OakInk: A Large-scale Knowledge Repository for Understanding Hand-Object Interaction

1,2 Lixin Yang⋆, 1 Kailin Li⋆, 1 Xinyu Zhan⋆, 1 Fei Wu, 1 Anran Xu, 1 Liu Liu, 1,2 Cewu Lu†
1 Shanghai Jiao Tong University, China 2 Shanghai Qi Zhi Institute, China
{siriusyang, kailinli, kelvin34501, legendary, xuanran, liuliu1993, lucewu}@sjtu.edu.cn

Abstract

Learning how humans manipulate objects requires machines to acquire knowledge from two perspectives: one for understanding object affordances and the other for learning human’s interactions based on the affordances. Even though these two knowledge bases are crucial, we find that current databases lack a comprehensive awareness of them. In this work, we propose a multi-modal and rich-annotated knowledge repository, OakInk, for visual and cognitive understanding of hand-object interactions. We start to collect 1,800 common household objects and annotate their affordances to construct the first knowledge base: Oak. Given the affordance, we record rich human interactions with 100 selected objects in Oak. Finally, we transfer the interactions on the 100 recorded objects to their virtual counterparts through a novel method: Tink. The recorded and transferred hand-object interactions constitute the second knowledge base: Ink. As a result, OakInk contains 50,000 distinct affordance-aware and intent-oriented hand-object interactions. We benchmark OakInk on pose estimation and grasp generation tasks. Moreover, we propose two practical applications of OakInk: intent-based interaction generation and handover generation. Our dataset and source code are publicly available at www.oakink.net.

1. Introduction

Enabling a machine to understand and imitate the behavior of humans has been a long-term vision in the history of science. Among the tasks derived from it, learning how humans manipulate objects is a fundamental challenging one. As most tools are designed for achieving function, human can easily learn to manipulate them through instruction or experiences. However, these experiences are hard for a machine to acquire. It was not until recently that data-driven approaches have begun to promote research on learning human manipulation [2, 15, 20, 35, 43, 59]. Prior work has tried to empower a machine complex skills such as hand-object localization [45], pose estimation [29], grasp generation [11], and action imitation [41].

Two fundamental components for learning human manipulation are 1) the affordance of the objects and 2) how human hand would interact with the objects based on those affordances. While the word “affordance” has different formulations in different tasks, in this paper, we denote “affordance” as the functionality of object. Since 2019, there have been at least 9 datasets of hand-object interaction released: ObMan [23], YCBAfford [11], HO3D [19], ContactPose [5], GRAB [51], DexYCB [10], two H2O [28, 57] and DexMV [41]. However, these datasets lack comprehensive awareness of the object’s affordance and the hand’s interactions with it. First, existing real-world datasets only contain a small number of objects and hand interactions. As two illustrative examples, only 20 objects were captured in DexYCB, and only 2.3K distinct interactions were captured among 2.9M images in ContactPose (0.08%). Second, even if synthetic dataset [23] can extend to large numbers of interactions in grasp simulator: GraspIt [33], the generated grasps neither reflect the distribution of human interactions nor consider the object’s affordance itself. To

⋆Equal contribution.
†Cewu Lu is the corresponding author. He is the member of Qing Yuan Research Institute and MoE Key Lab of Artificial Intelligence, AI Institute, Shanghai Jiao Tong University, and Shanghai Qi Zhi institute, China.
understand how humans manipulate objects, we propose to build the machine’s knowledge from two perspectives: object-centric and human-centric perspective. To this end, we construct two interrelated knowledge bases. One is an Object Affordance Knowledge base (Oak base, Sec. 3.1) in which we provide comprehensive descriptions of objects’ affordances within a knowledge graph, and the other is an Interaction Knowledge base (Ink base, Sec. 3.2) in which we collect diverse human hand interactions that provide demonstrations of manipulating the object according to its affordances.

To construct the Oak base, we firstly collect 1,800 household objects that are designed for single-hand manipulation. The sources of objects in Oak base are four-fold: 1) self-collected from online vendors, 2) ShapeNet [9] models, 3) YCB [6] and 4) ContactDB [3] objects. Second, through exhaustively reviewing the objects in the above sources, we build an object knowledge graph that arranges objects with two types of abstractions, namely taxonomy and attribute (Fig. 2). This object knowledge graph enables us to make a quick extension for new objects and conduct convenient clustering for objects of similar affordance.

To construct the Ink base, we start to collect human experiences on performing hand-object interactions based on the object’s affordance. We select 100 representative objects from Oak base, invite 12 human subjects to perform demonstrations, and set up a multi-sensor MoCap platform for recording (Fig. 3). The recorded sequences constitute a real-world image dataset that contains 230,064 RGB-D frames capturing 12 subjects performing up to 5 intent-oriented hand interactions with objects in a pool of total 100 instances from 32 categories. The objects that appeared in the recorded sequences are denoted as the “source” objects. Next, given the real-world human demonstration, we aim to transfer their experience on the source object to its virtual counterparts with similar affordances (target objects). The transferred hand interaction should not only ensure its physical plausibility, but also keep the consistent intent and match the size, shape, and affordance of the target object (Fig. 1). To this end, we propose a learning-fitting hybrid method: Tink for Transferring the Interaction Knowledge among objects (Sec. 3.3). Tink consists of three modules: namely an implicit shape interpolation, an explicit contact mapping, and an iterative pose refinement. With Tink, we extend the total number of distinct hand-object interactions in Ink base to 50,000.

