HOI4D: A 4D Egocentric Dataset for Category-Level Human-Object Interaction

Yunze Liu\textsuperscript{*1,3}, Yun Liu\textsuperscript{*1}, Che Jiang\textsuperscript{1}, Kangbo Lyu\textsuperscript{1}, Weikang Wan\textsuperscript{2}, Hao Shen\textsuperscript{2}, Boqiang Liang\textsuperscript{2}, Zhoujie Fu\textsuperscript{1}, He Wang\textsuperscript{2}, Li Yi\textsuperscript{†1,3}

\textsuperscript{1} Tsinghua University, \textsuperscript{2} Peking University, \textsuperscript{3} Shanghai Qi Zhi Institute

https://hoi4d.github.io

Abstract

We present HOI4D, a large-scale 4D egocentric dataset with rich annotations, to catalyze the research of category-level human-object interaction. HOI4D consists of 2.4M RGB-D egocentric video frames over 4000 sequences collected by 9 participants interacting with 800 different object instances from 16 categories over 610 different indoor rooms. Frame-wise annotations for panoptic segmentation, motion segmentation, 3D hand pose, category-level object pose and hand action have also been provided, together with reconstructed object meshes and scene point clouds. With HOI4D, we establish three benchmarking tasks to promote category-level HOI from 4D visual signals including semantic segmentation of 4D dynamic point cloud sequences, category-level object pose tracking, and egocentric action segmentation with diverse interaction targets. In-depth analysis shows HOI4D poses great challenges to existing methods and produces huge research opportunities.

1. Introduction

Tremendous progress [9, 14, 44, 47] has been made for naming objects and activities in images, videos or 3D point clouds over the last decade facilitated by significant dataset and benchmark efforts. However, these perception outcomes fail to satisfy the needs of more and more critical applications such as human-assistant robots and augmented reality where the perception of interactions from 4D egocentric sensory inputs (e.g., temporal streams of colored point clouds) is required. It becomes highly desirable for a computer vision system to build up a detailed understanding of human-object interaction from an egocentric point of view. Such understanding should unify semantic understanding of 4D dynamic scenes, the 3D pose of human hands under object occlusion, the 3D pose and functionality of novel objects of interaction interest, as well as the action and intention of humans, which pose new challenges for today’s computer vision systems.

To help tackle these challenges, large-scale and annotation-rich 4D egocentric HOI datasets as well as the corresponding benchmark suites are strongly needed. Recently some efforts [18, 19, 22] have been made to fulfill such needs. However, most of these works focus on what we called instance-level human-object interaction where the objects being interacted with all come from a very small pool of instances whose exact CAD models and sizes are known beforehand. This impedes their application to perceiving human interaction with the vast diversity of objects in our daily life. Moreover, these works tend to ignore articulated objects while only focusing on rigid objects with which the interaction patterns are relatively simpler. These limitations partially come from the challenging and cumbersome nature of jointly capturing hands, objects and real scenes in an egocentric manner. Curating synthetic datasets [20] might be an alternative. Nonetheless simulating natural human motion and functional grasping for generic objects are still open research problems, making it
We present the first dataset, HOI4D, for 4D egocentric category-level human-object interaction as depicted in Figure 1. We draw inspirations from recent category-level object pose estimation and pose tracking works [24,42,43] and aim to propel 4D HOI perception to a new era to handle category-level object variations in cluttered scenes. We collect a richly annotated 4D egocentric dataset, HOI4D, to depict humans interacting with various objects while executing different tasks in indoor environments. HOI4D consists of 2.4M RGB-D egocentric video frames over 4000 sequences of 9 participants interacting with 800 object instances. These object instances are evenly split into 16 categories including both rigid and articulated objects. Also instead of using a lab setting like most previous works, the camera wearers execute tasks revealing the functionality of each category without wearing any markers in 610 different indoor scenes. HOI4D is associated with reconstructed scene point clouds and object meshes for all sequences. HOI4D provides annotations for frame-wise panoptic segmentation, motion segmentation, 3D hand pose, rigid and articulated object pose, and action segmentation, delivering unprecedented levels of detail for human-object interaction at the category level.

