arXiv:2203.15709v1 [cs.CV] 29 Mar 2022

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 datasets and source code are publicly available at https://github.com/lixiny/OakInk.

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, 36, 44, 61]. Prior work has tried to empower a machine complex skills such as hand-object localization [46], pose estimation [30], grasp generation [11], and action imitation [42].

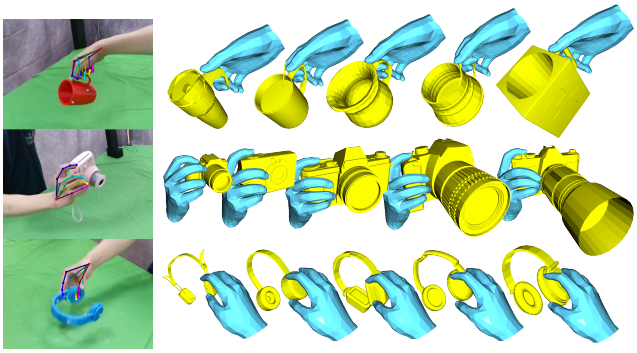

Figure 1. Illustration of different data modalities in *OakInk* repository. The left column shows human manipulating 3 source objects (mug, camera, and headphones). The right 5 columns show the transferred interactions on 15 virtual counterpart objects.

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 [52], DexYCB [10], two H2O [29, 59] and DexMV [42]. 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 [34], 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 **O**bject **A**ffordance **K**nowledge base (***Oak*** base, Sec. 3.1) in which we provide comprehensive descriptions of objects' affordances within a knowledge graph, and the other is an **In**teraction **K**nowledge 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 **T**ransferring the **In**teraction **K**nowledge 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, 31], shape reconstruction [23, 25], and action recognition [15, 29, 53] in computer vision, CV; **2)** grasp generation [24, 52, 61] and motion synthesis [7, 39] 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, 59, 60] or reflective markers [52]. **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 [40], tactile sensors [51] and thermal cameras [3, 5], or computed realistic contact through accurate pose tracking [52]. As contact can provide rich cues to reason the conjoint hand-object poses during interactions, recent methods leveraged contact to help optimize grasps in reconstruction [25, 57] 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 [54] or object [55] pose alone. Hasson *et al*. [23] have proposed the first conjoint hand-object pose estimation methods that brought the renaissance in this area [8,13,21,22,30,31]. 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, 29], imitation learning [43], teleoperation [20], and human-to-robot handover (and vise versa) [49,58] 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

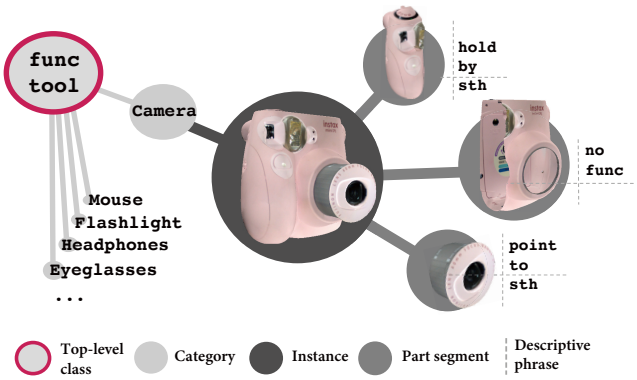

Figure 2. Object Affordance Knowledge Graph.

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 [35] 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 [27, 28], grasp types [11], part segmentation [12, 36], 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 *attribute*s. 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: $\boldsymbol{\mathcal{S}}_c$ and MoCap system: $\boldsymbol{\mathcal{S}}_m$. Our platform is shown in Fig. 3. All the sensors are rigidly mounted on the edges of a $1.5 \times 1.2 \times 1$ 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 *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.

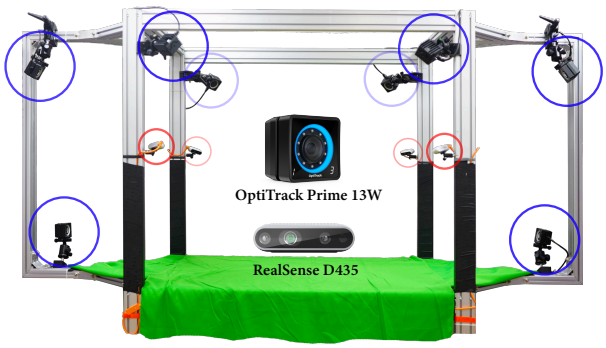

Figure 3. Our data collection platform with 4 RGB-D cameras (red circle) and 8 infrared MoCap cameras (blue circle).