Through combining the above two knowledge bases: Oak and Ink, we construct a large-scale knowledge repository: OakInk. The advantages of our OakInk are three-fold: 1) It provides comprehensive knowledge for understanding hand-object interactions from two perspectives: object affordances and human experiences; 2) It contains two large-scale datasets of image-based and geometry-based hand-object interaction; 3) It provides rich annotations including hand and object poses, scanned object models, affordances, fine-grained contact and stress patterns, and intents labels. OakInk can benefit researches in two communities: 1) pose estimation [13,21,30], shape reconstruction [23,25], and action recognition [15,28,52] in computer vision, CV; 2) grasp generation [24,51,59] and motion synthesis [7,38] in computer graphics, CG. Among all the topics above, we find pose estimation and pose generation are most relevant to our interests. In this paper, we benchmark OakInk on three existing tasks and propose two new tasks: One is an intent-based hand pose generation and the other is a human-to-human handover generation.

Our contributions are concluded in three-fold. First, we construct OakInk, a large-scale knowledge repository for understanding hand-object interactions. Second, inside OakInk, we propose a novel method Tink that transfers the interaction knowledge among objects with similar affordance. Finally, we provide extensive evaluations for benchmarking OakInk on three existing tasks and propose two novel tasks: generating plausible hand poses for more customized purposes.

2. Related Work

Datasets of Hand-Object Interaction (HOI). Current HOI datasets can be categorized as real-world and synthetic based on the data source. ObMan [23] and YCBAfford [11] represent the synthetic datasets that leveraged grasp simulators to synthesize or label static grasps. Real-world datasets are categorized into three types based on how they collect the annotations. 1). The marker-based datasets collect hand poses with the aid of hand-attached magnetic sensors [15,57,58] or reflective markers [51]. 2). The automatic marker-less datasets [5,19] aggregate the visual cues from methods of detection, segmentation and pose estimation to acquire annotations automatically. 3). The crowd-sourced marker-less datasets leverage human annotators to label the 2D poses of hands and objects [10]. In this paper, we collect 3D hand pose annotations through crowd-sourcing its 2D keypoints and optimizing them within multi-views. For object pose, we record its surface-attached reflective wafers in a synchronized MoCap system (Sec. 3.2.3). We will present the comprehensive comparisons and statistics of existing datasets in Sec. 3.4 (Tab. 1).

Contact of Hand-Object Interaction. In order to capture contact, previous methods used measurement devices like force transducers [39], tactile sensors [50] and thermal cameras [3,5], or computed realistic contact through accurate pose tracking [51]. As contact can provide rich cues to reason the joint hand-object poses during interactions, recent methods leveraged contact to help optimize grasps in reconstruction [25,55] and synthesis [24] tasks. In this paper, we derive contact regions and their stress pat-
terns through accurate pose tracking (Sec. 3.2.3). Later in Sec. 3.3, we map the contacts to virtual objects and optimize poses based on the contacts.

**Pose of Hand-Object Interaction.** Pose estimation is a common task for understanding how human manipulate objects. Previous methods either focus on hand [53] or object [54] pose alone. Hasson et al. [23] have proposed the first conjoint hand-object pose estimation methods that brought the renaissance in this area [8,13,21,22,29,30]. Another popular task of HOI is grasping pose generation. Researchers in this track have delved into synthesizing prehensile grasps based on image [11] or shape observation [25]. Many derivative tasks like: action recognition [15,28], imitation learning [42], teleoperation [20], and human-to-robot handover (and vise versa) [48,56] are powered by the above two tasks. In this paper, we benchmark our dataset on the classical pose estimation and pose generation tasks. We also introduce two interesting tasks that explore the generative model with a given intent and within a handover scenario.

### 3. Constructing the OakInk

*OakInk* consists of two interrelated knowledge bases. One is the object-centric affordance knowledge: *Oak* base, and the other is the human-centric interaction knowledge: *Ink* base. Once we decide the composition of *OakInk*, three questions shall be answered. 1) How to represent the objects’ affordances? 2) How to record human experiences on manipulating the objects based on the affordances? 3) How to transfer the recorded interactions to those objects with similar affordance? To address these questions, we describe the construction of the *Oak* base in Sec. 3.1, present how we record and annotate the human demonstration in Sec. 3.2, and introduce a novel interaction knowledge transfer method in Sec. 3.3. Finally, we provide the statistics and analysis in Sec. 3.4.

#### 3.1. Object-Centric Affordance Knowledge Base

We focus on objects that are commonly appeared in our daily life and are designed for single-hand manipulations. We collect a total of 1,800 household objects for these purposes. The source of these objects are four-fold: 1) self-collected from online vendors, 2) ShapeNet models, 3) YCB objects, and 4) ContactDB objects, in which we observe diverse object categories, shapes, and affordances. We organize all the objects into a knowledge graph (Fig. 2). The knowledge graph, as well as the objects’ scanned models form the main body of *Oak* base. Next, we will elaborate on how we arrange the objects (taxonomy) and how we describe the affordance of the objects (attribute). The taxonomy and attribute in *Oak* base should achieve 1) consensus that the consistent classification shall be made by a group of people based on their common experience, and have 2) scalability that new objects and new attributes can be easily extended to the current knowledge base. After exhaustively reviewing the objects in the above datasets, we found the taxonomy and the description of attributes can be concluded within limited categories.