The rich annotations in HOI4D also enable benchmarking on a series of category-level HOI tasks. In this paper, we focus on three tasks in particular: semantic segmentation of 4D dynamic point cloud sequences, category-level object pose tracking for hand-object interaction and egocentric hand action recognition with diverse interaction targets. We provide an in-depth analysis of existing approaches to these tasks. Our experiments suggest that HOI4D has posed great challenges to today’s computer vision algorithms. For category-level object and part pose tracking task, most of the previous datasets use synthetic data under simple scenes without hand occlusion. With the help of the proposed HOI4D dataset, researchers can now work on this more challenging task with real-world data. Due to a lack of annotated indoor datasets, semantic segmentation of 4D point clouds has been studied mainly for autonomous driving applications. HOI4D introduces more challenges such as heavy occlusion, fast ego-motion, and very different sensor noise patterns. Fine-grained action segmentation of video can help AI better comprehend interaction, but we found that existing coarse-grained methods cannot directly process fine-grained data well.

In summary, our contributions can be listed below:

- **We present a data collection and annotation pipeline combining human annotations with automatic algorithms, effectively scaling up our dataset.**
- **We benchmark on three category-level HOI tasks covering 4D dynamics scene understanding, category-level object pose tracking and hand action segmentation.** We provide a thorough analysis of existing methods and point out new challenges HOI4D has posed.

## 2. Related Work

### 2.1. Egocentric Human-Object Interaction Datasets

Understanding human-object interaction has long been a pursuit for computer vision researchers and many previous efforts have been focusing on constructing third-person view datasets [5, 8, 38, 40]. Recently we observe a surge of interest in perceiving human-object interaction from an egocentric view. A lot of these datasets focus on recognizing daily activities [4, 12, 17, 26, 33] and provide mostly 2D features omitting 3D annotations such as 3D hand poses and object poses, which are crucial for a comprehensive understanding of the underlying interactions.

Annotating 3D hand poses and object poses together is not an easy task though due to reciprocal occlusions. Some existing works leverage magnetic sensors or mocap markers to track 3D hand poses and object poses [5, 18, 48]. However, the attached markers might hinder natural hand motion and bias the appearance of hands and objects. Other works leverage carefully calibrated multi-camera systems [8, 38] or optimization algorithms [19] to ease the difficulty but are usually restricted to a third-person point of view. Most relevant to ours are a recent egocentric HOI dataset named H2O [22]. They collect egocentric RGB-D videos with annotations for 3D hand poses, instance-level object poses, and action labels. However, H2O is restricted to instance-level human-object interaction covering interactions with only 8 object instances. In addition, previous works only focus on rigid objects while we also consider articulated objects where richer interactions could happen.

As shown in Table 1, we are the first to present a large-scale 4D egocentric dataset for category-level HOI covering both rigid and articulated object categories with an unprecedented level of richness in annotations.

### 2.2. 4D Dynamic Scene Understanding

4D Dynamic Scene Understanding is important since it enables AI to understand the real world we live in. Existing methods are mainly based on outdoor datasets such as Synthia 4D [36] and SemanticKITTI [3]. MinkowskiNet [11] proposes to use 4D Spatio-Temporal ConvNets to extract 4D features. MeteorNet [29] takes point cloud
Table 1. Comparison of existing HOI Datasets.

<table>
<thead>
<tr>
<th>dataset</th>
<th>4D real</th>
<th>markerless 3D hand</th>
<th>6D obj</th>
<th>ego</th>
<th>#frames</th>
<th>#obj</th>
<th>#seq</th>
<th>dynamic grasp</th>
<th>action label</th>
<th>seg</th>
<th>category-level</th>
<th>articulated</th>
<th>functional intent</th>
</tr>
</thead>
<tbody>
<tr>
<td>GTEA+GAZE</td>
<td>✓✓</td>
<td>✓✓</td>
<td>×</td>
<td>✓</td>
<td>778K</td>
<td>-</td>
<td>37</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>EPIC-KITCHEN [12]</td>
<td>✓✓</td>
<td>✓✓</td>
<td>×</td>
<td>✓</td>
<td>20M</td>
<td>-</td>
<td>700</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>FPHA [18]</td>
<td>✓✓</td>
<td>✓✓</td>
<td>×</td>
<td>✓</td>
<td>105K</td>
<td>4</td>
<td>1,175</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>ObMan [20]</td>
<td>✓✓</td>
<td>×✓</td>
<td>×</td>
<td>✓</td>
<td>154K</td>
<td>3K</td>
<td>-</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>FreeHAND [50]</td>
<td>✓✓</td>
<td>×✓</td>
<td>×</td>
<td>✓</td>
<td>37K</td>
<td>27</td>
<td>-</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>ContactPose [5]</td>
<td>✓✓</td>
<td>×✓</td>
<td>×</td>
<td>✓</td>
<td>2,991K</td>
<td>25</td>
<td>2,303</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>H3-3D [19]</td>
<td>✓✓</td>
<td>✓✓</td>
<td>×</td>
<td>✓</td>
<td>78K</td>
<td>10</td>
<td>27</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>DexYCB [8]</td>
<td>✓✓</td>
<td>✓✓</td>
<td>×</td>
<td>✓</td>
<td>582K</td>
<td>20</td>
<td>1,000</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>H2O [22]</td>
<td>✓✓</td>
<td>✓✓</td>
<td>×</td>
<td>✓</td>
<td>571K</td>
<td>8</td>
<td>-</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Ours</td>
<td>✓✓</td>
<td>✓✓</td>
<td>✓✓</td>
<td>✓</td>
<td>2.4M</td>
<td>800</td>
<td>4K</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

