathbf{R}_0^{obj}$ and $\mathbf{t}_0^{obj}$ are initialized as $\mathbf{R}_0^{obj*}$ and $\mathbf{t}_0^{obj*}$ respectively, and s^{obj} is initialized to 1. The losses $\mathcal{L}_{\mathcal{M}_{obj}}$ and $\mathcal{L}_{\mathcal{D}_{obj}}$ follow the same definitions as Eq. 8 and Eq. 9, but are evaluated using the scaled object mesh vertices $V^{obj} = s^{obj} V^{obj*}$ for all frames $t \leq \tilde{t}_s$. During this interval, the object is assumed to be stationary. The estimate $\tilde{t}_s$ is only used for initial object scale and pose refinement. The final contact start and end frames (t_s, t_e) are computed using motion changes as described in Appendix E1.

E. Hand-Object Optimization

1) Contact Start and End Estimation: Following Sec. III-D1, once object poses for all frames have been determined, we estimate the contact start and end frames (t_s, t_e). Inspired by Dixuan *et al* [59], we set a translation threshold ϵ_o and rotation threshold ϵ_r. If at time t_s the change in object translation Δo or rotation Δr exceeds these

thresholds for m consecutive frames, t_s is determined to be the contact start frame.

$$\begin{aligned} t_s^o &= \inf\left\{ t \;:\; \min_{k\in\{t,\dots,t+m-1\}} \|\Delta o_k\|_2 \geq \epsilon_o \right\} \\ t_s^r &= \inf\left\{ t \;:\; \min_{k\in\{t,\dots,t+m-1\}} \|\Delta r_k\|_2 \geq \epsilon_r \right\} \\ t_s &= \min(t_s^o, t_s^r) \end{aligned} \tag{22}$$

Similarly, the end contact frame t_e is determined by identifying the earliest time at which the object translation or rotation falls below the respective thresholds for m consecutive frames.

$$\begin{aligned} t_e^o &= \inf\left\{ t \;:\; \min_{k\in\{t-m+1,\dots,t\}} \|\Delta o_k\|_2 < \epsilon_o \right\} \\ t_e^r &= \inf\left\{ t \;:\; \min_{k\in\{t-m+1,\dots,t\}} \|\Delta r_k\|_2 < \epsilon_r \right\} \\ t_e &= \max(t_e^o, t_e^r) \end{aligned} \tag{23}$$

2) Joint Hand-Object Refinement Auxiliary Losses: In addition to the losses defined in Sec. III-D2, we include auxiliary losses $\mathcal{L}_{\text{aux}}$ presented in Eq. 13 to further guide the hand-object optimization. Here, V^{hand} and V^{obj} denote the hand and object mesh vertices after applying the current pose transformations at each timestep.

Scene TSDF Loss. To prevent collisions with the static scene, a scene-level truncated signed distance field (TSDF) is constructed by rendering depth from the scene Gaussian Splat. A collision loss penalizes hand and object vertices that penetrate the scene surface.

$$\mathcal{L}_{\text{scene}} = \sum_{\mathbf{v}^h \in V^{hand}} \Phi^{scene}(\mathbf{v}^h) + \sum_{\mathbf{v}^o \in V^{obj}} \Phi^{scene}(\mathbf{v}^o), \tag{24}$$

Here, $\Phi^{scene}(\cdot)$ denotes the scene TSDF evaluated at hand vertex $\mathbf{v}^h \in V^{hand}$ and object vertex $\mathbf{v}^o \in V^{obj}$.