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 $\boldsymbol{\mathcal{S}}_m$. Then, we transform the object pose from $\boldsymbol{\mathcal{S}}_m$ to the MulCam system $\boldsymbol{\mathcal{S}}_c$ 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 [47]. To describe hand pose and geometry in 3D space, we use the MANO hand model [45]. MANO represents an articulated and deformable hand by *pose* $\boldsymbol{\theta} \in \mathbb{R}^{16\times3}$ and *shape* $\boldsymbol{\beta} \in \mathbb{R}^{10}$ parameter. Later in the paper, we denote "hand pose" as the 21 joint positions: $\boldsymbol{P}_h \in \mathbb{R}^{21\times3}$ in the $\boldsymbol{\mathcal{S}}_c$ system. With $\boldsymbol{\theta}$ and $\boldsymbol{\beta}$, we can recover the hand pose $\boldsymbol{P}_h$ and mesh vertices $\boldsymbol{V}_h \in \mathbb{R}^{778\times3}$ through a differentiable MANO layer: $\mathcal{M}(\cdot)$ [23]. Solving the $\boldsymbol{P}_h$ and $\boldsymbol{V}_h$ are formulated as an optimization task minimizing several hand-crafted 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 $\hat{\boldsymbol{p}}_{j,v}$ be the j-th 2D hand keypoint annotation in the v-th view, $\boldsymbol{P}_{h,j}$ be the j-th 3D hand pose estimation, and let $\boldsymbol{T}_v$, $\boldsymbol{K}_v$ be the extrinsic and intrinsic of

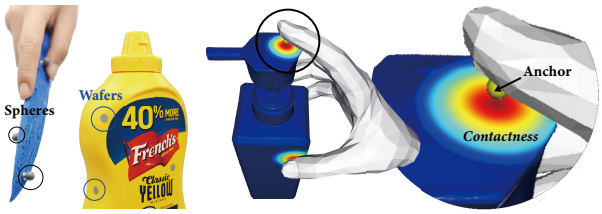

Figure 4. Illustration of the reflective markers (left two) for tracking object motion and the *contactness* that describes physical contact region (right two).

the camera of v-th view, we define the re-projection cost as:

$$E_{\text{repj}} = \frac{1}{\sum w_{j,v}} \sum_{v=1}^{N_v} \sum_{j=1}^{N_j} w_{j,v} \left\| \boldsymbol{K}_v \boldsymbol{T}_v \boldsymbol{P}_{h,j} - \hat{\boldsymbol{p}}_{j,v} \right\|_2^2, \quad (1)$$

where $w_{j,v}$ indicates the visibility of the keypoint $\hat{\boldsymbol{p}}_{j,v}$.

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*. [57]. Based on the efficient contact heuristic in GRAB [52], 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 **T**ransferring the **in**teraction **k**nowledge. *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 [38] 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. (8));

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: $\boldsymbol{\mathcal{O}}_i^s$ and j-th *target* object SDF: $\boldsymbol{\mathcal{O}}_j^t$, we perform linear interpolation between their latent shape vector: $\boldsymbol{o}_i^s$ and $\boldsymbol{o}_j^t$. 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 [32]. 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