sequences as inputs and aggregates information in the temporal and spatial neighborhoods. SpSequenceNet [37] manipulates the 4D point cloud data in the 3D cube style to reduce spatial information loss. PSTNet [15] proposes a point spatio-temporal convolution to achieve informative representations of point cloud sequences. P4transformer [13] is a novel Point 4D Transformer to avoid point tracking. 4D-Net [32] proposes a novel learning technique for fusing information in 4D from multi-modalities. In the indoor interactive scene, the scale of the object is smaller, the movement is more diverse, and there is even the deformation of the object, which presents new challenges to existing methods.

### 2.3. Category-Level Object Pose Estimation and Pose Tracking

To define the poses of novel objects, NOCS [42] proposes a Normalized Object Coordinate Space as a category-specific canonical reference frame. Every input object pixel is projected into the category-level canonical 3D space. ANCSH [25] extended the concept of NOCS to articulated objects and proposes Normalized Part Coordinate Space (NPCS), which is a part-level canonical reference frame. In terms of pose tracking, 6-PACK [41] tracks a small set of keypoints in RGB-D videos and estimates object pose by accumulating relative pose changes over time. CAPTRA [44] build an end-to-end differentiable pipeline for accurate and fast pose tracking for both rigid and articulated objects. BundleTrack [43] proposes a novel integration method and a memory-augmented pose graph optimization for low-drift accurate 6D object pose tracking. However, existing methods do not consider the pose tracking for the hand and the object jointly, which is very important in interactive scenes. Replacing the third angle of view with the ego-centric view, the problem of object occlusion becomes more serious, which also makes this task more difficult. In addition, existing datasets such as NOCS [42] are synthetic datasets, so the domain gap between real-world data and synthetic data also poses challenges to existing algorithms. With the help of the proposed HOI4D dataset, researchers can now work on the above more challenging tasks with little overhead.

### 3. Constructing HOI4D

#### 3.1. Hardware Setup and Data Collection

![Data capturing system.](image)

To construct HOI4D, we build up a simple head-mounted data capturing suite consisting of a bicycle helmet, a Kinect v2 RGB-D sensor, and an Intel RealSense D455 RGB-D sensor.

To define the poses of novel objects, NOCS [42] proposes a Normalized Object Coordinate Space as a category-specific canonical reference frame. Every input object pixel is projected into the category-level canonical 3D space. ANCSH [25] extended the concept of NOCS to articulated objects and proposes Normalized Part Coordinate Space (NPCS), which is a part-level canonical reference frame. In terms of pose tracking, 6-PACK [41] tracks a small set of keypoints in RGB-D videos and estimates object pose by accumulating relative pose changes over time. CAPTRA [44] build an end-to-end differentiable pipeline for accurate and fast pose tracking for both rigid and articulated objects. BundleTrack [43] proposes a novel integration method and a memory-augmented pose graph optimization for low-drift accurate 6D object pose tracking. However, existing methods do not consider the pose tracking for the hand and the object jointly, which is very important in interactive scenes. Replacing the third angle of view with the ego-centric view, the problem of object occlusion becomes more serious, which also makes this task more difficult. In addition, existing datasets such as NOCS [42] are synthetic datasets, so the domain gap between real-world data and synthetic data also poses challenges to existing algorithms. With the help of the proposed HOI4D dataset, researchers can now work on the above more challenging tasks with little overhead.
3.2. Data Annotation Pipeline