**Taxonomy.** We adopt a taxonomy that groups *Oak* base objects into two levels of classifications. We define the top-level classification that consists of two classes, namely manipulation tools (*manipTool*) and function tools (*funcTool*). Their definitions are as follows:

- **manipTool** class contains tools that are used to manipulate (affect) other entities. These objects usually contain a handle (for grasping) and an end effector (for affecting other entities). (e.g. mug, knife, pincer and drill);
- **funcTool** class manages tools that usually have a self-contained function and do not necessarily require end effectors. (e.g. camera and headphone);

Within the top-level classifications, we arrange the objects based on the WordNet [34] categories as the sub-level classification. The total number of categories in the *Oak* base is 32. We list all the categories in *Appx*.

**Attribute.** The notion of “affordance” was introduced by Gibson [17] as the characteristics of the functional properties of objects. Later in CV and robotics community, the affordance has been used with different formulations, such as graspable area [26, 27], grasp types [11], part segmentation [12, 35], contact region [4], and action-effects [14]. In this paper, we denote the “affordance” as the functionality of object. The affordance is represented by a set of *attributes*. Each *attribute* contains a part segmentation with one or several descriptive phrase(s) that describe the part’s functionality. For example, given a knife with two parts: blade and handle, we assign the phrase ⟨cut, something⟩ to the blade, and the phrase ⟨handled (by), something⟩ to the handle. We invite 10 volunteers with different backgrounds and ask them to firstly make a phrase of: ⟨verb (+ prep), something⟩ to describe each part of the objects. We only focus on parts with functionality. Hence for parts that may not have function, we ask volunteers to judge and assign them as ⟨no function⟩. We also encourage volunteers
to conclude the part-level similarity across different object categories. In the beginning, we create an empty candidate phrase pool. When a new phrase was initially proposed on a certain part, we first check whether it has a duplicated meaning in the candidate pool. Then, we seek the consensus among all the volunteers on whether to replace or add it. Finally, we gather all the phrases, conclude their meaning and vote for their occurrence on each part. Through exhaustively reviewing all the 32 object categories, we conclude total 30 phrases as the final attribute phrases. We list all the attribute phrases in Appx.

3.2. Human-Centric Interaction Knowledge Base

In this section, we elaborate on how we collect human demonstration and construct the Ink base. We first introduce the hardware setup for efficient recording in Sec. 3.2.1, provide a protocol for data acquisition in Sec. 3.2.2 and describe the details of data annotation in Sec. 3.2.3.

3.2.1 Hardware Setup

The data collection platform consists of a multi-camera system (MulCam) and an infrared motion capture system (MoCap). The MulCam system consists of 4 RealSense D435 cameras that are used to record the image-based interaction sequence. The MoCap system consists of 8 Optitrack Prime 13W cameras that are used to track the object’s motion during the interaction. We synchronized all the cameras in both sensor systems and calibrated the transformation between the MulCam system: $S_s$ and MoCap system: $S_m$. Our platform is shown in Fig. 3. All the sensors are rigidly mounted on the edges of a $1.5 \times 1.2 \times 1 \text{ m}^3$ cuboidal area, which enables the subject to freely interact with objects or other subjects without interference.

3.2.2 Interaction Sequence Acquisition

We invited 12 subjects and recorded their interactions with the given objects. Each subject is assigned a subset from the object database. A director will firstly elaborate on the attributes of each object and confirm the acknowledgment of these attributes among all the subjects. Then the subjects are asked to start from a hand pose lying flat on the table, pick up the assigned object, and finish the action with a given intent. For each object, we collect up to 5 intents, namely use, hold, lift-up, hand-out, and receive. The intent: use requires the subject to perform an action that makes use of the object’s attribute(s). The hold requires the subject to perform a steady grasp of the object. The lift-up asks the subject to pick up an overturned object and place it upright. When a subject was asked to hand-out an object, this subject (the giver) was also paired with another subject (the receiver) to perform receive. The paired sequences of hand-out and receive constitute an action of human-to-human handover. During handover, the giver was asked to determine where the receiver would receive the object to

Figure 3. Our data collection platform with 4 RGB-D cameras (red circle) and 8 infrared MoCap cameras (blue circle).

use. Meanwhile, the receiver was asked to determine how to receive the object from the giver without mutual contact. After each action finishes, a director will place the object with a random pose for the next action. We record each action for 5 seconds and manually discard the idle frames.

3.2.3 Data Annotation

During the entire course of the human demonstration, we are particularly interested in the poses and contact pattern of hand and object, as they embrace the human experience of manipulating objects.

Object Pose. We track the object’s 6 DoF pose by tracking the surface-attached reflective markers (Fig. 4 left) in the MoCap system $S_m$. Then, we transform the object pose from $S_m$ to the MulCam system $S_s$ by the system calibration.