Resting-on-Scene Loss. When the hand is not in contact with the object, we encourage the object to remain in contact with the static scene. Using the scene TSDF Φ^{scene}, we extract a scene point cloud P^{scene} and constrain object mesh vertices to remain close to the scene surface during non-contact frames, reducing floating object artifacts.

$$\mathcal{L}_{\text{rest}}(t) = \begin{cases} 0, & t_s \leq t \leq t_e, \\ \displaystyle\sum_{\mathbf{v}^o \in V^{obj}} \min_{\mathbf{p}\in P^{scene}} \|\mathbf{v}_t^o - \mathbf{p}\|_2^2, & \text{otherwise.} \end{cases} \tag{25}$$

Hand Projection Loss. The hand projection loss $\mathcal{L}_{\text{2D}}$ in Eq. 19 constrains the projected hand vertices to remain close to the original 2D keypoints predicted by HAMER.

Hand Pose Regularization Loss A hand pose regularization term is included to limit deviations of hand pose $\boldsymbol{\theta}$ from the original HAMER output $\hat{\boldsymbol{\theta}}$. This prevents unnatural hand shapes and maintains physically plausible hand configurations.

$$\mathcal{L}_{\text{hp}} = \|\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}\| \tag{26}$$

The auxiliary loss is defined as the weighted sum of all auxiliary terms described above.

$$\mathcal{L}_{\text{aux}} = \lambda_{\text{scene}}\mathcal{L}_{\text{scene}} + \lambda_{\text{rest}}\mathcal{L}_{\text{rest}} + \lambda_{\text{2D}}\mathcal{L}_{\text{2D}} + \lambda_{\text{hp}}\mathcal{L}_{\text{hp}} \tag{27}$$

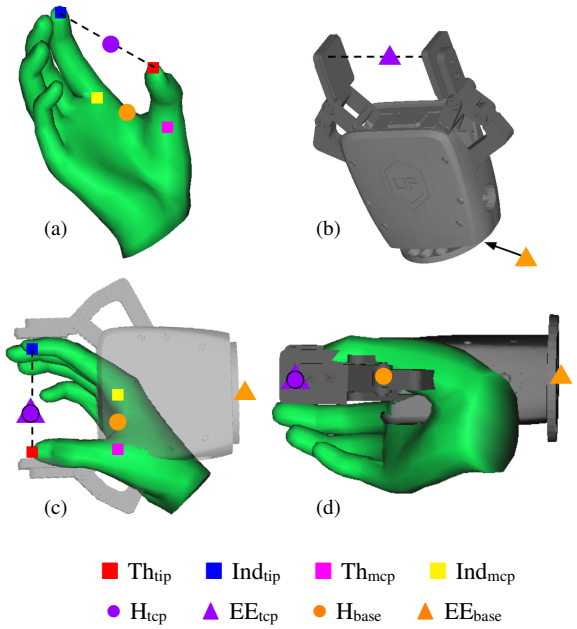

Fig. 10: Hand-to-end-effector pose mapping. (a) Example hand output showing thumb, index, and derived TCP and base keypoints. (b) Gripper model with corresponding TCP and base keypoints. (c) Front view of the gripper aligned to the hand using the mapped TCP and base keypoints. (d) Side view of the gripper alignment using the TCP and base keypoints.

F. Retargeting and Rendering

1) Hand-to-End-Effector Mapping: Fig. 10 shows our hand-to-end-effector mapping which is inspired by Papagianni *et al.* [71]. The tool center point (TCP) of the end-effector EE_{tcp} is mapped to the hand as the midpoint H_{tcp} between the thumb and index fingertips, Th_{tip} and Ind_{tip}. The vector connecting the thumb and index fingertips defines a principal axis used to orient the gripper such that the gripper base EE_{base} is aligned as closely as possible with the midpoint H_{base} between the thumb and index MCP joints, Th_{mcp} and Ind_{mcp}.