HOI4D consists of rich labels covering different aspects of category-level human-object interaction and collecting these annotations is not a trivial task. We show our data annotation pipeline in Figure 3. Given a dynamic RGB-D sequence, we first split moving content and static content to ease panoptic labeling by annotating framewise 2D motion segmentation. Then we mask out the moving content and reconstruct a 3D static scene via a SLAM algorithm [10, 49]. This allows us to efficiently annotate all the static content in the whole sequence. We manually annotate the reconstructed scene to obtain 3D static scene panoptic segmentation. Finally, the 2D motion segmentation and the 3D static scene panoptic segmentation are merged, resulting in the 4D dynamic scene panoptic segmentation. We explain the detailed process in Section 3.3. To obtain 3D hand pose labels, we first annotate a set of hand keypoints on RGB-D frames and then leverage an optimization module to recover the underlying 3D hand as described in Section 3.4. To obtain category-level object poses, we manually fit amodal oriented bounding boxes to objects or object parts in RGB-D frames and make sure the pose definitions are consistent across a certain object category. We further optimize the object poses by leveraging the object mesh reconstructed from a multi-view scanning process. The category-level object and part pose annotation process is described in Section 3.5. Moreover, we also describe the process for action annotation in Section 3.6.

3.3. 4D Panoptic Labeling

Our 4D panoptic labeling process is mainly divided into two parts named 2D motion segmentation labeling and 3D static scene segmentation labeling. In the process of 2D motion segmentation labeling, given an original RGB video, the annotators sample 10% of frames evenly from a video and then manually annotate segmentation masks of the objects that have moved in the video. While manually annotating the entire video frames is labor-consuming, we instead use the off-the-shelf 2D mask propagation tool [9] to propagate the existed manually annotated masks to other 90% frames in the same video. The annotators interact with the propagation tool to refine all of the segmentation masks until they are accurate. In the process of static scene segmentation labeling, given an original RGB-D video with 2D motion segmentation masks, we remove all of the masked objects from single frame point clouds and then reconstruct the static point cloud [10, 49] of the indoor scene containing only objects whose position hasn’t been changed. We label all the object instances and background stuff in the static scene and project the results back to each frame so that the segmentation masks of static objects are acquired. The final 4D panoptic segmentation labels are obtained by merging the motion segmentation and static segmentation masks.

3.4. Hand Pose Annotation

The entire hand pose annotation process includes four stages: annotation, initialization, propagation and refinement.

Annotators uniformly annotate 20% frames in each video. For annotation, marker-based annotation methods are unfeasible since we need realistic appearances of hands. Alternatively, we manually annotate 2D positions of a set of hand keypoints. We adopt the pre-defined 21 keypoints of hand joints that are widely used in previous works [19, 22, 35]. For each annotated frame, annotators provide the 2D position of 11 keypoints: the wrist, 5 fingertips and the second knuckles counted from tips. The reasonable positions of occluded keypoints are also estimated.

We represent hand pose by the MANO parametric hand model [35]. The shape parameters $\beta \in \mathbb{R}^{10}$ are fixed based upon the real hand information from the data capturer and we optimize hand pose $\theta_h = \{\theta_n, R, t\} \in \mathbb{R}^{51}$ that consists of the pose coefficient $\theta_m \in \mathbb{R}^{15}$ (3 DoF for each of the 15 hand joints) plus global rotation $R \in SO(3)$ and translation $t \in \mathbb{R}^3$. In the initialization stage, we estimate 3D hand poses from 2D annotations by minimizing loss functions for every annotated frame. The loss function is defined as:

$$\hat{\theta}_h = \arg \min_{\theta_h} (\lambda_j L_j + \lambda_{2D} L_{2D} + \lambda_d L_d + \lambda_{pc} L_{pc} + \lambda_m L_m)$$

(1)

where $L_j, L_{2D}, L_d, L_{pc}$ and $L_m$ represents joint angle loss, 2D joint loss, depth loss, point cloud loss and mask loss respectively. $\lambda_j, \lambda_{2D}, \lambda_d, \lambda_{pc}$ and $\lambda_m$ are balancing parameters. Details about loss terms’ definitions are given in the supplementary material. Considering the temporal consistency of the video, each frame is initialized by the hand pose of its previous frame to accelerate convergence. We propagate the hand poses $\theta_h$ of the annotated frames across the whole sequence by linear interpolation to obtain the coarse hand pose of each frame in the video.

