arXiv:2406.06843v4 [cs.CV] 11 Mar 2025

HO-Cap: A Capture System and Dataset for 3D Reconstruction and Pose Tracking of Hand-Object Interaction

Jikai Wang[1] Qifan Zhang[1] Yu-Wei Chao[2] Bowen Wen[2] Xiaohu Guo[1] Yu Xiang[1]

[1]The University of Texas at Dallas
{jikai.wang, qifan.zhang, xguo, yu.xiang}@utdallas.edu

[2]NVIDIA
{ychao, bowenw}@nvidia.com

Figure 1. Examples of RGB frames and renderings of the 3D shape and pose annotations of hands and objects in our dataset to images and in the NVIDIA Isaac Sim simulation.

Abstract

We introduce a data capture system and a new dataset, HO-Cap, for 3D reconstruction and pose tracking of hands and objects in videos. The system leverages multiple RGB-D cameras and a HoloLens headset for data collection, avoiding the use of expensive 3D scanners or mocap systems. We propose a semi-automatic method for annotating the shape and pose of hands and objects in the collected videos, significantly reducing the annotation time compared to manual labeling. With this system, we captured a video dataset of humans interacting with objects to perform various tasks, including simple pick-and-place actions, handovers between hands, and using objects according to their affordance, which can serve as human demonstrations for research in embodied AI and robot manipulation. Our data capture setup and annotation framework will be available for the community to use in reconstructing 3D shapes of objects and human hands and tracking their poses in videos.[1]

1. Introduction

Hand-object interaction has been a key area of research with broad applications in human-computer interaction, VR/AR, and robot learning from human demonstration. Specific problems, such as hand detection [32, 53]; hand pose estimation [15, 16]; hand shape reconstruction [34, 37, 44]; hand-object detection [9, 40]; object pose estimation [6, 26, 47, 48]; and object shape reconstruction [8, 46, 51], are actively studied within the community. To facilitate research and benchmarking of these problems, several datasets related to hands and objects have been introduced [7, 12, 20, 24, 29, 50]. Most datasets consist of videos of users manipulating objects in front of cameras,

[1]Data, code and videos for the project are available at https://irvlutd.github.io/HOCap.

with ground truth annotations such as hand poses and object poses obtained through various methods.

An easy way to obtain pose annotations is to use motion capture (mocap) systems [4, 12]. However, mocap systems are not only expensive, but also require the use of artificial markers on hands and objects during data capture. Alternatively, several datasets, such as DexYCB [7], OakInk [50] and HOI4D [29], rely on manual labeling. Given the large number of video frames, human annotation is highly time-consuming. Recently, a few systems have been proposed to automatically or semi-automatically generate pose annotations for hands and objects [18, 24]. However, these systems are not scalable to a wide variety of objects or hand-object interactions. For example, HO-3D [18] is limited to objects with known 3D models and cannot handle unseen objects. Similarly, H2O [24] requires training an object tracker for each captured object, making it difficult to scale to a large number of objects.

In this work, we introduce a new capture system for hand-object interaction. The system utilizes eight calibrated RGB-D cameras and a HoloLens headset [1] to provide both third-person and ego-centric views. We propose a semi-automatic annotation method that can accurately obtain 3D shape and pose annotations for hands and objects in videos. Unlike previous systems, our method does not rely on motion capture markers or expensive 3D scanners for reconstructing 3D object models, and it requires no domain-specific training. These properties make our capture system scalable and easily deployable for hand-object interaction, surpassing existing approaches [18, 24].

Specifically, we leverage recent large pre-trained vision models, including MediaPipe [31] for hand detection and pose estimation, BundleSDF [45, 46] for 3D object reconstruction and pose estimation, FoundationPose [47] for object pose tracking, and SAM2 [38] for object segmentation and tracking. To address noise and errors, we utilize multi-view consistency across eight RGB-D cameras. Additionally, we propose a Signed Distance Field (SDF)-based optimization method to refine both hand and object poses in 3D space. Consequently, our annotation pipeline can automatically process captured RGB-D videos to generate 3D shapes and poses for hands and objects. The only human annotation required is to manually select two points for each object in the first frame to generate an initial segmentation mask using SAM2 [38], and label the object name to register it in our database.

Using our capture system and annotation method, we created a new dataset called HO-Cap for hand-object interaction research. The dataset includes human demonstration videos of uni- and bi-manual interactions with objects, covering three interaction types: affordance-driven object use, pick-and-place, and handovers. It contains 64 videos with 656K RGB-D frames, captured from 9 subjects interacting with 64 objects. Ground-truth annotations of 3D shape and pose for hands and objects are provided for every frame. Fig. 1 shows some examples, where we also render the annotations in the NVIDIA Isaac Sim simulator.

Our dataset serves as a valuable benchmark for various hand-object recognition tasks. Specifically, we present baseline results for CAD-based object detection, open-vocabulary object detection, hand pose estimation, and unseen object pose estimation. The dataset can be used for training models for hand-object interaction or be used to test zero-shot capabilities of models trained on external data, enabling evaluation of large models for hands and object recognition. Additionally, the hand and object trajectories in the dataset can be used as human demonstrations for research in embodied AI and robot manipulation.

The contributions of this work are as follows:

- We introduce a data capture system and a semi-automatic annotation method for obtaining 3D shapes and poses of hands and objects from multi-view RGB-D videos.
- We introduce a new dataset for hand-object interaction, focusing on humans performing tasks with objects. It covers diverse grasping and multi-object rearrangement tasks, which are novel and valuable for the imitation learning community.
- We provide a benchmark with baseline results for object detection, hand pose estimation, and object pose estimation, which can benefit future reseach using our dataset.

2. Related Work

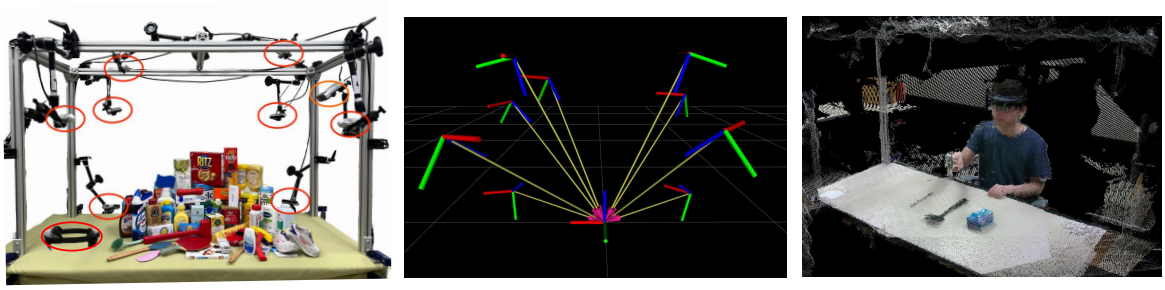
In recent years, a number of hand-object interaction datasets have been introduced. Representative datasets are summarized in Table 1 compared to ours.

Mocap vs. Natural capture. A straightforward way to obtain hand pose and object pose is to use mocap systems. By attaching reflective markers to the hands and objects, a mocap system can track these markers to obtain the hand pose and the object pose (e.g., FPHA [14], ContactPose [4], ARCTIC [12] and OakInk2 [52]). However, mocap systems are costly, require calibration between mocap markers and image cameras, and introduce artifacts. In contrast, our multi-camera setup captures markerless data, eliminating the need for such calibration.

Manual labeling vs. Automatic labeling. Large-scale datasets such as DexYCB [7], OakInk [50], and HOI4D [29] rely on manual labeling, which, though accurate, is labor-intensive. In contrast, some datasets use automated labeling (e.g., HO-3D [18]) or semi-automated labeling (e.g., H2O [24] and SHOWMe [42]). Fully automated methods often introduce annotation errors, while semi-automatic approaches integrate manual error correction [24] or initialize tracking processes manually [42]. Recently, HANDAL [17] introduced a semi-automatic pipeline for 6D object pose annotation, but it includes only limited dynamic human-object

Table 1. Comparison of our HO-Cap with recent hand-object interaction datasets.

dataset	year	modality	#seq.	#frames	#subj.	#obj.	#views	real image	marker-less	bi-manual	object reconst.	task	label
FPHA [14]	2018	RGB-D	1,175	105K	6	4	ego	✓	×	×	×	multi-task	mocap
Obman [19]	2019	RGB-D	–	154K	20	3K	1	×	✓	×	×	grasping	synthetic
HO-3D [18]	2020	RGB-D	27	78K	10	10	1-5	✓	✓	×	×	grasping & manipulation	automatic
ContactPose [4]	2020	RGB-D	2,303	2,991K	50	25	3	✓	×	✓	×	grasping & manipulation	mocap & thermal
GRAB [43]	2020	mesh	1,335	1,624K	10	51	–	×	×	✓	×	grasping	mocap
DexYCB [7]	2021	RGB-D	1,000	582K	10	20	8	✓	✓	×	×	grasping & handover	manual
H2O [24]	2021	RGB-D	–	571K	4	8	4+ego	✓	✓	✓	✓	multi-task	semi-auto
OakInk [50]	2022	RGB-D	792	230K	12	100	4	✓	×	×	✓	multi-task	manual
HOI4D [29]	2022	RGB-D	4,000	2,400K	4	800	ego	✓	✓	×	✓	multi-task	manual
AffordPose [20]	2023	mesh	–	–	–	641	–	×	–	×	×	multi-task	synthetic
SHOWMe [42]	2023	RGB-D	96	87K	15	42	1	✓	✓	×	✓	grasping	semi-auto
ARCTIC [12]	2023	RGB	399	2,100K	10	11	8+ego	✓	×	✓	✓	bimanual manipulation	mocap
OakInk2 [52]	2024	RGB	627	4.01M	9	75	3+ego	✓	×	✓	✓	multi-task	mocap
Ours	2024	RGB-D	64	656K	9	64	8+ego	✓	✓	✓	✓	multi-task	semi-auto



(a) Our hardware setup (b) Visualization of the camera poses (c) Point clouds from the cameras

Figure 2. Illustration of our data capture setup.

interactions and lacks hand annotations.