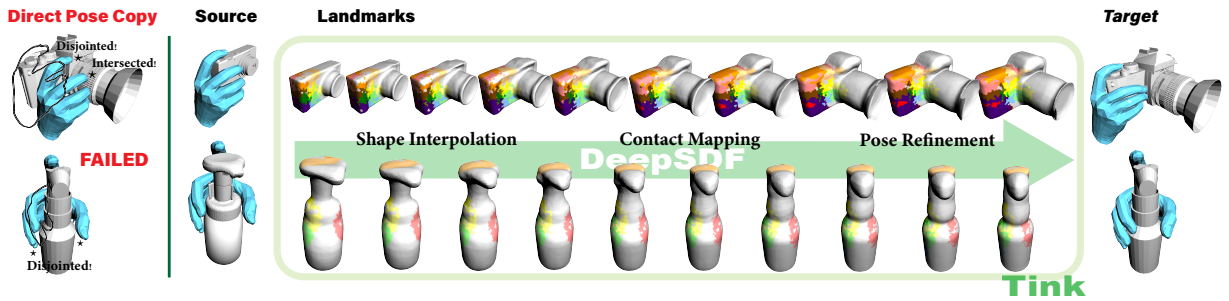

Figure 5. (*Best view in color*) Illustration of our *Tink* pipeline. "Direct pose copy": copying hand pose ($\boldsymbol{\theta}$) in *source* object system to *target* object without refinement. This pose copying usually suffers from unnatural disjointedness or intersection due to shape variant.

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 $\boldsymbol{\theta}$, shape $\boldsymbol{\beta}$, and wrist position $\boldsymbol{P}_{h_0}$ 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 $\boldsymbol{\mathcal{A}} = \{\boldsymbol{A}_i\}_{i=1}^{17}$, the vertices on the object surface that corresponds to the anchor $\boldsymbol{A}_i$ be $\boldsymbol{\mathcal{V}}_h^{(i)} = \{\boldsymbol{V}_{h,j}^{(i)}\}$, and the *contactness* between $\boldsymbol{A}_i$ and $\boldsymbol{V}_{h,j}^{(i)}$ be γ_{ij}. The contact consistency cost is expressed as:

$$E_{\text{consis}} = \frac{1}{\sum \gamma_{ij}} \sum_{\boldsymbol{A}_i} \sum_{\boldsymbol{V}_{h,j}^{(i)}} \gamma_{ij} \left\| \boldsymbol{A}_i - \boldsymbol{V}_{h,j}^{(i)} \right\|_2^2, \tag{2}$$

Direct optimization on the joints' rotations $\boldsymbol{\theta}$ is prone to anatomical abnormality. Hence we adopt the axial adaptations from Yang *et al*. [57] and constrain the rotation axes and angles. Let $\boldsymbol{a}_j$ and ϕ_j be the axial and angular components of the j-th joint's rotation, the $\mathbf{n}_j^t$ and $\mathbf{n}_j^s$ be the pre-defined *twist* and *splay* direction. The anatomical cost is defined as:

$$E_{\text{anat}} = \sum_{j \in \text{all}} \left(\boldsymbol{a}_j \cdot \mathbf{n}_j^t + \max\left((\phi_j - \frac{\pi}{2}), 0 \right) \right) + \sum_{j \notin \text{MCP}} \boldsymbol{a}_j \cdot \mathbf{n}_j^s, \tag{3}$$

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:

$$E_{\text{intp}} = \sum_{\boldsymbol{V}_{h,j}} -\min\left(\text{SDF}_{\boldsymbol{\mathcal{O}}}(\boldsymbol{V}_{h,j}), 0 \right), \tag{4}$$

where the $\text{SDF}_{\boldsymbol{\mathcal{O}}}(\cdot)$ calculates the signed distance value of a 3D hand vertex $\boldsymbol{V}_{h,j}$ to an object's SDF: $\boldsymbol{\mathcal{O}}$ provided at shape interpolation (Sec. 3.3.1). The total optimization problem is:

$$\boldsymbol{V}_h, \boldsymbol{P}_h \longleftarrow \underset{\boldsymbol{\theta}, \boldsymbol{\beta}, \boldsymbol{P}_{h_0}}{\text{argmin}} \left(E_{\text{consis}} + E_{\text{anat}} + E_{\text{intp}} \right), \tag{5}$$

where $\boldsymbol{V}_h, \boldsymbol{P}_h = \mathcal{M}(\boldsymbol{\theta}, \boldsymbol{\beta}) + \boldsymbol{P}_{h_0}$. 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 [50] 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) [52], 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.