2) Pre-Contact Trajectory Optimization: As discussed in Sec. III-E, pre-contact trajectory optimization is applied to prevent unintended gripper–object collisions. Inspired by Pan *et al.* [70], the optimization in Eq. 14 combines a funnel loss $\mathcal{L}_{\text{funnel}}$, collision loss $\mathcal{L}_{\text{col}}$, and smoothness loss $\mathcal{L}_{\text{smooth}}$. $\mathcal{L}_{\text{funnel}}$ preserves the pre-contact dynamics by encouraging the trajectories to converge to the same initial end-effector pose.

$$\mathcal{L}_{\text{funnel}} = \sum_t w_t \left\| \mathbf{t}_t^{ee} - \hat{\mathbf{t}}_t^{ee} \right\|_2^2, \tag{28}$$

Here, $\hat{\mathbf{t}}_t^{ee}$ and $\mathbf{t}_t^{ee}$ denote the original and optimized end-effector translation at timestep t respectively. The weights w_t increase from $w_{\min}$ to $w_{\max}$, which places greater emphasis on matching the demonstration near contact, while allowing more flexibility earlier in the trajectory.

$$w_t = w_{\min} + (w_{\max} - w_{\min}) \left(\frac{t}{T-1} \right)^3, \quad \sum_{t=0}^{T-1} w_t = 1. \tag{29}$$

$\mathcal{L}_{\text{col}}$ prevents end-effector penetration into the object by penalizing end-effector vertices V^{ee} that lie inside the object

TABLE V: Training hyperparameters per task. Img-H: image observation horizon; P-H: proprioception observation horizon; Act-H: action horizon; Freq: rollout frequency; Speed: rollout frequency relative to data collection; V-Arch: vision encoder architecture; Compute: GPU configuration used for training.

Task	Img-H	P-H	Act-H	Freq	Speed	V-Arch	Compute
Rotate Box	2	2	8	10	0.33	ViT-B/16	4xV100
Pour Mug	2	2	8	10	0.33	ViT-B/16	4xV100
Bottle from Rack	2	2	12	10	0.33	ViT-B/16	4xV100
Wipe Brush	2	2	12	10	0.33	ViT-B/16	4xV100
Can on Plate	2	2	8	10	0.33	ViT-B/16	4xV100
Can on Plate OOD	1	2	8	5	0.17	ViT-L/14	4xL40

TABLE VI: Common training hyperparameters.

Hyperparameter	Value
Image Resolution	224×224
Pretrained Vision Encoder	CLIP[1]
Denoising Steps	50
Optimizer	AdamW
Base Learning Rate	3×10^{-4}
Learning Rate Scheduler	Cosine Decay
Weight Decay	1×10^{-6}
Momentum	$\beta_1, \beta_2 = 0.95, 0.999$
Warmup Steps	2000
Training Epochs	120
Batch Size per GPU	64
Proprioception Input	relative eef xyz, relative 6d rotation, binary gripper open/close
Action Output	relative eef xyz, relative 6d rotation, binary gripper open/close

[1] A. Radford *et al.*, *Learning Transferable Visual Models from Natural Language Supervision*, arXiv:2103.00020, 2021.

surface. This is evaluated using the object TSDF Φ^{obj}.

$$\mathcal{L}_{\text{col}} = \sum_{\mathbf{v}^{ee} \in V^{ee}} \Phi^{obj}(\mathbf{v}^{ee}) \tag{30}$$

Lastly, $\mathcal{L}_{\text{smooth}}$ encourages temporally smooth motion.

$$\mathcal{L}_{\text{smooth}} = \sum_{t} \|\mathbf{t}^{ee}_{t+1} - \mathbf{t}^{ee}_{t}\|_2^2 + \|\log\big((\mathbf{R}^{ee}_{t})^\top \mathbf{R}^{ee}_{t+1}\big)\|_2^2 \tag{31}$$