In our work, we introduce a semi-automatic pipeline for annotating 3D shapes and poses of both hands and objects in videos. The only required human input is selecting two points on the object in the initial frame as prompts for SAM2 [38] segmentation. The method then automatically annotates subsequent frames. As highlighted in Table 1, our dataset contains markerless unimanual and bimanual videos, with the capability for 3D shape reconstruction of novel objects. The most similar work to ours is H2O [24]. However, unlike H2O, our approach requires no domain-specific training for object pose trackers. Instead, we leverage pre-trained vision models with a multi-camera setup, making our annotation method more scalable across diverse hand-object interactions.

3. Data Capture Setup

Our hardware setup, illustrated in Fig. 2(a), consists of eight Intel RealSense D455 cameras and one Microsoft Azure Kinect [2] positioned above a table. These RGB-D cameras provide full workspace coverage, with the higher-resolution Kinect primarily used for 3D object reconstruction. We calibrated the intrinsic and extrinsic parameters of each camera using Vicalib [3], enabling fusion of point clouds into a common 3D coordinate frame (Fig. 2(b)-(c)).

Additionally, users wear a Microsoft HoloLens AR headset [1] during data collection (Fig. 2(c)), allowing us to capture synchronized ego-centric (first-person) and third-person RGB-D videos. We also track the 6D poses of the HoloLens headset, enabling fusion of its point clouds with those from the RealSense cameras.

4. Annotation Method

Our goal is to provide 3D shapes and poses of hands and objects in the captured videos, with the ability to handle arbitrary objects whose 3D models were not available before the capture. We propose a semi-automatic annotation method based on a multi-view camera setup, eliminating the need for expensive 3D scanners or mocap systems.

4.1. 3D Object Reconstruction

Our annotation process begins by reconstructing 3D models of the objects. Instead of relying on pre-existing 3D meshes (e.g., YCB Object Set [5]) or 3D scanners [42], we use BundleSDF [46], a neural reconstruction method, to reconstruct a textured 3D mesh for each object, assuming that the object is rigid. BundleSDF uses a sequence of RGB-D images along with the segmentation masks of the object to precisely track its 6D pose and reconstruct a textured mesh. This enables us to use only an RGB-D camera for object reconstruction and obtain 3D meshes for various objects, as shown in Fig. 3.

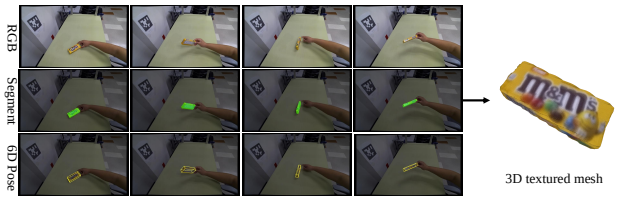


Figure 3. Illustration of our pipeline for 3D object reconstruction.

To prepare the input data, we manually move and rotate an object in front of the Azure Kinect camera, ensuring exhaustive coverage of the surfaces for high-fidelity reconstruction. For segmentation, we prompt the SAM2 [38], a unified model for segmenting objects across images and videos, with two manually selected points in the initial frame to track the object mask throughout the remaining frames. Given the object masks, BundleSDF leverages feature matching based on LoFTR [41] for coarse pose initialization, followed by an online pose graph optimization to estimate the objects' 6D poses in video frames. Simultaneously, a neural object field is trained to model object-centric geometry and appearance, refining keyframe poses to reduce tracking drift. Finally, a textured 3D mesh is extracted from the neural field using marching cubes [30] and color projection. In total, we reconstructed 64 objects in our dataset all of which are easily available for purchase online. For visualizations of these reconstructed objects, please refer to the supplementary material.

4.2. Object Pose Estimation

After obtaining the 3D object models, we use them to represent and estimate object poses in videos where subjects manipulate these objects. The pipeline for hand and object pose estimation is shown in Fig. 4.

4.2.1. Object Pose Initialization

In our framework, each object's pose is represented as a homogeneous transformation $\mathbf{T} = (\mathbf{R}, \mathbf{t}) \in \mathbb{SE}(3)$, where $\mathbf{R}$ and $\mathbf{t}$ denote the object's 3D rotation and translation relative to the world frame. The world frame is defined near the center of the table (see Fig. 2(b)). Since all cameras are extrinsically calibrated, the transformations between camera frames and the world frame are known. Our pose estimation algorithm initializes each object's pose using the pose estimation results from FoundationPose [47], a unified foundation model for 6D object pose estimation and tracking, and then refines the object pose with SDF optimization on the segmented point cloud.

First, following the segmentation process in Section 4.1, we obtain segmentation masks for each object throughout the sequence (examples shown in Fig. 4). To determine an object's initial pose at step t, denoted as $\mathbf{T}_t = (\mathbf{R}_t, \mathbf{t}_t)$, we use FoundationPose to track the object pose across all camera views. Since FoundationPose operates on single-camera input and is affected by occlusions during manipulation tasks, the tracked pose may be inaccurate, leading to tracking failures in subsequent frames for individual views. To ensure robust tracking across the video sequence, we optimize pose consistency across views by aligning tracked poses from multiple camera perspectives to reduce discrepancies and enhance accuracy.

Specifically, given the tracked pose $\mathbf{T}c_i$ from each camera view c_i, we transform these poses from their respective camera frames into the world frame. If the distribution of pairwise distances between all the transformed poses varies significantly, indicating that the poses are noisy, the optimization process is skipped, and the pose from the previous frame is used instead. If the distances between all transformed poses and the previous pose $\mathbf{T}_{t-1}$ are within a predefined threshold, suggesting no significant pose change, we incorporate the previous frame's pose into the RANSAC algorithm [13] to improve robustness in identifying outliers. Finally, the RANSAC algorithm is applied to identify and filter out outliers based on the computed distances, resulting in a single, coherent pose for the object. The optimized initial pose $\mathbf{T}_t$ in the world frame is then projected back into each camera frame and provided as input for FoundationPose to track the object in the next frame, ensuring reliable and accurate tracking across frames.

4.2.2. SDF-based Object Pose Tracking

Although these initial poses may be effective for FoundationPose to track in the next frame, small inaccuracies and misalignment with the object point clouds can still occur. These errors can arise due to occlusions, camera noise, or drift over time. To address this, we apply a Signed Distance Field (SDF)-based algorithm to optimize the poses across the sequence. Our SDF-based pose tracking leverages the complete fused point cloud from all eight cameras, resulting in higher-quality pose annotations.

At time step t, our goal is to optimize the initial object pose $\mathbf{T}_t$ given the previous pose $\mathbf{T}_{t-1}$, where $t = 1, 2, \ldots, T$ and T is the total number of frames. We minimize the following loss function

$$\mathbf{T}_t^* = \arg\min_{\mathbf{T}_t} \left(L_{\text{sdf}}(\mathbf{T}_t) + \lambda_1 L_{\text{smooth}}(\mathbf{T}_t, \mathbf{T}_{t-1}) \right), \quad (1)$$

where λ_1 is a weight to balance the two loss terms. The SDF loss function is defined as

$$L_{\text{sdf}}(\mathbf{T}_t) = \frac{1}{|\mathcal{X}_t|} \sum_{\mathbf{x} \in \mathcal{X}_t} |\text{SDF}(\mathbf{x}, \mathbf{T}_t)|^2, \quad (2)$$

where $\mathcal{X}_t$ denotes a set of 3D points of the object at time step t in the world frame, which is fused from multiple camera views. The function $\text{SDF}(\mathbf{x}, \mathbf{T}_t)$ computes the signed distance of a 3D world point $\mathbf{x} \in \mathbb{R}^3$ transformed into the object frame according to $\mathbf{T}_t$. Minimizing the SDF loss func-