Hand Pose and Geometry. We rely on manually labeled 2D hand keypoints from multi-views to acquire the 3D hand joints annotation. Following the practice in [10], we set up an annotation task on an online crowd-sourcing platform and require workers to locate every keypoint in all 4 views of all the assigned frames. We adopt the standard 21 hand keypoints following the orders and locations defined in [46]. To describe hand pose and geometry in 3D space, we use the MANO hand model [44]. MANO represents an articulated and deformable hand by pose $\theta \in \mathbb{R}^{16 \times 3}$ and shape $\beta \in \mathbb{R}^{10}$ parameter. Later in the paper, we denote “hand pose” as the 21 joint positions: $P_h \in \mathbb{R}^{21 \times 3}$ in the $S_c$ system. With $\theta$ and $\beta$, we can recover the hand pose $P_h$ and mesh vertices $V_h \in \mathbb{R}^{778 \times 3}$ through a differentiable MANO layer: $\mathcal{M}(\cdot)$ [23]. Solving the $P_h$ and $V_h$ are formulated as an optimization task minimizing several handcrafted cost functions. In the main paper, we only describe the core term: 3D-2D keypoints re-projection error among multi-views. For other auxiliary costs such as geometrical consistency, temporal smoothing, and silhouette constraint, please visit Appx. Let $\hat{p}_{j,v}$ be the $j$-th 2D hand keypoint annotation in the $v$-th view, $P_{h,b}$ be the $j$-th 3D hand pose estimation, and let $T_v, K_v$ be the extrinsic and intrinsic of the camera of $v$-th view, we define the re-projection cost as:
Contact and Stress Pattern. Given the accurate hand and object poses, we can derive the per-hand-part contact region on object surfaces. We adopt the 17 hand parts segmentation and part-level anchor location in Yang et al. [55]. Based on the efficient contact heuristic in GRAB [51], we automatically assign a part label to those vertices on the object’s surface if an anchor is close to them (within the threshold of 25 mm). The vertices with a labeled hand part form the contact regions of the object. Physical contact commonly results in an elastic deformation of both hand and object [18], in which the stress and strain will spread across the deformation area. Though MANO and rigid object model cannot reflect this behavior, we can imitate the stress pattern by adding a ring-shape spreading and decreasing value in the contact region, as we call it contactness. As shown in Fig. 4 right, the contactness takes the maximum value 1 at the point closest to a certain anchor, centrally decreasing as the distance increases, and finally becomes 0 when the distance is greater than 25 mm. We delay the demonstration on the use of contactness until Sec. 3.3.3.

3.3. Tink: Transferring Interaction Knowledge

This section describes how we transfer the hand’s interactions with the real-world objects (recorded in human demonstration) to the virtual counterpart objects (collected in Oak base) of the same category. The transferred interactions should be consistent with those collected regarding contact, pose, intent, and human perception. However, as different objects vary in shapes and sizes, direct pose copying (as shown in Fig. 5 left) would fail in most cases. To address this issue, we propose a hybrid learning-fitting method: Tink for Transferring the interaction knowledge. Tink consists of three sequential modules, namely shape interpolation, contact mapping, and pose refinement.

We refer to the objects that have been recorded in real-world human demonstrations as the source objects, and the virtual objects in Oak as the target objects. As a recorded sequence only has one type of hand-object interaction (handover sequence has two), we manually select 1 (or 2) steady interacting pose(s) to represent each sequence. These selected hand poses are the source poses for interaction transfer. Later, we call the set of those selected interactions OakInk-Core.

3.3.1 Implicit Shape Interpolation

Once we decide to transfer the interaction from one source object to another target object, an instant question is how to express the object shape and perform continuous shape deformation. To answer this, we first represent the object shape as an implicit function (SDF, signed distance function), as SDF is naturally continuous. Now the question is how to perform the shape interpolation between the SDF of source and target. To address this, we adopt a neural generative model: DeepSDF [37] that maps complex 3D shapes into a continuous latent space. Using DeepSDF has three advantages. 1) We can acquire a compact representation of complex shape, namely the shape vector; 2) We can perform accurate shape interpolation by interpolating the shape vectors in the latent space; 3) Later in the Sec. 3.3.3, we can mitigate the hand-object interpenetration by penalizing the negative query positions (Eq. (4));

We firstly train a DeepSDF model on all the source and target object SDFs of a certain category. Then for the i-th source object SDF: $O_i^s$ and j-th target object SDF: $O_j^t$, we perform linear interpolation between their latent shape vector: $o_i^s$ and $o_j^t$. During the interpolation, we sample $N_{itpl}$ equally spaced quantiles as landmarks. Finally, We decode the shape vector at landmarks to its SDF and reconstruct a mesh model by Marching Cubes [31]. The $N_{itpl}$ artificial objects constitute a path connecting the source and target.

3.3.2 Explicit Contact Mapping

As shown in Fig. 5 left, directly copying the hand pose would fail. We need to find a piece of consistent information shared among the source, the target, and the $N_{itpl}$ landmarks along the path. Compared to pose, contact regions are more invariant against shape deformation. We start mapping contact regions from the source object, sequentially pass through the $N_{itpl}$ landmarks, and finally reach the target object. As long as the interval between each two landmarks is small enough, we can neglect shape variation between the i-th and (i+1)-th objects. The contact mapping is illustrated in Fig. 5 (contact regions of different finger parts are painted with different colors). Considering the trade-off between efficiency and accuracy, we empirically find that $N_{itpl} = 10$ is sufficient enough. We map the contact label of a vertex on the i-th object to its closest vertex on the (i+1)-th object. At each i $(0 \leq i < N_{itpl})$ step, we adopt the iterative closest point (ICP) to link the corresponding vertices.