3) Contact Grasp Refinement: At the contact start frame t_s, the end-effector pose $\mathbf{T}^{ee}_{t_s}$ and gripper width g_{t_s} are refined to ensure a physically plausible grasp. For both the thumb and index fingertips, 50 contact points $V^{\text{contact}} \subset V^{obj}$ are identified based on the hand mesh's proximity to the object mesh (Fig. 4(c-d)). Using these contact points, the end-effector position and gripper width are optimized to align the grasp with the hand-held object

$$\min_{\mathbf{T}^{ee}_{t_s},\, g_{t_s}} \quad \lambda_{\text{contact}}\mathcal{L}_{\text{contact}} + \lambda_{\text{width}}\mathcal{L}_{\text{width}} + \lambda_{\text{col}}\mathcal{L}_{\text{col}} \tag{32}$$

where

$$\begin{aligned} \mathcal{L}_{\text{contact}} &= \sum_{\mathbf{v}^{ee} \in V^{ee}} \min_{\mathbf{v}^{c} \in V^{\text{contact}}} \|\mathbf{v}^{ee} - \mathbf{v}^{c}\|_2^2 \\ \mathcal{L}_{\text{width}} &= g_{t_s} \end{aligned} \tag{33}$$

and $\mathcal{L}_{\text{col}}$ is the same gripper-object collision loss defined in Eq. 30. $\mathcal{L}_{\text{width}}$ encourages the gripper to be as closed as possible while $\mathcal{L}_{\text{col}}$ prevents gripper-object penetration.

G. Training Details

We use a diffusion policy based on the UMI implementation. Task-specific hyperparameters are reported in Table V, while hyperparameters shared across all tasks are listed in Table VI.

H. Out-of-Distribution Scenes

Visualizations of training and testing scenes for the out-of-distribution experiments (Sec. IV-D) are shown in Fig. 11.

I. Object Pose Tracking Visual Comparison

We provide qualitative visualizations in Fig. 12 comparing object pose tracking with FoundationPose [96] to the presented hand–object optimization from Sec. III-D. As discussed in Sec. IV-D, a qualitative success-rate analysis is reported for the Rotate Box and Can on Plate tasks.

J. Data Processing Times

Data processing times averaged across all tasks and demonstrations are reported in Table VII. This includes operations performed once per task and once per demonstration. Augmentation is reported separately, as augmentation is performed multiple times per demonstration, while retexturing can be reused across demonstrations. Reported timings correspond to an unoptimized sequential implementation and depend on the specific CPU and GPU hardware used. Although processing requires approximately seven minutes per demonstration, all steps are fully automated and can be executed offline and in parallel with other demonstrations. This shifts effort away from human data collection and towards automated computation, which is more efficient to scale and less labor intensive. Future work will focus on reducing processing time through parallelization and pipeline-level optimizations.