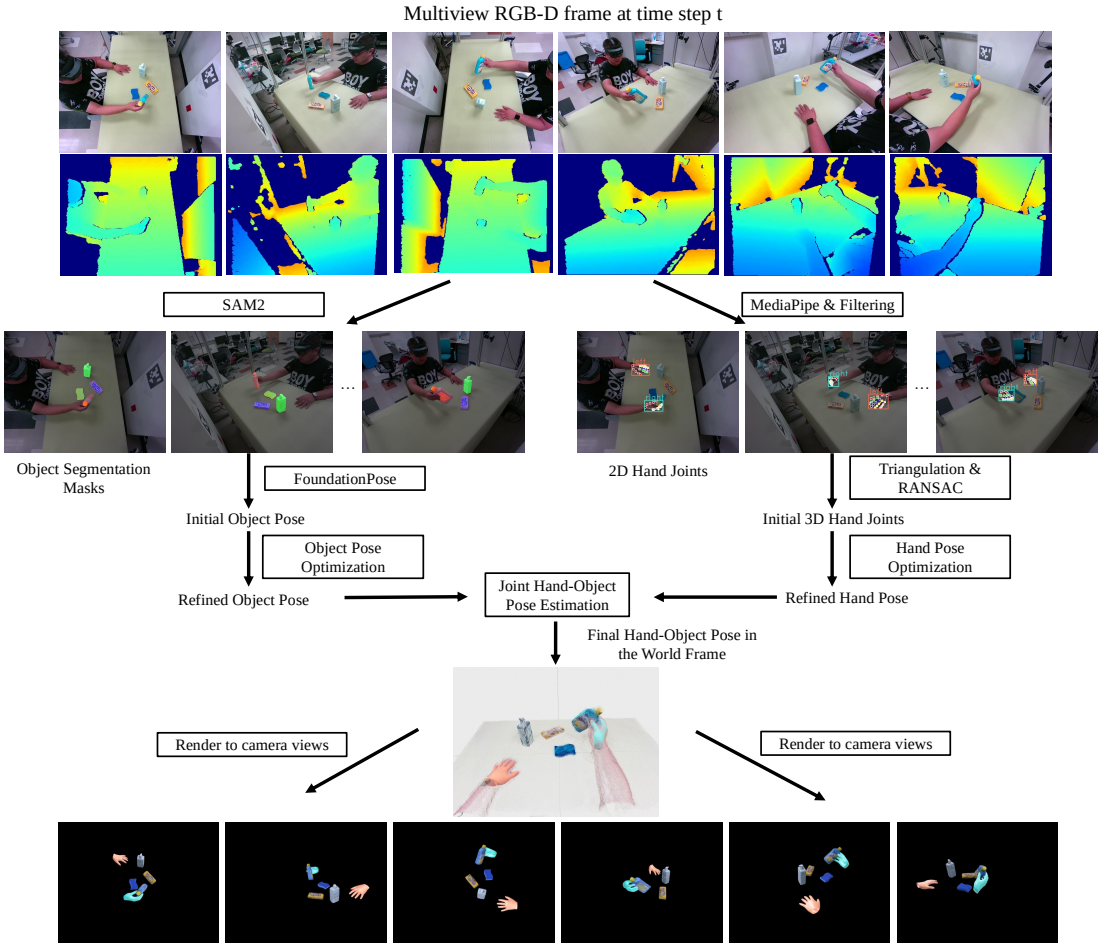


Figure 4. Illustration of our pipeline for obtaining poses of hands and objects from multi-view RGB-D videos.

tion results in an object pose that aligns the transformed 3D points with the object model surface.

The smoothness loss term is defined as

$$L_{\text{smooth}}(\mathbf{T}_t, \mathbf{T}_{t-1}) = \|\mathbf{q}_t - \mathbf{q}_{t-1}\|^2 + \|\mathbf{t}_t - \mathbf{t}_{t-1}\|^2, \quad (3)$$

where $\mathbf{q}_t$ and $\mathbf{q}_{t-1}$ are the quaternions of the 3D rotations, and $\mathbf{t}_t$ and $\mathbf{t}_{t-1}$ denote the 3D translations, respecively. The smoothness term prevents large jumps of the poses during optimization. By solving Eq. (1) for every object and every time step, we obtain the poses of all the objects in the video sequence. These poses provide initializations which will be further refined jointly with the hands, as detailed in Sec. 4.4.

4.3. Hand Pose Estimation

We use MANO model [39] to represent human hands. This parametric model enables detailed hand mesh modeling through shape and pose parameters. The shape parameters, $\beta \in \mathbb{R}^{10}$, capture individual hand identity, reflecting variations among different subjects, while the pose parameters, $\theta \in \mathbb{R}^{51}$, describe the dynamic positions and orientations of the hand. Before optimizing hand poses, we pre-calibrate the MANO shape parameters for each subject in our dataset (see details in the supplementary material).

In our initial attempts to optimize hand poses, we applied the same loss function used in Sec. 4.2 for object pose estimation, expecting similar results. However, practical experimentation revealed unique challenges that required a different approach for hand pose optimization. A primary issue was the difficulty in accurately segmenting the hand using the SAM2 method, which often resulted in segmentation masks that included the forearm and other non-hand areas. This lack of precision in hand segmentation made it challenging to isolate the hand accurately for pose optimization. Additionally, we found that the higher dimensionality of the MANO hand pose parameter θ, compared to the 6D pose of objects, led to overfitting when using only the SDF loss, often resulting in unrealistic hand poses.

To address these challenges, we recognized the need to incorporate additional robust constraints into our hand pose optimization method. Specifically, we introduced 2D/3D hand keypoints as anchor points to guide the optimization process. These keypoints serve as strong priors, helping to ensure more realistic and physically plausible hand poses. At time step t of an input video, our goal is to estimate the MANO hand pose θ_t of a hand. We solve the following

optimization problem to estimate the hand pose:

$$\theta_t^* = \arg\min_{\theta_t} \left(L_{\text{keypoint}}(\theta_t) + \lambda_2 L_{\text{reg}}(\theta_t) \right), \tag{4}$$

where L_{keypoint} is a loss function based on the estimated 3D keypoints of the hand, L_{reg} is a regularization term, and λ_2 is a weight to balance the two terms. The regularization term is simply defined as the squared L2 norm of the pose parameter: $L_{\text{reg}}(\theta_t) = \|\theta_t\|^2$.

4.3.1. 3D Keypoint Loss Function

To generate accurate 3D keypoints for hand pose optimization, we first detect 2D hand landmarks across multiple views using MediaPipe [31]. While MediaPipe can comprehensively identify all 21 hand joints, it lacks confidence scores, making it challenging to distinguish between occluded or inaccurately estimated joints. This can introduce inaccuracies in the pose optimization process.

To address these challenges, we use a RANSAC [13] filtering approach to improve the accuracy of our 3D keypoints. For each hand joint, assume that it is detected by MediaPipe on C_{valid} camera views. We compute a set of candidate 3D keypoints, $\mathcal{X}_i = \{\mathbf{x}_i\}_{i=1}^{N_{\text{valid}}}$, $N_{\text{valid}} = C_{\text{valid}}(C_{\text{valid}} - 1)/2$ by triangulating pairs of valid views. A projection loss function is defined for a 3D keypoint:

$$L_{\text{proj}}(\mathbf{x}_i) = \sum_{c \in C_{\text{valid}}} \|\Pi^c(\mathbf{x}_i) - \mathbf{h}_i^c\|^2, \tag{5}$$

where $\Pi^c(\cdot)$ is the projection function for camera c, and $\mathbf{h}_i^c$ represents the detected 2D landmark for the 3D keypoint $\mathbf{x}_i$ in camera c. This loss measures the discrepancy between each candidate 3D keypoint and its corresponding 2D landmarks across views. Minimizing this projection loss refines the 3D keypoint positions, and we select the candidate with the lowest projection error for each joint. This critical step helps exclude views affected by incorrect MediaPipe detections, enhancing the reliability of the derived 3D keypoints. These optimized 3D keypoints serve as robust priors in the subsequent optimization stages.

For video frames where MediaPipe does not successfully detect hand landmarks across all camera views, we found these instances to be a minor portion of the video sequence. Additionally, the frames affected are temporally close to the ones with accurate hand pose detections, typically exhibiting minimal variance in hand pose. This observation suggests that linear interpolation between adjacent frames with reliably detected 3D hand joints is an effective strategy for estimating missing data. After filling these gaps via linear interpolation, we further refine the hand motion trajectory using cubic spline interpolation. This approach not only smooths the spatial transitions of the hand joints but also ensures continuity in the first and second derivatives of the motion trajectory, corresponding to the hand's velocity and acceleration. By implementing this refinement, we achieve a more cohesive and realistic representation of hand motion across the entire sequence, enhancing the fluidity and naturalness of the observed actions.

Using the estimated 21 3D hand joints $(\mathbf{x}_1, \ldots, \mathbf{x}_{21})$, the 3D keypoint loss function is defined as:

$$L_{\text{keypoint}}(\theta_t) = \frac{1}{21} \sum_{i=1}^{21} \|J_i(\theta_t) - \mathbf{x}_i\|^2, \tag{6}$$

where $J_i(\theta_t) \in \mathbb{R}^3$ represents the ith 3D hand joint from the MANO model under pose θ_t. Finally, solving the optimization problem in Eq. (4) will find the MANO hand pose θ_t that fits the estimated 3D keypoints.

4.4. Joint Hand-Object Pose Optimization

Separately solving hand and object poses can lead to unrealistic scenarios, such as intersections between hand and object meshes. To address this limitation and improve pose accuracy, we propose a joint pose optimization method that refines hand and object poses together. This approach reduces mesh intersections and enhances the physical realism of the estimated poses.