Dataset	mod.	reslution	#frame	#subj	#obj	#views	#inten	#intact	real / syn.	label method	intac. inten	obj pose	dynamic intac.	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 [52]	Mesh	–	1624K	10	51	–	4	1.3K	real	marker	✓	✓	✓	✗	✓
DexYCB [10]	RGBD	640 × 480	582K	10	20	8	–	1K	real	crowd	✗	✓	✓	✗	✗
H2O [29]	RGBD	1280 × 720	571K	4	8	5	7	1.8K	real	auto	✓	✓	✓	✗	✗
Ours *OakInk*-Image	RGBD	848 × 480	230K	12	**100**	4	5	1K	real	crowd	✓	✓	✓	✓	✓
Ours *OakInk*-Shape	Mesh	–	–	12	**1,700**	–	5	**49K**	real	*Tink*	✓	✓	✗	✓	✓

Table 1. Comparison of our *OakInk* with the publicly available datasets of hand-object interactions.

Train	**Test**	**MPJPE** (*mm*) ↓
1) HO3D	DexYCB	55.38
2) *OakInk*-Image	DexYCB	44.81
1) & 2) mixture	DexYCB	**39.70**

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. cm^3↓	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 ⋆(:*OakInk*)-Shape. To note: evaluation on PFHAB and HO3D are only conducted on frames of hand grasping objects (minimal distance $\leq$ 5 mm).

4. Tasks and Benchmark Results

We benchmark three existing tasks (Sec. F.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. F.1) [33, 37], 3D hand-object pose estimation (HOPE, Sec. 4.2) [21, 53], and grasp pose generation (GraspGen, Sec. 4.3) [52]. 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 $\boldsymbol{P}_h \in \mathbb{R}^{21\times3}$ and geometry $\boldsymbol{V}_h \in \mathbb{R}^{778\times3}$ 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 three views form the train/val sequences (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 [37], and the other is a hybrid inverse kinematic method: HandTailor [33]. 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*mm*], and mean per vertex position error (**MPVPE**) in wrist-relative system. We show results on **SP0** test set in Tab. 8 and Fig. 6 (top row). More quantitative results on other splits are provided in **Appx**.

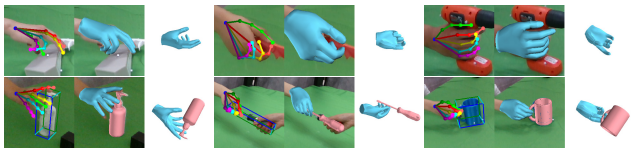

Figure 6. Qualitative results of I2L-MeshNet [37] on HMR task (top row), and Hassen *et al*. [21] on HOPE task (bottom row).

Splits	**Methods**	**MPJPE** (*AUC*)	**MPVPE**
SP0	I2LMeshNet [37]	12.10 (*0.784*)	12.29
	HandTailor [33]	11.20 (*0.884*)	11.75

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

Method	**MPJPE**	**MPCPE** (all category)	**⋆MPCPE** (per category)			
			knife	*lotion*	*mug*	*camera*
Hasson *et al*. [21]	27.26	56.09	68.40	60.70	37.26	68.13
Tekin *et al*. [53]	23.52	52.16	57.29	57.11	35.44	56.87

Table 5. HOPE results in *mm*. ⋆: only list 4 categories.

4.2. Hand-Object Pose Estimation

The HOPE task is to simultaneously estimate the hand pose $\boldsymbol{P}_h$ and the object pose (rotation $\boldsymbol{R}_o \in \mathbf{SO}(3)$, center translation $\boldsymbol{t}_o \in \mathbb{R}^3$) 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*. [53] 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-relative system. We show the test set results in Tab. 5 and Fig. 6 (bottom row).

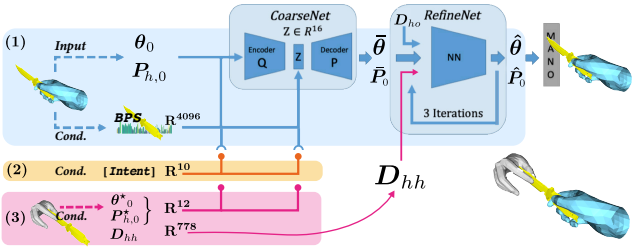

Figure 7. **Network architectures.** (1). GrabNet; (1)+(2): Intent-based interaction generation; (1)+(3): Handover generation.