In the refinement stage, we further optimize $\hat{\theta}_h$ to get precise poses for all frames. The loss function is defined as:

$$\tilde{\theta}_h = \arg \min_{\theta_h} (\lambda_j L_j + \lambda_d L_d + \lambda_{pc} L_{pc} + \lambda_m L_m$$

$$+ \lambda_{Contact} L_{Contact} + \lambda_{tc} L_{tc})$$

(2)

where $L_{Contact}$ and $L_{tc}$ represents contact loss and temporal consistency loss respectively. More details about loss terms’ definition are given in the supplementary material.

To balance the efficiency and limited computational resource, we select 6-11 consecutive frames as an optimization batch containing 2-3 annotated frames. We manually detect failure frames since the optimization process may fail due to ambiguous hand poses or a bad initialization. The hand poses of failure frames are manually rectified.
Figure 3. Overview of annotation pipeline. **Red branch:** Given a dynamic RGB-D sequence, we first annotate frame-wise 2D motion segmentation. Then we mask out the moving content and reconstruct a 3D static scene. We manually annotate the reconstructed scene to obtain 3D static scene panoptic segmentation. Finally, the 2D motion segmentation and the 3D static scene panoptic segmentation are merged, resulting in the 4D dynamic scene panoptic segmentation. **Blue branch:** To obtain 3D hand pose labels, we first annotate a set of hand keypoints on RGB-D frames and then leverage an optimization module to recover the underlying 3D hand. For category-level object poses, we manually fit amodal oriented bounding boxes to objects or object parts in RGB-D frames and further optimize it by leveraging the object mesh. **Green branch:** We directly annotated fine-grained action labels on the original video.

### 3.5. Category-Level Pose Annotation

The process of annotating category-level poses for both rigid and articulated objects contains three stages: object measurement and annotation, model scanning and reconstruction and label propagation and pose optimization.

**Object measurement and annotation.** To balance the annotation quality and the labor intensity, we manually annotate tight amodal oriented bounding boxes for objects of interest every ten frames in each video. Specifically, we first measure each object of interest physically and define its coordinate system to get its tight amodal bounding box. Then the annotators will manually rotate and place these boxes to fit objects in the depth point cloud of each frame. Together with the box scales, we finally get 9D poses for rigid objects of interest. For articulated objects, we treat each part as an individual object and annotate them separately.

**Model scanning and reconstruction.** Labeling object poses in 3D is a challenging task for human annotators while not all annotations are fully reliable. In addition, we only manually annotate 1 out of 10 frames so that some label propagation and optimization techniques such as [19] are needed to produce labels for all video frames. Existing pose optimization techniques are mostly designed for instance-level pose annotation which requires corresponding CAD models for objects of interest. Therefore we have scanned all the 800 objects in our dataset for pose optimization purposes. We have covered a variety of object categories with varying sizes, materials, and topologies, making object scanning non-trivial as well. Although a commercial 3D scanner can be used to model objects of small size, modeling objects with large size, especially with complex topology and materials remains challenging. We choose to first manually decorate objects with various stickers that enrich the object texture and hide some of the highly specular areas. We then use off-the-shelf software packages [6, 30] to reconstruct the object mesh from multi-view high-resolution color images. Specifically, we take images from various depression angles to entirely cover the outer surface of an object. We then adopt a series of algorithms provided by the software packages to automatically align the images, reconstruct the object mesh and calibrate model specifications. For articulated objects, we additionally provide part annotations similar to PartNet [31]. Worth mentioning, by providing object meshes, HOI4D could facilitate research in instance-level HOI and also makes it possible to transfer the human interaction trajectories to a simulation environment for applications such as robot imitation learning [34].

**Label propagation.** To propagate manually annotated object poses to intermediate frames, we convert all object poses into a world coordinate system using the camera matrix. Then we linearly interpolate the translation, rotation, and joint angles between annotated object poses.