3.3.3 Iterative Pose Refinement

In the last module, we map the interacting hand pose of the source object to its counterpart target objects. As
we express the knowledge of interaction as the semantics on the object surface, namely the contact regions (recall Sec. 3.2.3), pose mapping is conducted by enforcing the contact consistency between the source and target object. We formulate pose mapping as an iterative optimization. The variables during optimization are the pose \( \theta \), shape \( \beta \), and wrist position \( P_h \), of a newly transferred hand.

We start to attract the anchors on the hand surface to its corresponding contact regions on the target object. Let the anchors of total 17 hand regions be \( \mathcal{A} = \{ A_i \}_{i=1}^{17} \), the vertices on the object surface that corresponds to the anchor \( A_i \) be \( V_h^{(i)} = \{ V_{h,j}^{(i)} \} \), and the contactness between \( A_i \) and \( V_h^{(i)} \) be \( \gamma_{ij} \). The contact consistency cost is expressed as:

\[
E_{\text{cons}} = \frac{1}{\sum \gamma_{ij} \sum A_i V_{h,j}^{(i)} || A_i - V_{h,j}^{(i)} ||^2},
\]

Direct optimization on the joints’ rotations \( \theta \) is prone to anatomical abnormality. Hence we adopt the axial adaptations from Yang et al. [55] and constrain the rotation axes and angles. Let \( \alpha_j \) and \( \phi_j \) be the axial and angular components of the \( j \)-th joint’s rotation, the \( n_j^a \) and \( n_j^s \) be the pre-defined twist and splay direction. The anatomical cost is defined as:

\[
E_{\text{anat}} = \sum_{j \in \text{all}} \left( \alpha_j n_j^a + \max \left( \left( \phi_j - \frac{\pi}{2} \right), 0 \right) \right) + \sum_{j \in \text{MCP}} \alpha_j n_j^s,
\]

where “MCP” indicates the five Metacarpal joints.

In order to control the hand-object interpenetration, we also introduce an interpenetration cost to penalize those hand vertices inside the object surface:

\[
E_{\text{intp}} = \sum_{V_{h,j}} - \min \left( \text{SDF}(V_{h,j}), 0 \right),
\]

where the SDF evaluates the signed distance value of a 3D hand vertex \( V_{h,j} \) to an object’s SDF: \( \Omega \) provided at shape interpolation (Sec. 3.3.1). The total optimization problem is:

\[
V_h, P_h \leftarrow \arg \min_{\theta, \beta, P_h} \left( E_{\text{cons}} + E_{\text{anat}} + E_{\text{intp}} \right),
\]

where \( V_h, P_h = \mathcal{M}(\theta, \beta) + P_h \). We run 1,000 iterations per source-target pair. The whole pipeline is implemented in PyTorch with Adam solver.

**Perceptual Evaluation.** Finally, all the transferred interactions are sent to 5 volunteers for perceptual evaluation. Given the source object and its interacting hand pose as a reference, the volunteers are asked to make a judgment on whether the transferred hand pose on target object demonstrates the same intents and satisfies visual plausibility. We only select the interactions that achieve consensus on plausibility among the 5 volunteers.

### 3.4 Dataset Analysis

In this section, we provide the statistic and analysis of OakInk. As a summary, we collected 230K image frames of 12 subjects performing up to 5 intent-oriented interactions with total 100 real-world objects of 32 categories, and transferred the interactions to the rest of 1,700 virtual counterpart objects. The total number of distinct hand-object interactions is 50,000. We denote the image-based dataset as OakInk-Image. We select 1 (or 2) representative hand-object interaction for each image sequence and denote their collection as OakInk-Core. All the selected and transferred interactions constitute another geometry-based dataset: OakInk-Shape. We make a comprehensive comparison with the existing hand-object datasets in Tab. 1, and visualize hand pose and contact distribution in Appx.

**Image Dataset Cross Validation.** To evaluate the merit of OakInk-Image, we perform cross-dataset validation in Tab. 2. We train an image-based 3D pose estimation model [49] separately on three training sets: HO3D, OakInk-Image, and their mixture, and report the hands’ MPJPE on DexYCB testing set. We observe consistent MPJPE improvements on the model trained on OakInk-Image (alone and mixture), verifying that OakInk-Image complements HO3D dataset and improves the network model.

**Geometry Dataset Qualities.** To evaluate the quality of OakInk-Shape, we inspect several physical-based metrics that assess the feasibility and stability of grasps. We also compare those metrics with the other three datasets: FPHAB [16], GRAB (GrabNet split) [51], and HO3D [19], representing three different data annotation methods: active magnetic transmitter, passive reflective markers, and automatic marker-less, respectively. Tab. 3 shows that OakInk-Shape exhibits high physical-based qualities.
Table 1. Comparison of our OakInk with the publicly available datasets of hand-object interactions.