4.3. Grasp Pose Generation

The GraspGen task is to generate diverse hand poses that interact with a given object shape. Existing GraspGen methods [24,25,52] widely adopted a conditional VAE [48] architecture to this end. As shown in Fig. 7 (1), the model is trained with an object shape (as BPS [41] $\in \mathbb{R}^{4096}$) and its interacting hand pose ($\boldsymbol{\theta_0}$, $\boldsymbol{P}_{h_0}$) 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: $\mathcal{Z}$. Then during the testing, given a test object, the model decodes a hand pose from its embedding space $\mathcal{Z}$. 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 [52], a representative method toward GraspGen. The evaluation metrics of GraspGen consist of four terms. We evaluate 1) penetration depth, 2) solid intersection volume following [57], 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 [24,25,52]. 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.

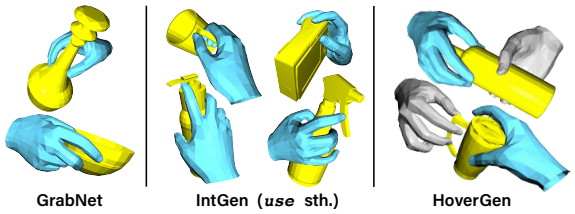

Figure 8. Qualitative results of GrabNet, IntGen and HoverGen on *OakInk*-Shape. (blue: the generated hand; gray: the giver's hand.)

Metrics	GrabNet [52]	IntGen				Hover Gen
		mug	*trigger sprayer*	*camera*	*lotion bottle*	
Penet. Depth. cm↓	0.67	0.45	0.71	1.54	1.57	0.62
Solid Intsec. Vol. cm^3↓	6.60	4.22	9.99	14.32	18.04	6.99
Sim. Disp. Mean cm↓	1.21	0.86	0.69	2.88	2.02	1.30
Sim. Disp. Std cm↓	2.05	1.51	0.81	4.53	2.99	2.03
Percep. score (1,..,5)↑	3.66	3.86	3.93	3.94	3.98	4.03

Table 6. Quantitative results on three generation tasks.

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. 8

[2] Dafni Antotsiou, Guillermo Garcia-Hernando, and Tae-Kyun Kim. Task-oriented hand motion retargeting for dexterous manipulation imitation. In *ECCV Workshops*, 2018. 1

[3] Samarth Brahmbhatt, Cusuh Ham, Charles C. Kemp, and James Hays. ContactDB: Analyzing and Predicting Grasp Contact via Thermal Imaging. In *CVPR*, 2019. 2

[4] Samarth Brahmbhatt, Ankur Handa, James Hays, and Dieter Fox. ContactGrasp: Functional Multi-finger Grasp Synthesis from Contact. In *IROS*, 2019. 3

[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. 1, 2, 7

[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. 2

[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. 2

[8] Zhe Cao, Ilija Radosavovic, Angjoo Kanazawa, and Jitendra Malik. Reconstructing hand-object interactions in the wild. In *ICCV*, 2021. 3

[9] Angel X. Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li 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. 2

[10] Yu-Wei Chao, Wei Yang, Yu Xiang, Pavlo Molchanov, Ankur Handa, Jonathan Tremblay, Yashraj S. Narang, Karl Van Wyk, Umar Iqbal, Stan Birchfield, Jan Kautz, and Dieter Fox. DexYCB: A benchmark for capturing hand grasping of objects. In *CVPR*, 2021. 1, 2, 4, 7

[11] Enric Corona, Albert Pumarola, Guillem Alenya, Francesc Moreno-Noguer, and Grégory Rogez. Ganhand: Predicting human grasp affordances in multi-object scenes. In *CVPR*, 2020. 1, 2, 3, 7

[12] Thanh-Toan Do, Anh Nguyen, and Ian Reid. Affordancenet: An end-to-end deep learning approach for object affordance detection. In *ICRA*, 2018. 3

[13] Bardia Doosti, Shujon Naha, Majid Mirbagheri, and David Crandall. Hope-net: A graph-based model for hand-object pose estimation. In *CVPR*, 2020. 2, 3