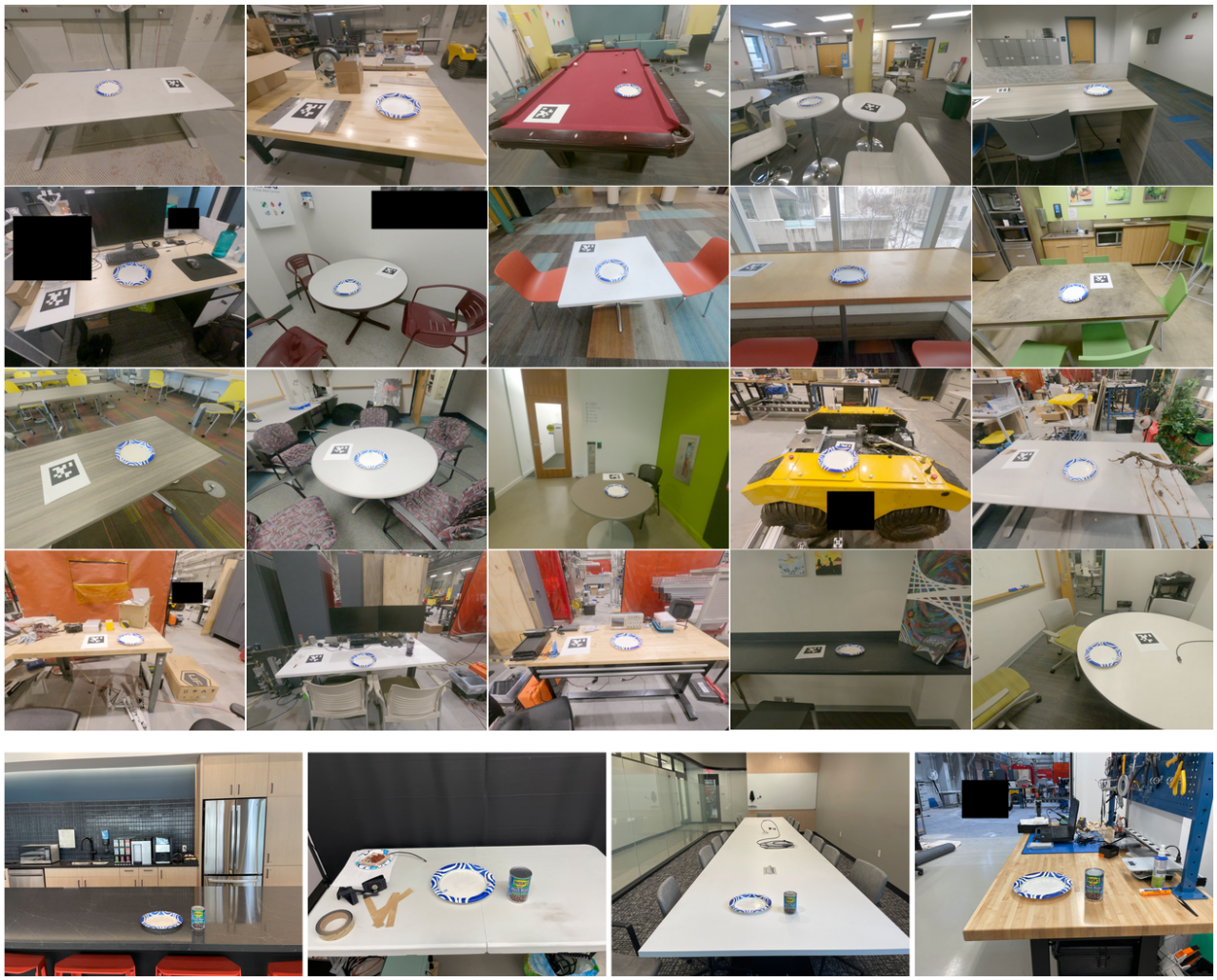

Fig. 11: Scenes used in the out-of-distribution experiments. The top four rows show training scenes, and the bottom row shows scenes used for evaluation.