| Dataset        | mod. resolution | #frame | #subj | #obj | #views | #inten | #intact | real/label | intac. | obj pose | dynamic | hand-over | hand-obj contact |
|----------------|-----------------|--------|-------|------|--------|--------|---------|------------|--------|----------|---------|-----------|-----------------|
| ObMan [23]     | RGBD 256 × 256  | 154K   | 20    | 3K   | 1      | –      | –       | syn simulate | ✓      | ✓        | ✓       | ✓         | ✓               |
| YCBAfford [11] | RGB 133K 1 21 | 1      | –     | –    | –      | –      | 367     | syn simulate | ✓      | ✓        | ✓       | ✓         | ✓               |
| FPHAB [16]     | RGBD 1920 × 1080 | 105K   | 6     | 4    | 1      | 3      | 273     | real marker | ✓      | ✓        | ✓       | ✓         | ✓               |
| HO3D [19]      | RGBD 640 × 480  | 78K    | 10    | 10   | 1-5    | –      | 68      | real auto   | ✓      | ✓        | ✓       | ✓         | ✓               |
| ContactPose [5] | RGBD 960 × 540 | 2991K  | 50    | 25   | 3      | 2      | 2.3K    | real auto   | ✓      | ✓        | ✓       | ✓         | ✓               |
| GRAB [51]      | Mesh 1624K 10   | 51     | 4     | 1.3K | 4      | real marker | ✓      | ✓        | ✓       | ✓         | ✓               |
| DexYCB [10]    | RGBD 640 × 480  | 582K   | 10    | 20   | 8      | 1K     | real crowd | ✓      | ✓        | ✓       | ✓         | ✓               |
| H2O [28]       | RGBD 1280 × 720 | 571K   | 4     | 5    | 7      | 1.8K   | real auto | ✓      | ✓        | ✓       | ✓         | ✓               |
| Ours OakInk-Image | RGBD 848 × 480 | 230K   | 12    | 100  | 4      | 5      | 1K      | real crowd | ✓      | ✓        | ✓       | ✓         | ✓               |
| Ours OakInk-Shape | Mesh 848 × 480 | 49K    | –     | –    | 12     | 1,700  | 5 49K    | real 7Ink  | ✓      | ✓        | ✓       | ✓         | ✓               |

Table 2. Cross dataset validation on OakInk-Image

| Metrics          | +Core | +Shape | FPHAB | GRAB | HO3D |
|------------------|-------|--------|-------|------|------|
| Penet. Depth. cm | 0.18  | 0.11   | 1.95  | 2.53 | 1.16 |
| Solid Intsec. Vol. cm³ | 1.03 | 0.62   | 22.87 | 7.61 | 2.08 |
| Sim. Disp. Mean cm | 0.98  | 0.94   | 6.60  | 2.04 | 1.91 |
| Sim. Disp. Std cm | 1.74  | 1.62   | 5.34  | 3.17 | 2.88 |

Table 3. Quality assessment of our OakInk with the publicly available datasets of hand-object interactions.

Table 4. HMR results in mm. AUC are shown in parentheses.

| Splits | Methods          | MPJPE (AUC) | MPVPE |
|--------|------------------|-------------|-------|
| SP0    | I2L-MeshNet [36] | 12.10 (0.784) | 12.29 |
| HandTailor [32] | 11.20 (0.884) | 11.75 |

Table 5. HOPE results in mm. AUC are shown in parentheses.

| Method                  | MPJPE | MPCPE | ∗MPCPE | sMPCPE |
|-------------------------|-------|-------|--------|--------|
| Hasson et al. [21]      | 27.26 | 56.09 | 68.40  | 60.70  |
| Tekin et al. [52]       | 23.52 | 52.16 | 57.29  | 57.11  |

Figure 6. Qualitative results of I2L-MeshNet [36] on HMR task (top row), and Hassen et al. [21] on HOPE task (bottom row).

4. Tasks and Benchmark Results

We benchmark three existing tasks (Sec. 4.1-4.3) and propose two novel tasks (Sec. 4.4) on our OakInk. The three existing tasks are: 3D hand mesh recovery (HMR, Sec. 4.1) [32, 36], 3D hand-object pose estimation (HOPE, Sec. 4.2) [21, 52], and grasp pose generation (GraspGen, Sec. 4.3) [51]. The two novel tasks are intent-based interaction generation (IntGen, Sec. 4.4 A) and human-to-human handover generation (HoverGen Sec. 4.4 B).

4.1. Hand Mesh Recovery

The HMR task is to estimate the hand pose \( P_h \in \mathbb{R}^{21 \times 3} \) and geometry \( V_h \in \mathbb{R}^{778 \times 3} \) from a single image. To benchmark OakInk on this task, we first generate the train/test splits of the image frames collected in Sec. 3.2. We call this image-based subset: OakInk-Image. We randomly select one view per sequence and mark all images from this view as the test sequence, while the rest are used as the train sequence. We use \( \text{train/val} \) splits (train/val/test: 70%/5%/25%). We call this split SP0 (default split). Next, we benchmark two HMR methods: one is a direct image-to-vertices method: I2L-MeshNet [36], and the other is a hybrid inverse kinematic method: HandTailor [32]. We evaluate these methods with three metrics: mean per joint position error (MPJPE), percentages of correct keypoints under the curve (AUC) within range: \([0, 50]\), and mean per vertex position error (MPVPE) in wrist-relate system. We show results on

SP0 test set in Tab. 4 and Fig. 6 (top row). More quantitative results on other splits are provided in Appx.

4.2. Hand-Object Pose Estimation

The HOPE task is to simultaneously estimate the hand pose \( P_h \) and the object pose (rotation \( R_o \in \text{SO}(3) \), center translation \( t_o \in \mathbb{R}^3 \)) from a single image. Most previous methods focused on instance-level object pose estimation. The object models (in the form of mesh vertices or corners) are provided as input when computing the loss during training. Following the same protocol, we train and test the neural networks on the same objects. The data split for training HOPE task follows OakInk-Image SP0.