[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. 3

[15] 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 *CVPR*, 2018. 1, 2, 3

[16] 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 *CVPR*, 2018. 6, 7

[17] James J Gibson. *The ecological approach to visual perception: classic edition*. Psychology Press, 2014. 3

[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. 5

[19] Shreyas Hampali, Mahdi Rad, Markus Oberweger, and Vincent Lepetit. Honnotate: A method for 3d annotation of hand and object poses. In *CVPR*, 2020. 1, 2, 6, 7

[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. 1, 3

[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. 2, 3, 7

[22] Yana Hasson, Gül Varol, Ivan Laptev, and Cordelia Schmid. Towards unconstrained joint hand-object reconstruction from rgb videos. In *arXiv preprint arXiv:2108.07044*, 2021. 3

[23] 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 *CVPR*, 2019. 1, 2, 3, 4, 7, 8

[24] Hanwen Jiang, Shaowei Liu, Jiashun Wang, and Xiaolong Wang. Hand-object contact consistency reasoning for human grasps generation. In *ICCV*, 2021. 2, 8

[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. 2, 3, 8

[26] Hiroharu Kato, Yoshitaka Ushiku, and Tatsuya Harada. Neural 3d mesh renderer. In *CVPR*, 2018. 12

[27] Mia Kokic, Danica Kragic, and Jeannette Bohg. Learning task-oriented grasping from human activity datasets. *IEEE Robotics and Automation Letters (RAL)*, 2020. 3

[28] Mia Kokic, Johannes A Stork, Joshua A Haustein, and Danica Kragic. Affordance detection for task-specific grasping using deep learning. In *Humanoids*, 2017. 3

[29] Taein Kwon, Bugra Tekin, Jan Stuhmer, Federica Bogo, and Marc Pollefeys. H2O: Two hands manipulating objects for first person interaction recognition. In *ICCV*, 2021. 1, 2, 3, 7

[30] Kailin Li, Lixin Yang, Xinyu Zhan, Jun Lv, Wenqiang Xu, Jiefeng Li, and Cewu Lu. ArtiBoost: Boosting articulated 3d hand-object pose estimation via online exploration and synthesis. In *CVPR*, 2022. 1, 3

[31] Shaowei Liu, Hanwen Jiang, Jiarui Xu, Sifei Liu, and Xiaolong Wang. Semi-supervised 3d hand-object poses estimation with interactions in time. In *CVPR*, 2021. 2, 3

[32] William E Lorensen and Harvey E Cline. Marching cubes: A high resolution 3d surface construction algorithm. *ACM Siggraph Computer Graphics*, 1987. 5

[33] Jun Lv, Wenqiang Xu, Lixin Yang, Sucheng Qian, Chongzhao Mao, and Cewu Lu. HandTailor: Towards high-precision monocular 3d hand recovery. In *BMVC*, 2021. 7, 13

[34] A.T. Miller and P.K. Allen. Graspit! a versatile simulator for robotic grasping. *IEEE Robotics Automation Magazine*, 2004. 1

[35] George A Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 1995. 3

[36] Kaichun Mo, Leonidas Guibas, Mustafa Mukadam, Abhinav Gupta, and Shubham Tulsiani. Where2act: From pixels to actions for articulated 3d objects. In *ICCV*, 2021. 1, 3

[37] Gyeongsik Moon and Kyoung Mu Lee. I2l-meshnet: Image-to-lixel prediction network for accurate 3d human pose and mesh estimation from a single rgb image. In *ECCV*, 2020. 7, 13

[38] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *CVPR*, 2019. 5

[39] Mathis Petrovich, Michael J. Black, and Gül Varol. Action-conditioned 3D human motion synthesis with transformer VAE. In *ICCV*, 2021. 2

[40] Tu-Hoa Pham, Nikolaos Kyriazis, Antonis A Argyros, and Abderrahmane Kheddar. Hand-object contact force estimation from markerless visual tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017. 2

[41] Sergey Prokudin, Christoph Lassner, and Javier Romero. Efficient learning on point clouds with basis point sets. In *ICCV*, 2019. 8

[42] Yuzhe Qin, Yueh-Hua Wu, Shaowei Liu, Hanwen Jiang, Ruihan Yang, Yang Fu, and Xiaolong Wang. DexMV: Imitation learning for dexterous manipulation from human videos. In *arXiv preprint arXiv:2108.05877*, 2021. 1