For a sequence with $N_H \in \{1, 2\}$ hands and N_O objects, at each time step t, we jointly refine the object poses $\mathcal{P}_t^O = \{\mathbf{T}_t^o\}_{o=1}^{N_O}$ and hand poses $\mathcal{P}_t^H = \{\theta_t^h\}_{h=1}^{N_H}$. Our idea is to utilize the SDFs of objects and hands to optimize the poses as in our object pose estimation method. The loss function for this joint optimization is defined as

$$L_{\text{joint}}(\mathcal{P}_t^O, \mathcal{P}_t^H) = \frac{1}{N_O} \sum_{o=1}^{N_O} \Big(\frac{1}{|\mathcal{X}_t^o|} \sum_{\mathbf{x} \in \mathcal{X}_t^o} |\text{SDF}_o(\mathbf{x}, \mathbf{T}_t^o)|^2 \Big)$$
$$+ \frac{1}{N_H} \sum_{h=1}^{N_H} \Big(\frac{1}{|\mathcal{X}_t^h|} \sum_{\mathbf{x} \in \mathcal{X}_t^h} |\text{SDF}_h(\mathbf{x}, \theta_t^h)|^2 + \lambda_3 \|\theta_t^h\|^2 \Big),$$

where $\mathcal{X}_t^o$ and $\mathcal{X}_t^h$ denote the segmented point clouds for object o and hand h at the time step t in the world frame. And SDF_o and SDF_h represent the signed distance fields of the object o and the hand h, respectively. λ_3 is a weight to balance the regularization term for the hand pose. Additionally, the smoothness loss (Eq. 3) is added to ensure the temporal consistency of object poses and hand global translation and rotation.

The point clouds of objects can be obtained using the depth images and the segmentation masks of the objects. To obtain point clouds for hands, given the optimized 3D hand keypoints, we can get the bounding box and 2D keypoints on camera images. Using these 2D keypoints along with the bounding box as input, we employ SAM2 [38] to generate high-quality hand masks. We further isolate hand points from the surrounding environment using a point-to-mesh distance threshold based on the initial hand pose.

Figure 5. Comparison between the published and refined HoloLens poses

Table 2. Ground-truth pose evaluation results in pixel errors.

Mean (std)	Object ↓	Left hand ↓	Right hand ↓
Initial Poses	3.67(±3.05)	5.31(±3.43)	5.16(±3.73)
SDF-Refined Poses	3.50(±1.70)	**4.58**(±2.73)	3.83(±2.29)
Jointly-Refined Poses	**3.42**(±1.73)	**4.58**(±2.72)	**3.82**(±2.29)

Hand and object poses are initialized from the previous stages of our annotation process, and joint optimization requires only a few refinement steps to achieve robust results. This process effectively reduces mesh intersections and enhances the realism of hand-object interactions, providing accurate and cohesive pose annotations. After estimating the poses of hands and objects in the world frame, we project their 3D shapes to the camera views and obtain 2D annotations of images as shown in Fig. 4.

4.5. HoloLens Camera Pose Estimation

During data collection, the HoloLens head pose data often exhibited jitter and non-uniform movement speeds, making it difficult to compute precise HoloLens camera poses in real time. Therefore, we cannot directly use the pose data from HoloLens. We applied an optimization technique to refine the head poses provided by HoloLens.

At each time step t, we have the HoloLens camera pose $\mathbf{T}_t^H$ in the world frame published from the device. To refine this camera pose, we obtain the optimized object poses $\mathcal{P}_t^O$ as described in Sec. 4.4. By treating the group of objects as a single merged entity, we define the combined object pose as $\mathbf{T}_t^O$ in the world frame. By transforming this object pose into the HoloLens camera frame (denoted as $\mathbf{T}_t^{OH}$), and assuming proper synchronization between the HoloLens and RealSense frames, we can proceed with further refinement. Using FoundationPose [47], we optimize the object pose $\mathbf{T}_t^{OH}$ in the HoloLens camera frame using the RGB image from HoloLens, resulting in a refined camera pose $\mathbf{T}_t^{*H}$. As shown in Fig. 5, this refinement reduces jitter and improves alignment accuracy for the projected objects in the HoloLens camera frame.

5. The HO-Cap Dataset

Dataset Statistics. Our HO-Cap dataset consists of 64 videos, capturing 9 participants performing 3 hand-object interaction tasks with 64 unique objects. The approximately 656K frames provide rich temporal information for studying dynamic interactions, where each frame is captured from 8 calibrated RealSense cameras plus a first-person-view camera from the HoloLens, facilitating 3D reconstruction and egocentric understanding. The 64 different objects, each with a textured 3D mesh model, enabling fine-grained 6D pose annotations, and a diverse set of 9 subjects ensures variability in hand shapes, grasping styles, and interaction patterns. The both single-hand and bimanual interactions, supporting tasks that require complex hand coordination. Table 3 provides handedness statistics for left/right hands and bi-manual interactions, and a breakdown of task categories. More dtails about the object shape and hand-object pose diversity are provided in supplementary material.

Table 3. Detailed Dataset Statistics. The statistics are grouped by handness (Right, Left, Both) and tasks (T1: pick-and-place, T2: handover between hands, T3: affordance usage).

Statistics	Handness			Tasks		
	R	L	B	T1	T2	T3
#sequence	35	8	21	28	21	15
#frames	41,327	8,373	23,244	29,706	23,244	19,994

Annotation Quality. Using our annotation method, we generated 3D shapes and world space poses for hands and objects. To validate the annotation accuracy, we randomly selected 800 images across 8 RealSense camera views and manually annotated visible hand joints and object keypoints. The object keypoints were chosen from predefined mesh vertices that are easily identifiable. We then computed the Euclidean distance between the 2D projections of our 3D annotations and these human labeled 2D points. Table 2 presents the evaluation results. 1) The final annotation error is within 5 pixels for both hands and objects, demonstrating the reliability of our method, and (2) The error consistently decreases across annotation refinement steps, validating the effectiveness of our optimization strategy.

6. Baseline Experiments

After building the dataset, we provide several baseline experimental results to demonstrate the usage of our dataset.

6.1. Hand Pose Estimation

First, our dataset supports hand pose estimation by providing annotations for the 2D and 3D positions of 21 hand joints. In this experiment, we evaluated two recent hand pose estimation models: A2J-Transformer [21] and HaMeR [37] which are trained on external data, using ground truth hand bounding boxes as inputs. The evaluation results are presented in Table 4. We used the PCK (Percentage of Correct Keypoints) metric for 2D hand pose estimation and the MPJPE (Mean Per Joint Position Error) metric for 3D hand pose estimation. A2J-Transformer extends the depth-based A2J [49] to RGB input and incorporates a

Table 4. Evaluation of hand pose estimation. The numbers in parentheses are thresholds for PCK, and unit for MPJPE.

Method	PCK(0.05) ↑	PCK(0.1) ↑	PCK(0.15) ↑	PCK(0.2) ↑	MPJPE (mm) ↓
A2J-Transformer [21]	12.1	26.8	39.4	50.5	78.7
HaMeR [37]	43.7	79.2	88.5	91.4	28.9

transformer architecture to capture non-local information. This model was trained on the InterHand2.6M dataset [33]. HaMeR [37] uses a large-scale ViT backbone [11] followed by a transformer decoder to regress the parameters of the hand, and was trained on a large-scale dataset of 2.7M images from 10 hand pose datasets. In Table 4, HaMeR significantly outperforms A2J-Transformer, likely due to its ViT backbone and large-scale training data. The MPJPE of HaMeR is 28.9mm, which is much larger than the errors reported in other datasets in [37]. This suggests our dataset presents unique challenges for hand pose estimation, especially in cases of occlusions between hands and objects.

6.2. Object Detection

We evaluated object detection in two scenarios: novel object detection and seen object detection. For novel object detection, the model detects objects not encountered during training. While for seen object detection, models are trained specifically on our dataset. CNOS [35] and GroundingDINO [28] served as baselines for novel object detection, while YOLO11 [22] and RT-DETR [54] were trained for seen object detection on our dataset.

Novel Object Detection Baselines: CNOS [35] is a CAD-based approach that generates object templates by rendering CAD models and uses SAM [23] and DINOv2 CLS tokens [36] to classify proposals based on template similarity. GroundingDINO [28], a vision-language model, was tested with concatenated object names from our dataset as text prompts to detect objects.

Seen Object Detection Baselines: We trained YOLO11 and RT-DETR on our train/val split, which includes all subjects, views, and objects, and evaluated them on a test split with sequences not shared with train/val split. YOLO11, the latest in the YOLO series, features improved architecture and training methods for efficient detection. RT-DETR, a transformer-based model, is designed for robust performance across diverse object scales and complex scenes.

The results, shown in Table 5 with MSCOCO [27] Average Precision (AP) metrics, reveal challenges in novel object detection, as CNOS and GroundingDINO produced many false positives due to mismatches and ambiguities. For seen object detection, YOLO11 and RT-DETR both show particularly strong performance on medium and large objects. These results highlight the utility of our dataset for training object detectors, with RT-DETR and YOLO11 excelling on objects seen during training.

Table 5. Evaluation of object detection. Mean AP on all 64 objects in our dataset. Marker * indicates trained on our dataset.

Method	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
CNOS [35]	25.3	27.9	24.8	1.6	27.6	24.9
GroundingDINO [28]	17.0	27.6	21.5	1.4	24.3	7.5
YOLO11* [22]	71.4	85.9	78.7	20.7	75.2	72.6
RT-DETR* [54]	75.9	90.0	83.4	21.1	79.8	84.8

Table 6. Evaluation of object pose estimation for novel objects. The numbers are the AUC percentage of the ADD and ADD-S metrics on all 64 objects in our dataset.