**Pose optimization.** We leverage multi-modal data, including RGB-D images, the reconstructed object mesh, as well
as the 2D motion segmentation masks, to reduce the error caused by label propagation. We utilize a differentiable renderer SoftRas [28] and auxiliary loss terms in HOnnotate [19] to optimize the pose by gradient descent. The object pose $\theta_o$, which consists of rotation $R \in \text{SO}(3)$, translation $t \in \mathbb{R}^3$, and joint angles $\theta \in \mathbb{R}^\# \text{of joints}$ of articulated objects, should minimize the loss function defined as:

$$\hat{\theta}_o = \arg \min_{\theta_o} (\lambda_{2D} L_{2D} + \lambda_d L_d + \lambda_{cd} L_{cd} + \lambda_{tc} L_{tc}).$$ (3)

where $L_{2D}$ and $L_d$ are computed by SoftRas [28], which penalize the silhouette discrepancy between the rendered object mask and the ground truth mask acquired in 3.3, and the depth residual between the cropped real depth map and the depth map of rendered reconstructed mesh respectively. The auxiliary loss terms are defined similar to HOnnotate [19], where $L_{cd}$ refers to the chamfer distance between the reconstructed object point cloud and the collected object point cloud, and $L_{tc}$ maintains the temporal consistency of the pose trajectory. For articulated objects, we additionally set the joint angle to be restricted within the physical limits of each joint.

3.6. Action annotation

Detecting and temporally locating action segments in long untrimmed 4D videos is a challenging task. For each frame in a video, we annotate its action category to support the study of action segmentation. It is worth noting that we define fine-grained actions in the interactive scene, which is significantly different from the existing datasets. Detailed categories are provided in the supplementary materials.

4. Dataset Statistics

Category overview. Figure 4 shows the object categories contained in our dataset. We select 16 common object categories in our daily life (7 rigid object categories, 9 articulated object categories) to construct our dataset. Each category consists of 50 unique object instances, and each object instance corresponds to a CAD model reconstructed from a set of high-resolution RGB images. It is worth mentioning that these categories are mainly selected from ShapeNet [7] and Sapien Assets [45]. This makes HOI4D well connected with popular synthetic 3D datasets and facilitates studying sim-to-real knowledge transfer. The reconstructed meshes and human hand trajectories can be potentially put into simulation environments to support robot learning as demonstrated in the supplementary materials.

A RealSense D455 and a KinectV2 are used to capture human-object interaction simultaneously, providing opportunities to study knowledge transfer across different depth sensors. Each video is captured at 15fps for 20 seconds. As a result, HOI4D contains 2.4M frames in total.

Diversity of interaction task. To reflect the functionality of different object categories, we define interactive tasks based on object functions as shown in Figure 5. We have defined 54 tasks across all 16 categories. Each object includes a pick and place task and several functionality-based tasks, which can be used to support perceiving object mobility and functionality in interactive scenarios. According to the difficulty of the tasks and the complexity of the scenes, we divide the tasks into two levels: the simple level and the complex level. For simple tasks, the captured sequences contain a subject performing a pick-and-place task over a target object with a relatively clean background regardless of the functionality of the objects. For complex tasks, we randomly pick 10-20 objects from our 800-object pool and place them in a cluttered manner. Tasks with different difficulties naturally support different research directions. Simple tasks better support research regarding pose tracking and robot learning, while complex tasks introduce interesting research problems such as 4D panoptic segmentation.

5. Cross-Dataset Evaluations

We conduct a cross-dataset evaluation to further assess the merit of our HOI4D dataset. We consider three tasks including 3D hand pose estimation, category-level object pose tracking and action segmentation.

Settings. For 3D hand pose estimation, we consider H2O [22] with ego-centric views consistent with HOI4D.
6. Tasks and Benchmarks

In this section, we design three specific tasks on HOI4D: category-level object and part pose tracking, semantic segmentation of 4D point cloud videos, and egocentric hand action segmentation. We follow a ratio of 7:3 to randomly split our 4D sequences into training and test sets and use the annotations from Section 3.5, Section 3.3, and Section 3.6 to support these three tasks respectively. We provide results of baseline methods and in-depth analyses of existing methods and discuss the new challenges that emerged from HOI4D.

### 6.1. Category-Level Object and Part Pose Tracking

Most existing 6D object pose estimation or tracking approaches assume access to an object instance’s 3D model [2, 46]. In the absence of such 3D models, generalization for novel instances becomes very difficult. To alleviate the dependence on CAD models, category-level object and part pose tracking is a promising direction.