[43] Ilija Radosavovic, Xiaolong Wang, Lerrel Pinto, and Jitendra Malik. State-only imitation learning for dexterous manipulation. In *IROS*, 2020. 3

[44] Grégory Rogez, James S. Supancic, and Deva Ramanan. Understanding everyday hands in action from rgb-d images. In *ICCV*, 2015. 1

[45] Javier Romero, Dimitrios Tzionas, and Michael J Black. Embodied hands: Modeling and capturing hands and bodies together. *ACM Transactions on Graphics*, 2017. 4, 11

[46] Dandan Shan, Jiaqi Geng, Michelle Shu, and David Fouhey. Understanding human hands in contact at internet scale. In *CVPR*, 2020. 1

[47] Tomas Simon, Hanbyul Joo, Iain Matthews, and Yaser Sheikh. Hand keypoint detection in single images using multiview bootstrapping. In *CVPR*, 2017. 4

[48] Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning structured output representation using deep conditional generative models. In *NIPS*, 2015. 8

[49] Halit Bener Suay and Emrah Akin Sisbot. A position generation algorithm utilizing a biomechanical model for robot-human object handover. In *ICRA*, 2015. 3

[50] Xiao Sun, Bin Xiao, Fangyin Wei, Shuang Liang, and Yichen Wei. Integral human pose regression. In *ECCV*, 2018. 6

[51] Subramanian Sundaram, Petr Kellnhofer, Yunzhu Li, Jun-Yan Zhu, Antonio Torralba, and Wojciech Matusik. Learning the signatures of the human grasp using a scalable tactile glove. *Nature*, 2019. 2

[52] Omid Taheri, Nima Ghorbani, Michael J Black, and Dimitrios Tzionas. GRAB: A dataset of whole-body human grasping of objects. In *ECCV*, 2020. 1, 2, 5, 6, 7, 8, 12

[53] Bugra Tekin, Federica Bogo, and Marc Pollefeys. H+O: Unified egocentric recognition of 3d hand-object poses and interactions. In *CVPR*, 2019. 2, 7

[54] Dimitrios Tzionas, Luca Ballan, Abhilash Srikantha, Pablo Aponte, Marc Pollefeys, and Juergen Gall. Capturing hands in action using discriminative salient points and physics simulation. *International Journal of Computer Vision*, 2016. 3

[55] Dimitrios Tzionas and Juergen Gall. 3d object reconstruction from hand-object interactions. In *ICCV*, 2015. 3

[56] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 2008. 12

[57] Lixin Yang, Xinyu Zhan, Kailin Li, Wenqiang Xu, Jiefeng Li, and Cewu Lu. CPF: Learning a contact potential field to model the hand-object interaction. In *ICCV*, 2021. 2, 5, 6, 8, 12

[58] Wei Yang, Chris Paxton, Maya Cakmak, and Dieter Fox. Human grasp classification for reactive human-to-robot handovers. In *IROS*, 2020. 3

[59] Ruolin Ye, Wenqiang Xu, Zhendong Xue, Tutian Tang, Yanfeng Wang, and Cewu Lu. H2O: A benchmark for visual human-human object handover analysis. In *ICCV*, 2021. 1, 2

[60] Shanxin Yuan, Qianru Ye, Björn Stenger, Siddhant Jain, and Tae-Kyun Kim. Bighand2.2m benchmark: Hand pose dataset and state of the art analysis. In *CVPR*, 2017. 2

[61] Tianqiang Zhu, Rina Wu, Xiangbo Lin, and Yi Sun. Toward human-like grasp: Dexterous grasping via semantic representation of object-hand. In *ICCV*, 2021. 1, 2

OAKINK

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

https://github.com/lixiny/OakInk

Appendices

Contents

A *Oak* base Details;
B Data Annotation Details;
C More Dataset Analysis;
D Implementation: IntGen and HoverGen;
E Perceptual Survey for Generation Tasks;
F Additional Benchmark Results;
- **F.1** Hand Mesh Recovery: Other Splits;
- **F.2** Unseen Out-of-domain Object;
- **F.3** More Visualization;

G Discussion on Personally Identifiable Data;

A. *Oak* Base Details

In this section, we provide the details of the **O**bject **A**ffordance **K**nowledge base (*Oak* base), covering the lists of total 32 categories and 30 *attribute* phrases in Tab. 7.