Method	ADD (%)	ADD-S (%)
MegaPose [25]	67.1	83.0
FoundationPose [47]	89.3	95.7

6.3. Novel Object Pose Estimation

Traditional object pose estimation methods [10, 48] require the same objects to be used in both training and testing, preventing to generalize to new objects. Recent novel object pose estimation approaches have been developed to address this limitation by leveraging large-scale datasets and pre-trained models. Methods such as MegaPose [25] and FoundationPose [47] can estimate poses for unseen objects using only 3D models, without per-instance training. We thus evaluated MegaPose and FoundationPose on our dataset, providing ground truth 2D bounding boxes as input. Table 6 reports the AUC of ADD and ADD-S metrics [10, 48]. FoundationPose outperforms MegaPose, likely due to its powerful transformer-based architecture and extensive synthetic training. See supplemental for more visualizations.

7. Conclusion and Discussion

We introduced a novel capture system and dataset, HO-Cap, designed for hand-object interaction research. Our system uses multiple RGB-D cameras and a HoloLens to capture videos from third- and first-person views. A semi-automatic annotation method combines pre-trained vision models for 3D object reconstruction, segmentation, pose estimation, and hand joint detection, with no domain-specific training required. Additionally, an SDF-based optimization refines the hand and object poses. HO-Cap includes diverse human-object interactions and is a valuable resource for studying hand-object interaction, as well as for embodied AI and robotics manipulation research.

Limitations. Our annotation framework has three main limitations. First, we saw that BuddleSDF [46] cannot reconstruct certain types of objects very well, such as some textureless objects and metal objects. Therefore, we were unable to include these objects in our dataset. Second, MediaPipe [31] occasionally fails to detect hand joints accurately. Since our hand pose estimation relies on MediaPipe's detections, undetected hand joints prevent pose estimation, causing us to exclude videos with frequent MediaPipe detection failures. Finally, when small or cylindrical objects (such as spatulas or hammers) are held within the

hand, only minimal color, depth, or geometric information is visible, making it challenging to accurately estimate the object's pose, particularly its rotation. This limited visibility leads to inconsistencies across views and prevents reliable pose tracking in the world coordinate frame, causing a significant challenge for our annotation pipeline. As a result, such videos are also excluded. Future research could consider overcoming these limitations.

Acknowledgement

This work was supported in part by the DARPA Perceptually-enabled Task Guidance (PTG) Program under contract number HR00112220005, the Sony Research Award Program, and the National Science Foundation (NSF) under Grant No. 2346528.

References

[1] Microsoft hololens. https://www.microsoft.com/en-us/hololens. 2, 3

[2] Microsoft kinect. https://azure.microsoft.com/en-us/services/kinect-dk. 3

[3] Vicalib. https://github.com/arpg/vicalib. 3

[4] Samarth Brahmbhatt, Chengcheng Tang, Christopher D Twigg, Charles C Kemp, and James Hays. Contactpose: A dataset of grasps with object contact and hand pose. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIII 16*, pages 361–378. Springer, 2020. 2, 3

[5] Berk Calli, Aaron Walsman, Arjun Singh, Siddhartha Srinivasa, Pieter Abbeel, and Aaron M Dollar. Benchmarking in manipulation research: The ycb object and model set and benchmarking protocols. *arXiv preprint arXiv:1502.03143*, 2015. 3

[6] Pedro Castro and Tae-Kyun Kim. Crt-6d: Fast 6d object pose estimation with cascaded refinement transformers. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 5746–5755, 2023. 1

[7] Yu-Wei Chao, Wei Yang, Yu Xiang, Pavlo Molchanov, Ankur Handa, Jonathan Tremblay, Yashraj S Narang, Karl Van Wyk, Umar Iqbal, Stan Birchfield, et al. Dexycb: A benchmark for capturing hand grasping of objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9044–9053, 2021. 1, 2, 3

[8] Zerui Chen, Shizhe Chen, Cordelia Schmid, and Ivan Laptev. gsdf: Geometry-driven signed distance functions for 3d hand-object reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12890–12900, 2023. 1

[9] Ahmad Darkhalil, Dandan Shan, Bin Zhu, Jian Ma, Amlan Kar, Richard Higgins, Sanja Fidler, David Fouhey, and Dima Damen. Epic-kitchens visor benchmark: Video segmentations and object relations. *Advances in Neural Information Processing Systems*, 35:13745–13758, 2022. 1

[10] Xinke Deng, Arsalan Mousavian, Yu Xiang, Fei Xia, Timothy Bretl, and Dieter Fox. Poserbpf: A rao–blackwellized particle filter for 6-d object pose tracking. *IEEE Transactions on Robotics*, 37(5):1328–1342, 2021. 8

[11] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 8

[12] Zicong Fan, Omid Taheri, Dimitrios Tzionas, Muhammed Kocabas, Manuel Kaufmann, Michael J Black, and Otmar Hilliges. Arctic: A dataset for dexterous bimanual hand-object manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12943–12954, 2023. 1, 2, 3

[13] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981. 4, 6

[14] Guillermo Garcia-Hernando, Shanxin Yuan, Seungryul Baek, and Tae-Kyun Kim. First-person hand action benchmark with rgb-d videos and 3d hand pose annotations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 409–419, 2018. 2, 3

[15] Liuhao Ge, Yujun Cai, Junwu Weng, and Junsong Yuan. Hand pointnet: 3d hand pose estimation using point sets. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8417–8426, 2018. 1

[16] Jia Gong, Lin Geng Foo, Zhipeng Fan, Qiuhong Ke, Hossein Rahmani, and Jun Liu. Diffpose: Toward more reliable 3d pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13041–13051, 2023. 1

[17] Andrew Guo, Bowen Wen, Jianhe Yuan, Jonathan Tremblay, Stephen Tyree, Jeffrey Smith, and Stan Birchfield. Handal: A dataset of real-world manipulable object categories with pose annotations, affordances, and reconstructions. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 11428–11435. IEEE, 2023. 2

[18] Shreyas Hampali, Mahdi Rad, Markus Oberweger, and Vincent Lepetit. Honnotate: A method for 3d annotation of hand and object poses. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3196–3206, 2020. 2, 3, 1

[19] Yana Hasson, Gul Varol, Dimitrios Tzionas, Igor Kalevatykh, Michael J Black, Ivan Laptev, and Cordelia Schmid. Learning joint reconstruction of hands and manipulated objects. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11807–11816, 2019. 3

[20] Juntao Jian, Xiuping Liu, Manyi Li, Ruizhen Hu, and Jian Liu. Affordpose: A large-scale dataset of hand-object interactions with affordance-driven hand pose. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14713–14724, 2023. 1, 3

[21] Changlong Jiang, Yang Xiao, Cunlin Wu, Mingyang Zhang, Jinghong Zheng, Zhiguo Cao, and Joey Tianyi Zhou. A2j-transformer: Anchor-to-joint transformer network for 3d interacting hand pose estimation from a single rgb image. In

Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 8846–8855, 2023. 7, 8, 2, 4

[22] Glenn Jocher and Jing Qiu. Ultralytics yolo11, 2024. 8

[23] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. *arXiv preprint arXiv:2304.02643*, 2023. 8

[24] Taein Kwon, Bugra Tekin, Jan Stühmer, Federica Bogo, and Marc Pollefeys. H2o: Two hands manipulating objects for first person interaction recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10138–10148, 2021. 1, 2, 3

[25] 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. *arXiv preprint arXiv:2212.06870*, 2022. 8, 2, 6, 7

[26] Taeyeop Lee, Jonathan Tremblay, Valts Blukis, Bowen Wen, Byeong-Uk Lee, Inkyu Shin, Stan Birchfield, In So Kweon, and Kuk-Jin Yoon. Tta-cope: Test-time adaptation for category-level object pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21285–21295, 2023. 1

[27] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014. 8

[28] 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. 8, 2

[29] Yunze Liu, Yun Liu, Che Jiang, Kangbo Lyu, Weikang Wan, Hao Shen, Boqiang Liang, Zhoujie Fu, He Wang, and Li Yi. Hoi4d: A 4d egocentric dataset for category-level human-object interaction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21013–21022, 2022. 1, 2, 3

[30] William E Lorensen and Harvey E Cline. Marching cubes: A high resolution 3d surface construction algorithm. In *Seminal graphics: pioneering efforts that shaped the field*, pages 347–353. 1998. 4

[31] Camillo Lugaresi, Jiuqiang Tang, Hadon Nash, Chris McClanahan, Esha Uboweja, Michael Hays, Fan Zhang, Chuo-Ling Chang, Ming Yong, Juhyun Lee, Wan-Teh Chang, Wei Hua, Manfred Georg, and Matthias Grundmann. Mediapipe: A framework for perceiving and processing reality, 2019. 2, 6, 8

[32] Arpit Mittal, Andrew Zisserman, and Philip HS Torr. Hand detection using multiple proposals. In *BMVC*, page 5. Citeseer, 2011. 1

[33] Gyeongsik Moon, Shoou-I Yu, He Wen, Takaaki Shiratori, and Kyoung Mu Lee. Interhand2.6m: A dataset and baseline for 3d interacting hand pose estimation from a single rgb image. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XX 16*, pages 548–564. Springer, 2020. 8