We benchmark two representative HOPE architecture designs: Tekin et al. [52] and Hasson et al. [21]. To note, as these two methods output the object pose in different ways, we provide adaption layers at their output. We represent object pose as the oriented 8 corners on 3D object bounding box. We evaluate these methods with two metrics: MPJPE and mean per corners position error (MPCPE), both in the hand wrist-relate system. We show the test set results in Tab. 5 and Fig. 6 (bottom row).
4.3. Grasp Pose Generation

The GraspGen task is to generate diverse hand poses that interact with a given object shape. Existing GraspGen methods [24, 25, 51] widely adopted a conditional VAE [47] architecture to this end. As shown in Fig. 7 (1), the model is trained with an object shape (as BPS [40] $\in \mathbb{R}^{4096}$) and its interacting hand pose $(\theta_0, P_{h0})$ as input, and is supervised to generate the consistent hand with the input hand. As a result, the model learns an object-conditioned hand embedding space: $Z$. Then during the testing, given a test object, the model decodes a hand pose from its embedding space $Z$. To benchmark our OakInk on GraspGen, we randomly select 80% of objects from Oak base for training, 10% for validation, and the rest 10% for testing. All the object models are paired with their interacting hand poses in group. We denote this shape-based subset as OakInk-Shape.

We benchmark OakInk-Shape on GrabNet [51], a representative method toward GraspGen. The evaluation metrics of GraspGen consist of 4 terms. We evaluate 1) penetration depth, 2) solid intersection volume following [55], and 3) simulation displacement following [23]. To investigate general audience’s opinion about the generated poses, we also provide a 4) perceptual survey on the network predictions at Amazon Mechanical Turk [1] following the practices in [24, 25, 51]. We ask the workers to rate the generated hand poses with scores ranging from 1 (strongly unsatisfied) to 5 (strongly satisfied). Protocols and demonstrations of this perceptual survey are shown in Appx. We report all the four evaluation results in Tab. 6 column 2.

4.4. Two Novel Generation Tasks

Previous GraspGen methods can only generate general grasp poses that are agnostic toward intents. In this paper, we investigate pose generation with two applicable purposes, namely, A), to generate plausible poses with a specific intent and B), to generate plausible poses for receiving the objects from a giver. We illustrate their network design in Fig. 7. For implementation details, please visit Appx.

**A) Intent-Based Interaction Generation.** We start modifying the network design in GrabNet. As shown in Fig. 7 (1)+(2), apart from the object shape (original condition), we introduce another condition: the word embedding of a given intent. The model learns a hand embedding space conditioned on two dimensions: shape and intent. During testing, given by a test object and an assigned intent, the model decodes an intent-based interacting pose that is shown in Fig. 8 middle. We provide the evaluation results in Tab. 6.

**B) Handover Generation.** We provide another modification of GrabNet that take the object shape as well as the giver’s hand as conditions (Fig. 7 (1)+(3)). The model learns to decode a receiver’s hand for achieving human-to-human handover. We provide evaluation on several test objects in Tab. 6 last column and Fig. 8 right. The generated receiver hand matches our expectation: the receiver’s hand should avoid colliding with or hindering the retraction path of the giver’s hand.

5. Discussion

**Limitation.** Current OakInk does not record dynamic hand interactions with the movable parts of the articulated object (e.g. scissors), and does not consider transferring the interaction knowledge from human hands to the multi-finger robot arms. We will address the limitations in future works.

**Conclusion.** In this work, we constructed a large-scale knowledge repository OakInk that builds machines’ knowledge on understanding human hand-object interactions. OakInk consists of two interrelated knowledge bases Oak and Ink that contain rich data and experiences. Even though we only benchmark OakInk on CV and CG tasks, we are quite eager to apply OakInk to the robotics community and explore future chances for robot learning.

**Acknowledgment.** This work was supported by National Key Research and Development Project of China (No.2021ZD0110700), Shanghai Municipal Science and Technology Major Project (2021SHZDZX0102), Shanghai Qi Zhi Institute, and SHEITC (2018-RGZN-02046). The computing resources were provided by High-Flyer AI.
References

[1] Amazon Mechanical Turk. https://www.mturk.com.

[2] Dafni Anotsiou, Guillermo Garcia-Hernando, and Tae-Kyun Kim. Task-oriented hand motion retargeting for dexterous manipulation imitation. In ECCV Workshops, 2018.

[3] Samarth Brahmbhatt, Cusuh Ham, Charles C. Kemp, and James Hays. ContactDB: Analyzing and Predicting Grasp Contact via Thermal Imaging. In CVPR, 2019.

[4] Samarth Brahmbhatt, Ankur Handa, James Hays, and Dieter Fox. ContactGrasp: Functional Multi-finger Grasp Synthesis from Contact. In IROS, 2019.

[5] 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 ECCV, 2020.

[6] Berk Calli, Arjun Singh, Aaron Walsman, Siddhartha Srinivasa, Pieter Abbeel, and Aaron M. Dollar. The ycb object and model set: Towards common benchmarks for manipulation research. In ICAR, 2015.

[7] Zhe Cao, Hang Gao, Karttikeya Mangalam, Qi-Zhi Cai, Minh Vo, and Jitendra Malik. Long-term human motion prediction with scene context. In ECCV, 2020.