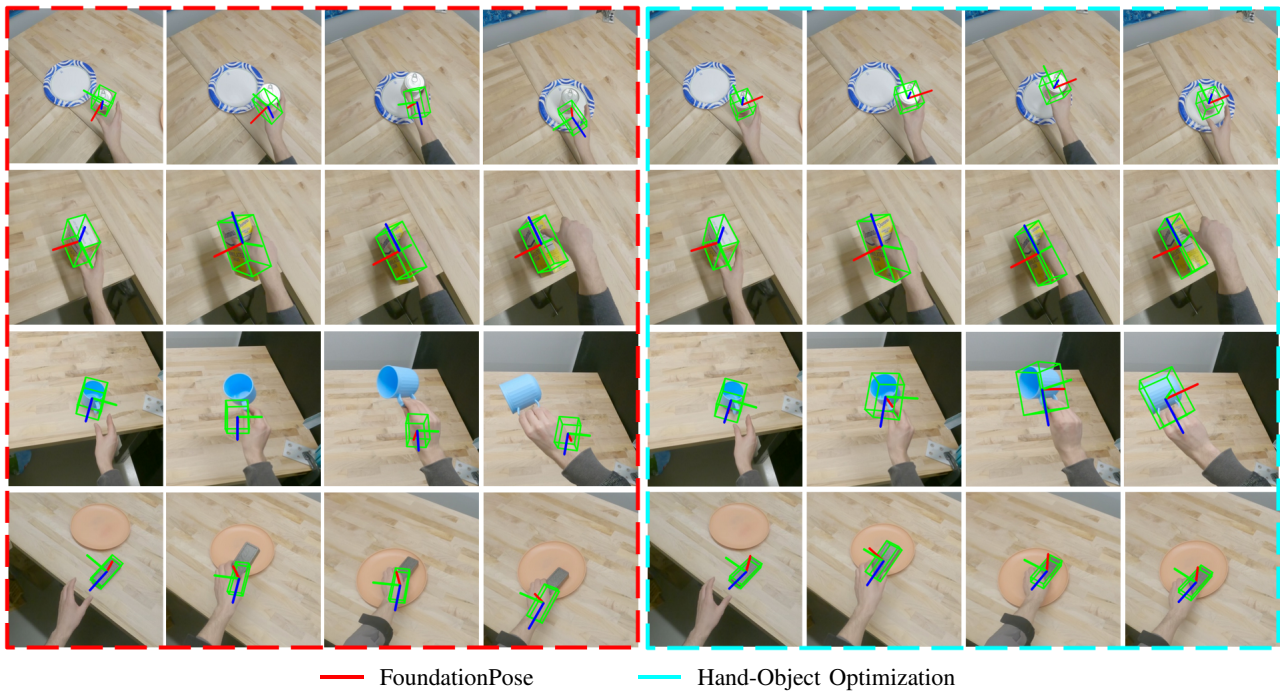

Fig. 12: Qualitative comparison of object pose tracking between FoundationPose and the hand–object optimization used by WARPED. Hand–object optimization more reliably tracks object pose, particularly for smaller objects.

TABLE VII: Data processing times averaged across all tasks and demonstrations on an AMD Ryzen 9 7950X CPU and an NVIDIA GeForce RTX 3090 GPU. Timings reflect a sequential, unoptimized implementation.

Stage	**Operation**	**Time**
Once per task		
Scene Reconstruction	Structure-from-Motion	04:15
	Scene Gaussian Splat (15000 steps)	04:05
	Total (Scene Reconstruction)	**08:20**
Object Mesh Reconstruction	SAM3D Object Mesh Build	00:17
	Mesh Rendering	00:24
	Object Gaussian Splat (7000 steps)	01:07
	Total (Object Mesh Reconstruction)	**01:48**
Per demonstration		
Interactive Scene Initialization	Localization	00:33
	SpatialTrackerV2 Monocular Depth Prediction	00:20
	Scene-Level Scale Alignment	00:06
	Total (Scene Initialization)	**00:59**
Hand Pose Initialization	Hand Detection	00:16
	Hand Pose Refinement	00:29
	Total (Hand Pose Initialization)	**00:45**
Object Pose Initialization	Grounding DINO Object Detection	00:02
	SAM2 Mask Propagation	00:10
	MegaPose Pose Estimation	00:08
	Object Pose Refinement	00:28
	Total (Object Pose Initialization)	**00:48**
Hand-Object Optimization	Object Pose Estimation	01:19
	Joint Hand-Object Refinement	02:33
	Total (Hand-Object Optimization)	**03:52**
Retargeting and Rendering	Pre-Contact Trajectory Optimization	00:07
	Contact Grasp Refinement	00:12
	Wrist-Camera Rendering	00:14
	Total (Retargeting and Rendering)	**00:33**
	Total (Per demonstration)	**06:57**
Augmentation		
Shared Across Augmentations	Retexturing (Trellis[2])	00:20
Per-Augmentation Pass	Pose and Viewpoint Randomization	00:04
	Wrist-Camera Rendering	00:14
	Total (Per augmentation pass)	**00:18**

[2] Xiang et al., "Structured 3D Latents for Scalable and Versatile 3D Generation," arXiv:2412.01506, 2024.