[34] Franziska Mueller, Micah Davis, Florian Bernard, Oleksandr Sotnychenko, Mickeal Verschoor, Miguel A Otaduy, Dan Casas, and Christian Theobalt. Real-time pose and shape reconstruction of two interacting hands with a single depth camera. *ACM Transactions on Graphics (ToG)*, 38(4):1–13, 2019. 1

[35] Van Nguyen Nguyen, Thibault Groueix, Georgy Ponimatkin, Vincent Lepetit, and Tomas Hodan. Cnos: A strong baseline for cad-based novel object segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2134–2140, 2023. 8, 2

[36] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. 8

[37] Georgios Pavlakos, Dandan Shan, Ilija Radosavovic, Angjoo Kanazawa, David Fouhey, and Jitendra Malik. Reconstructing hands in 3D with transformers. In *CVPR*, 2024. 1, 7, 8, 2, 4

[38] 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. 2, 3, 4, 6

[39] 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), 2017. 5

[40] Dandan Shan, Jiaqi Geng, Michelle Shu, and David F Fouhey. Understanding human hands in contact at internet scale. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9869–9878, 2020. 1

[41] Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. Loftr: Detector-free local feature matching with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8922–8931, 2021. 4

[42] Anilkumar Swamy, Vincent Leroy, Philippe Weinzaepfel, Fabien Baradel, Salma Galaaoui, Romain Brégier, Matthieu Armando, Jean-Sebastien Franco, and Grégory Rogez. Showme: Benchmarking object-agnostic hand-object 3d reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1935–1944, 2023. 2, 3

[43] Omid Taheri, Nima Ghorbani, Michael J Black, and Dimitrios Tzionas. Grab: A dataset of whole-body human grasping of objects. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IV 16*, pages 581–600. Springer, 2020. 3

[44] Zhigang Tu, Zhisheng Huang, Yujin Chen, Di Kang, Linchao Bao, Bisheng Yang, and Junsong Yuan. Consistent 3d hand

reconstruction in video via self-supervised learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023. 1

[45] Bowen Wen and Kostas Bekris. Bundletrack: 6d pose tracking for novel objects without instance or category-level 3d models. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 8067–8074. IEEE, 2021. 2

[46] Bowen Wen, Jonathan Tremblay, Valts Blukis, Stephen Tyree, Thomas Muller, Alex Evans, Dieter Fox, Jan Kautz, and Stan Birchfield. Bundlesdf: Neural 6-dof tracking and 3d reconstruction of unknown objects. *CVPR*, 2023. 1, 2, 3, 8

[47] Bowen Wen, Wei Yang, Jan Kautz, and Stan Birchfield. Foundationpose: Unified 6d pose estimation and tracking of novel objects. *arXiv preprint arXiv:2312.08344*, 2023. 1, 2, 4, 7, 8, 6

[48] Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. *arXiv preprint arXiv:1711.00199*, 2017. 1, 8

[49] Fu Xiong, Boshen Zhang, Yang Xiao, Zhiguo Cao, Taidong Yu, Joey Tianyi Zhou, and Junsong Yuan. A2j: Anchor-to-joint regression network for 3d articulated pose estimation from a single depth image. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 793–802, 2019. 7

[50] Lixin Yang, Kailin Li, Xinyu Zhan, Fei Wu, Anran Xu, Liu Liu, and Cewu Lu. Oakink: A large-scale knowledge repository for understanding hand-object interaction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20953–20962, 2022. 1, 2, 3

[51] Sid Yingze Bao, Manmohan Chandraker, Yuanqing Lin, and Silvio Savarese. Dense object reconstruction with semantic priors. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1264–1271, 2013. 1

[52] Xinyu Zhan, Lixin Yang, Yifei Zhao, Kangrui Mao, Hanlin Xu, Zenan Lin, Kailin Li, and Cewu Lu. Oakink2: A dataset of bimanual hands-object manipulation in complex task completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 445–456, 2024. 2, 3

[53] Fan Zhang, Valentin Bazarevsky, Andrey Vakunov, Andrei Tkachenka, George Sung, Chuo-Ling Chang, and Matthias Grundmann. Mediapipe hands: On-device real-time hand tracking. *arXiv preprint arXiv:2006.10214*, 2020. 1

[54] Yian Zhao, Wenyu Lv, Shangliang Xu, Jinman Wei, Guanzhong Wang, Qingqing Dang, Yi Liu, and Jie Chen. Detrs beat yolos on real-time object detection, 2023. 8

HO-Cap: A Capture System and Dataset for 3D Reconstruction and Pose Tracking of Hand-Object Interaction

Supplementary Material

8. Details on Data Collection

8.1. MANO Shape Calibration

We calibrated each subject's hand shape using a three-step process, assuming both hands share the same shape. First, we performed initial pose estimation and 3D point collection. Participants were instructed to align their right hand with a rendered neutral MANO hand mesh displayed in front of a camera (Fig. 6). During this process, 3D points around the mesh were dynamically collected and filtered based on their distance to the mesh surface, retaining only those within a predefined threshold. The hand pose was then optimized by fitting the 3D points to the signed distance field (SDF) of the MANO model, keeping the shape parameter β fixed at zero to focus solely on pose refinement. The optimized pose and filtered 3D points were saved for the next step. Second, we iteratively optimized both the hand shape β and pose θ using the saved pose and points. This involved alternately fixing one parameter while optimizing the other, gradually improving alignment between the hand mesh and the collected 3D point data. Finally, the optimized hand shape could be further refined by using sequences collected in the dataset. Once finalized, the calibrated hand shape was fixed throughout the following optimization process for hand pose estimation.

Figure 6. Visualization of the rendered MANO hand mesh used as a reference for participants during calibration.

8.2. Synchronization of the Camera Streams

To facilitate efficient control and data collection, all cameras are integrated into a ROS server for seamless acquisition. The 8 static RealSense cameras are wired to the server, where they publish synchronized RGB and depth images. The HoloLens, connected via TCP/IP, operates in Research Mode to access and publish RGB camera streams and head tracking data. All image frames and pose data are recorded into a single ROS bag file, with synchronization performed based on timestamps to ensure temporal alignment.

9. Properties of HO-Cap

9.1. Annotation Details

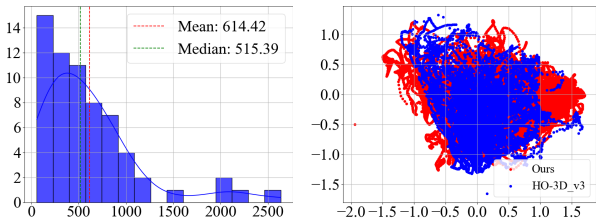
The HO-Cap dataset provides MANO-based 3D hand pose and 6D object pose annotations, optimized within a global world frame using 8 RealSense cameras. Camera intrinsics and extrinsics for all views are included, enabling the transformation of world-frame annotations into camera-frame 3D poses. The 2D hand joint keypoints are obtained by projecting the 3D hand joints onto the image plane. Additionally, segmentation masks offer pixel-wise annotations for objects and hands, facilitating precise scene understanding. The First-Person View (FPV) data from the HoloLens headset provides egocentric perspectives crucial for human perception research and assistive AR applications.

9.2. Visualization and Scene Simulation

The HO-Cap dataset provides textured 3D meshes for 64 objects (Fig. 8), compatible with physics simulators, enabling realistic scene reconstruction and interaction modeling. For sequence data visualization and scene replay in Isaac Sim, please refer to the supplementary video.

9.3. Shape and Pose Diversity

As illustrated in Fig. 7, our dataset (1) encompasses a diverse range of hand-held object shapes and sizes. and (2) offers a broader variety of hand poses compared to HO-3D [18], capturing more natural and dynamic interactions.



(a) (b)

Figure 7. Distribution of (a) object volume (mm^3), (b) hand pose with comparison to HO-3D (first two MANO PCA coefficients).

10. Comparison to Similar Datasets

The most related datasets to HO-Cap are DexYCB[7], HO-3D[18], and H2O [24]. **DexYCB** focuses on 20 objects from the YCB benchmark, primarily capturing single-hand grasping and handover interactions using 8 camera views. In contrast, HO-Cap provides 64 unique objects with textured 3D meshes, offering greater object diversity. Additionally, HO-Cap includes both single-hand and bimanual interactions and incorporates egocentric (FPV) data, enhancing its applicability for AR/VR and human-centered AI research. **HO-3D** features grasping-centric hand poses with 10 objects, captured from 1–5 views per sequence, and relies on marker-based motion capture for annotations. In contrast, HO-Cap supports a significantly wider range of hand poses and grasps across 64 objects and employs a semi-automatic optimization-based annotation pipeline that does not require markers. **H2O** is one of the few datasets that support bimanual hand-object interactions while providing 6D object poses and 3D hand poses. However, it is limited to 8 objects. In comparison, HO-Cap extends this to 64 objects, enabling a wider range of bimanual grasping scenarios and more diverse hand-object interactions.

11. Experimental Details

We benchmarked three tasks using our dataset: hand pose estimation, object detection, and novel object pose estimation. To enhance evaluation efficiency, all methods were tested on sampled keyframes from the dataset.

For hand pose estimation, the evaluation frames were subsampled at 10 frames per second (FPS) across all eight RealSense cameras, resulting in 189,435 frames. This subset is sufficiently representative to capture the diversity of hand poses in the dataset. When computing the mean per joint position error (MPJPE), predictions and ground truths are aligned by replacing the root (wrist) location with the ground truth, eliminating translational ambiguity.