In this section, we benchmark the state-of-the-art category-level object and part pose tracking algorithm BundleTrack [43] on HOI4D. BundleTrack [43] is a generic 6D pose tracking framework for new objects that do not rely on an instance or class-level 3D models. The evaluation protocol is the same as in prior work [43, 44]. A perturbed ground-truth object pose is used for initialization. And we also provide an ICP [49] baseline, which leverages the standard point-to-plane ICP algorithm implemented in Open3D [49]. We select 4 rigid object categories and 1 articulated object category for experiments. The following metrics are used: 5°5cm: percentage of estimates with orientation error <5° and translation error <5cm. \(R_{err}\): mean orientation error in degrees. \(T_{err}\): mean translation error in centimeters.

#### Table 3. Category-Level Object Pose Tracking on 4 rigid objects.

<table>
<thead>
<tr>
<th></th>
<th>ICP</th>
<th>BundleTrack</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>toy car</td>
<td>mug</td>
</tr>
<tr>
<td>5°5cm</td>
<td>0.7</td>
<td>9.7</td>
</tr>
<tr>
<td>(R_{err})</td>
<td>88.3</td>
<td>21.0</td>
</tr>
<tr>
<td>(T_{err})</td>
<td>47.6</td>
<td>28.4</td>
</tr>
<tr>
<td></td>
<td>20.1</td>
<td>28.4</td>
</tr>
<tr>
<td></td>
<td>13.9</td>
<td>4.1</td>
</tr>
</tbody>
</table>

Table 3 shows the results of 4 rigid objects and Table 4 shows the results of the laptop category. Taking the bottle category as an example, BundleTrack can achieve an accuracy of 86.5 (5°5cm) on the NOCS [42] dataset where data is not captured during human-object interaction and does not suffer from heavy hand occlusions. Now it only reaches 19.3 on HOI4D. This performance degradation proves category-level pose tracking is indeed very challenging in real-world interactive scenarios, where data suffers a lot from the sensor noise, complex backgrounds, hand occlusions, as well as fast motions. Most previous algorithms are firstly developed using synthetic datasets. It is interesting to see how this previously successful experience could transfer to egocentric human-object interactions. No matter focusing on sim-to-real transfer or directly learning from the real data, HOI4D makes it easy to follow both paths. It is expected that more research will focus on category-level object pose tracking in real-world interactive scenarios.
Recent advances in video action segmentation have accomplished promising achievements in coarse-level segmentation on the Breakfast [21], 50 Salad [39], and GTEA datasets [1]. Fine-grained video action segmentation can help AI better understand interactive actions in interactive tasks. However, few works focus on the fine-grained video action segmentation in interactive scenarios due to the lack of a large-scale fine-grained dataset.

We consider three representative and high-performance methods: MS-TCN [16], MS-TCN++ [23] and Asformer [47]. We use the I3D feature extracted according to Section 5.1 to train the network. We use the videos’ temporal resolution at 15 fps, and the dimension of the I3D feature for each frame is 2048-d. The following three metrics are reported: framewise accuracy (Acc), segmental edit distance, as well as segmental F1 scores at the overlapping thresholds of 10%, 25%, and 50%. Overlapping thresholds are determined by the IoU ratio.

Table 6 shows the results. Unsurprisingly, performance for all three algorithms drops by a large margin from the coarse level to the fine-grained level. Take Asformer [47] as an example, it can only achieve an accuracy of 46.8 on HOI4D but can obtain 85.6 on 50Salads [39], which shows that the existing model does not perceive the most fine-grained actions very well. Fig 6 is an analysis of failure results: Although the prediction is completely wrong, the sequence of actions is correct. From this, we speculate that the current network learns more about the order of actions but lacks the ability to perceive the current action itself. When we use the finest-grained action labels to break the inherent sequence of actions, the performance of the existing method is greatly reduced.

7. Limitations and Future Work

The main limitation of HOI4D is that human manipulation tasks with two hands are not covered since the single-hand manipulation tasks remain challenging for current research. Furthermore, the two-handed setting brings more challenges that we hope to study in the future such as cooperation of hands. As mentioned in Section 3.5, we have built a generic pipeline for creating various CAD models and the corresponding object poses of each HOI4D object category. We hope that our realistic models and rich data from HOI4D can build the bridge between simulation environments and the real world, and inspire more future research for robot learning and augmented reality applications.
References


[23] Shi-Jie Li, Yazan AbuFarha, Yun Liu, Ming-Ming Cheng, and Juergen Gall. Ms-tcn++: Multi-stage temporal convolutional network for action segmentation. IEEE transactions on pattern analysis and machine intelligence, 2020. 8