[8] Zhe Cao, Ilija Radosavovic, Angjoo Kanazawa, and Jitendra Malik. Reconstructing hand-object interactions in the wild. In ICCV, 2021.

[9] Angel X. Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Lo Yi, and Fisher Yu. ShapeNet: An Information-Rich 3D Model Repository. Technical Report arXiv:1512.03012, Stanford University, Princeton University, Toyota Technological Institute at Chicago, 2015.

[10] Yu-Wei Chao, Wei Yang, Yu Xiang, Pavlo Molchanov, Ankur Handa, Jonathan Tremblay, Yashraj S. Narang, Karl Van Wyk, Umar Iqbal, Stan Birchfield, Jian Kautz, and Dieter Fox. DexYCB: A benchmark for capturing hand grasping of objects. In CVPR, 2021.

[11] Emric Corona, Albert Pumarola, Guillem Alenyà, Francesc Moreno-Noguer, and Grégory Rogez. Ganhand: Predicting human grasp affordances in multi-object scenes. In CVPR, 2020.

[12] Thanh-Toan Do, Anh Nguyen, and Ian Reid. Affordancenet: An end-to-end deep learning approach for object affordance detection. In ICRA, 2018.

[13] Bardia Doosti, Shujon Naha, Majid Mirbagheri, and David Crandall. Hope-net: A graph-based model for hand-object pose estimation. In CVPR, 2020.

[14] Kuan Fang, Yuke Zhu, Animesh Garg, Andrey Kurenkov, Viraj Mehta, Li Fei-Fei, and Silvio Savarese. Learning task-oriented grasping for tool manipulation from simulated self-supervision. The International Journal of Robotics Research, 2020.

[15] Guillermo Garcia-Hernando, Shaxin Yuan, Seungryul Baek, and Tae-Kyun Kim. First-person hand action benchmark with rgb-d videos and 3d hand pose annotations. In CVPR, 2018.

[16] Guillermo Garcia-Hernando, Shaxin Yuan, Seungryul Baek, and Tae-Kyun Kim. First-person hand action benchmark with rgb-d videos and 3d hand pose annotations. In CVPR, 2018.

[17] James J Gibson. The ecological approach to visual perception: classic edition. Psychology Press, 2014.

[18] Patrick Grady, Chengcheng Tang, Christopher D Twigg, Minh Vo, Samarth Brahmbhatt, and Charles C Kemp. Contactopt: Optimizing contact to improve grasps. In CVPR, 2021.

[19] Shreyas Hampali, Mahdi Rad, Markus Oberweger, and Vincent Lepetit. Honnotate: A method for 3d annotation of hand and object poses. In CVPR, 2020.

[20] Ankur Handa, Karl Van Wyk, Wei Yang, Jacky Liang, Yu-Wei Chao, Qian Wan, Stan Birchfield, Nathan D. Ratliff, and Dieter Fox. DexPilot: Vision-based teleoperation of dexterous robotic hand-arm system. In ICRA, 2020.

[21] Yana Hasson, Bugra Tekin, Federica Bogo, Ivan Laptev, Marc Pollefeys, and Cordelia Schmid. Leveraging photometric consistency over time for sparsely supervised hand-object reconstruction. In CVPR, 2020.

[22] Yana Hasson, Güll Varol, Ivan Laptev, and Cordelia Schmid. Towards unconstrained joint hand-object reconstruction from rgb videos. In arXiv preprint arXiv:2108.07044, 2021.

[23] Yana Hasson, Güll Varol, Dimitrios Tzionas, Igor Kalevatykh, Michael J Black, Ivan Laptev, and Cordelia Schmid. Learning joint reconstruction of hands and manipulated objects. In CVPR, 2019.

[24] Hanwen Jiang, Shaowei Liu, Jiashun Wang, and Xiaolong Wang. Hand-object contact consistency reasoning for human grasps generation. In ICCV, 2021.

[25] Korrawe Karunratanakul, Jinlong Yang, Yan Zhang, Michael J Black, Krikamol Muandet, and Siyu Tang. Grasping field: Learning implicit representations for human grasps. In 3DV, 2020.

[26] Mia Kockic, Danica Kragic, and Jeannette Bohg. Learning task-oriented grasping from human activity datasets. IEEE Robotics and Automation Letters (RAL), 2020.

[27] Mia Kockic, Johannes A Stork, Joshua A Haustein, and Danica Kragic. Affordance detection for task-specific grasping using deep learning. In Humanoids, 2017.

[28] Taerin Kwon, Bugra Tekin, Jan Stuhmer, Federica Bogo, and Marc Pollefeys. H2O: Two hands manipulating objects for first person interaction recognition. In ICCV, 2021.

[29] Kailin Li, Lixin Yang, Xinyu Zhan, Jun Lv, Weniqiang Xu, Jiefeng Li, and Cewu Lu. ArtiBoost: Boosting articulated 3d hand-object pose estimation via online exploration and synthesis. In CVPR, 2022.

[30] Shaowei Liu, Hanwen Jiang, Jiari Xu, Sifei Liu, and Xiao-long Wang. Semi-supervised 3d hand-object poses estimation with interactions in time. In CVPR, 2021.

[31] William E Lorensen and Harvey E Cline. Marching cubes: A high resolution 3d surface construction algorithm. ACM Siggraph Computer Graphics, 1987.