For 6D object pose estimation, due to significant occlusions at the two lowest camera angles and the egocentric perspectives dominated by hand views, we selected 11,758 frames from the remaining six viewpoints for evaluation. These chosen frames encompass diverse poses of objects. Ground truth bounding boxes were provided as input for the pose estimation methods.

For 2D object detection, evaluation frames were sampled from the RealSense camera feeds. For novel object detection, a random sampling strategy was employed, yielding 7,293 frames. Table 7 provides the statistics of the evaluation setup for seen object detection. Full frames were sampled across all eight RealSense cameras with a subsampling factor of 10 and divided into train/val/test splits. For each split, we list the number of subjects ("#sub"), objects ("#obj"), views ("#view"), sequences ("#seq"), image samples ("#image"), and the object annotations ("#obj anno").

Table 7. Statistics of the evaluation setup for seen object detection.

splits	#sub	#obj	#view	#seq	#image	#obj anno
train	9	64	8	48	36,295	177,564
val	9	64	8	48	9,073	4,4535
test	8	64	8	16	1,2336	62,598

12. Qualitative: 3D Hand Pose Estimation

Fig. 9 shows qualitative results of 3D hand pose estimation using HaMeR [37] and A2J-Transformer [21] methods. As shown in the figure, the results include both 2D hand joints and 3D hand joints. HaMeR demonstrates superior accuracy in hand pose estimation compared to A2J-Transformer. Due to the limitations of the A2J-Transformer training process, it lacks consideration for the interaction between the hand and surrounding objects. The results show that as the interaction area between the hand and the object increases or when the hand is occluded by the object, the performance of A2J-Transformer deteriorates. Conversely, HaMeR exhibits a robust adaptability to these challenging conditions by training a ViT model with large-scale training images.

13. Qualitative: Novel Object Detection

Fig. 10 shows qualitative results of 2D novel object detections with GroundingDINO [28] and CNOS [35]. For GroundingDINO, we used combined object product names as prompt captions. For CNOS, feature templates were created for each object by rendering textured 3D models from multiple viewpoints. Both methods exhibit a significant number of false positives when tested on our dataset. Given that each scene contains only four objects, Fig. 10 highlights the top four detected bounding boxes with the highest detection scores.

14. Qualitative: 6D Object Pose Estimation

Fig. 11 shows qualitative results of 6D object pose estimation. The object models, rendered using the estimated poses, are overlaid onto a darkened input image for visualization. As shown, FoundationPose[47] generates more accurate 6D pose predictions compared to MegaPose [25] on novel objects.

15. Quantitative: 6D Object Pose Estimation

For 6D object pose estimation, we include more detailed results for the evaluation metrics—namely, Average Distance (ADD) and Symmetric Average Distance (ADD-S)—on a per-object basis in Table 8. The relationships of objects and their IDs can be found in Fig. 12. We observe that FoundationPose [47] significantly surpasses MegaPose [25] in handling novel objects, demonstrating a substantial improvement in accuracy.

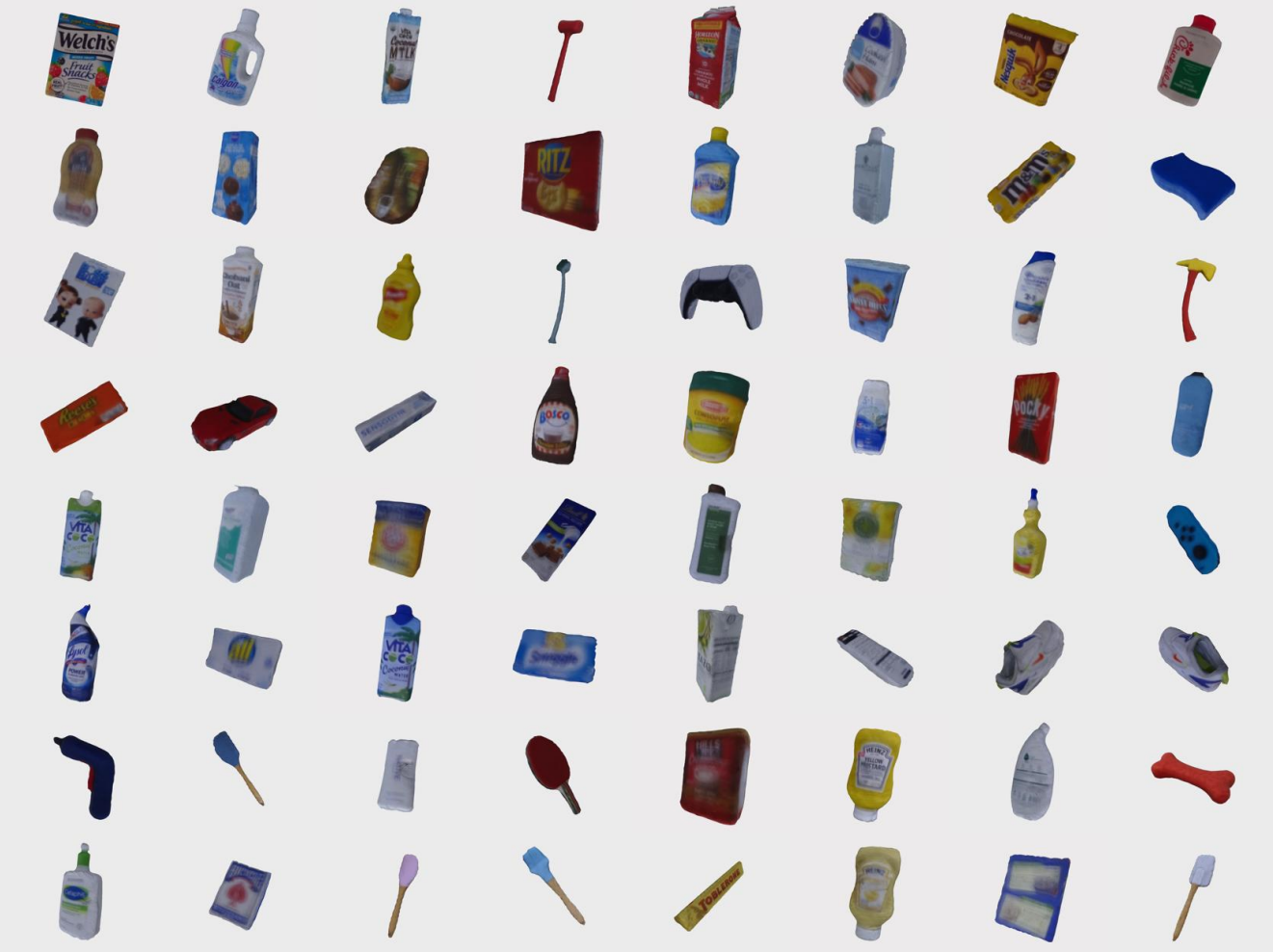


Figure 8. Reconstructed textured meshes of the 64 objects in HO-Cap

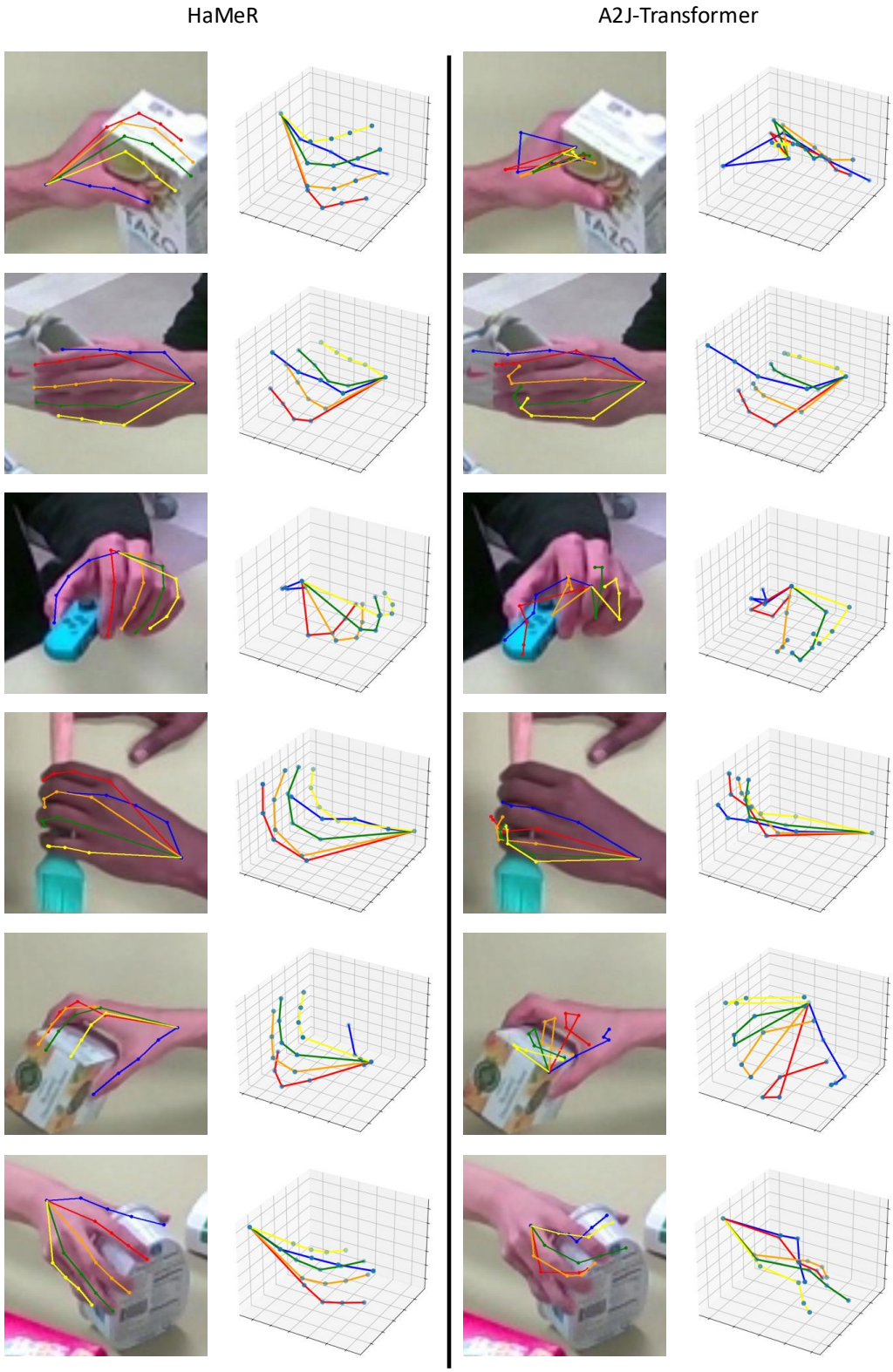


Figure 9. Qualitative results of the predicted 3D hand pose using HaMeR [37] (left two columns) and A2J-Transformer [21] (right two columns).

Figure 10. Qualitative results for novel object detection

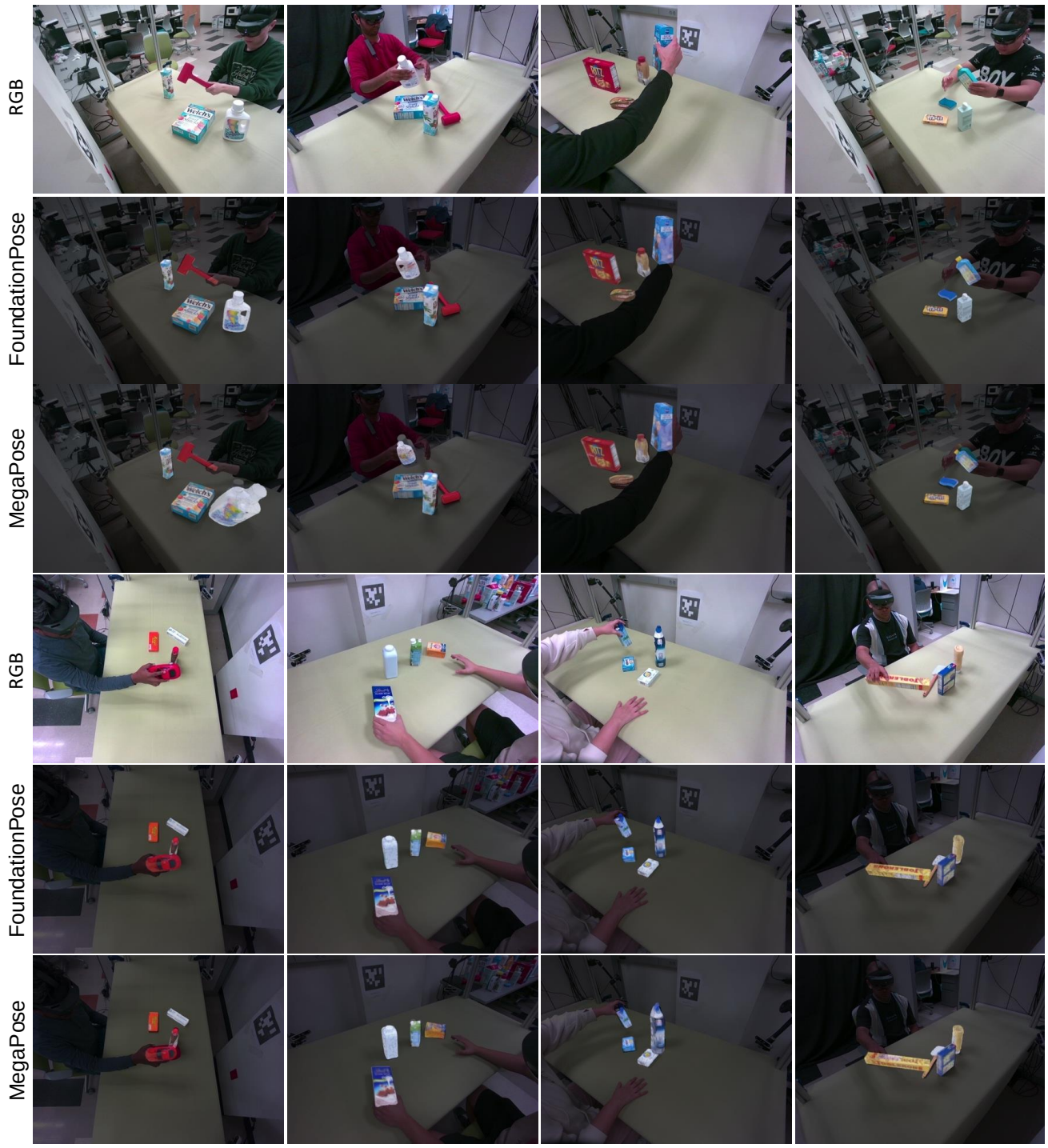


Figure 11. Qualitative results of 6D object pose estimation. (Top to down: Input RGB frame, FoundationPose [47], MegaPose [25])

Object ID	FoundtionPose [47]		MegaPose [25]		Object ID	FoundtionPose [47]		MegaPose [25]	
	ADD	ADD-S	ADD	ADD-S		ADD	ADD-S	ADD	ADD-S
G01_1	93.66	96.00	88.90	93.59	G11_1	89.40	95.74	66.49	82.61
G01_2	93.55	96.18	80.08	88.33	G11_2	89.43	95.74	66.20	82.51
G01_3	92.98	95.96	71.38	81.96	G11_3	89.39	95.67	65.92	82.62
G01_4	88.39	95.58	69.44	82.26	G11_4	89.42	95.67	65.45	82.21
G02_1	89.60	95.72	69.37	82.89	G15_1	89.43	95.67	65.37	82.18
G02_2	90.19	95.79	71.22	84.50	G15_2	89.41	95.67	65.33	82.18
G02_3	90.82	95.93	71.72	85.56	G15_3	89.42	95.67	65.31	82.18
G02_4	91.00	95.83	70.88	85.11	G15_4	89.53	95.68	65.28	82.10
G04_1	91.10	95.85	69.84	84.39	G16_1	89.57	95.69	65.56	82.14
G04_2	91.25	95.88	69.17	84.33	G16_2	89.65	95.70	65.57	82.18
G04_3	91.29	95.87	69.33	84.37	G16_3	89.67	95.69	65.42	81.97
G04_4	91.36	95.87	70.08	84.73	G16_4	89.72	95.70	65.57	82.03
G05_1	90.54	95.90	69.25	84.79	G18_1	89.67	95.61	65.41	82.05
G05_2	90.86	95.92	69.14	84.76	G18_2	89.70	95.61	65.48	82.04
G05_3	90.82	95.94	69.21	84.80	G18_3	89.72	95.61	65.68	82.13
G05_4	86.23	95.90	66.57	85.10	G18_4	89.77	95.59	65.92	82.25
G06_1	87.28	95.94	67.48	85.33	G19_1	89.75	95.57	66.25	82.32
G06_2	88.13	95.99	66.02	84.59	G19_2	88.11	95.59	65.90	81.72
G06_3	88.42	96.01	66.97	84.84	G19_3	87.67	95.56	64.78	81.61
G06_4	88.41	95.83	65.75	82.95	G19_4	87.70	95.57	65.09	81.75
G07_1	88.52	95.81	65.58	82.50	G20_1	87.76	95.57	64.57	81.79
G07_2	88.57	95.81	65.65	82.52	G20_2	87.83	95.58	64.64	81.88
G07_3	88.72	95.84	66.04	82.66	G20_3	87.59	95.58	64.89	82.01
G07_4	88.78	95.81	66.14	82.50	G20_4	87.64	95.59	64.30	81.84
G09_1	88.96	95.71	67.04	82.84	G21_1	87.73	95.60	64.40	81.99
G09_2	89.20	95.68	68.11	83.26	G21_2	87.79	95.59	64.68	82.15
G09_3	89.51	95.73	67.68	83.27	G21_3	87.68	95.56	64.29	81.63
G09_4	89.72	95.75	67.51	83.34	G21_4	87.65	95.56	63.87	81.13
G10_1	89.82	95.77	67.64	83.44	G22_1	87.63	95.57	64.01	81.23
G10_2	89.81	95.78	67.58	83.48	G22_2	87.69	95.57	63.68	81.13
G10_3	89.88	95.78	67.72	83.51	G22_3	87.73	95.58	63.64	81.12
G10_4	89.52	95.76	66.92	83.03	G22_4	89.33	95.74	63.53	80.97

Table 8. 6D object pose estimation results of representative approaches in ADD and ADD-S.

Figure 12. Objects with their IDs in our dataset.