`maniptool`	knife, screwdriver, hammer, wrench, toothbrush, pen, frying pan, drill, pincer, scissors, stapler, mug, teapot, cup, can, box, bowl, wineglass, cylinder bottle, trigger sprayer, lotion bottle
`functool`	eyeglasses, headphones, binoculars, game controller, lightbulb, camera, flashlight, mouse, phone, apple, banana, donut
Attribute phrases	contain sth, cover sth, pump out sth, cut sth, stab sth, flow in/out sth, tighten sth, loosen sth, clamp sth, brush sth, trigger sth, observe sth, point to sth, shear sth, attach to sth, connect sth, knock sth, spray sth, no function; hold by sth, screwed by sth, unscrewed by sth, pressed by sth, handled by sth, plug by sth, unplug by sth, squeeze by sth, pour out by sth

Table 7. The **categories** and **attribute phrases** in our *Oak* base

B. Data Annotation Details

This section is a supplementary of the Sec. 3.2.3: **Hand Pose and Geometry**. Given the manually labeled 2D hand keypoints, we aim to solve the pose $\boldsymbol{\theta} \in \mathbb{R}^{16\times 3}$, shape $\boldsymbol{\beta} \in \mathbb{R}^{10}$ parameters and the wrist's position $\boldsymbol{P}_{h,0} \in \mathbb{R}^3$ of a 3D hand. These parameters will drive a 3D hand model by a differentiable MANO layer: $\mathcal{M}(\cdot)$ [45]:

$$\boldsymbol{V}_h, \boldsymbol{P}_h = \mathcal{M}(\boldsymbol{\theta}, \boldsymbol{\beta}) + \boldsymbol{P}_{h,0} \tag{6}$$

where $\boldsymbol{P}_h \in \mathbb{R}^{21\times 3}$ is the hand joints' 3D position, and $\boldsymbol{V}_h \in \mathbb{R}^{778\times 3}$ is the hand mesh vertices' 3D position. The objective cost function for solving $\boldsymbol{\theta}$, $\boldsymbol{\beta}$ and $\boldsymbol{P}_{h,0}$ consists of 5 terms.

Reprojection Error. First, we want the 2D projections of the 3D hand joints $\boldsymbol{P}_h$ to match its 2D keypoints annotation $\hat{\boldsymbol{p}}$. Let the subscript j and v be the joint's ID and view's ID, we have the reprojection cost:

$$E_{\text{repj}} = \frac{1}{\sum w_{j,v}} \sum_{v=1}^{4} \sum_{j=1}^{21} w_{j,v} \left\| \boldsymbol{K}_v \boldsymbol{T}_v \boldsymbol{P}_{h,j} - \hat{\boldsymbol{p}}_{j,v} \right\|_2^2 \tag{7}$$

The gradients from E_{repj} will back propagate to $\boldsymbol{P}_h$ and then update the $\boldsymbol{\theta}$, $\boldsymbol{\beta}$ and $\boldsymbol{P}_{h,0}$.

Geometry Consistency. Second, we want the 3D geometry model of hand and object to be consistent with their real-world observation: no interpenetration would occur. Hence, we introduce the second cost function: interpenetration loss. We acquire the object's sign distance field: $\boldsymbol{\mathcal{O}}$ from its scanned model, transform the $\boldsymbol{\mathcal{O}}$'s pose from MoCap system to the camera system, and calculate the sign distance value of a 3D hand vertex $\boldsymbol{V}_{h,i}$ to $\boldsymbol{\mathcal{O}}$. The interpenetration cost penalizes those hand vertices inside the object surface (with negative sign distance values).

$$E_{\text{intp}} = \sum_{\boldsymbol{V}_{h,i}} -\min\Big(\text{SDF}_{\boldsymbol{\mathcal{O}}}(\boldsymbol{V}_{h,i}), 0\Big), \tag{8}$$

The gradients from E_{intp} will back propagate to each $\boldsymbol{V}_{h,i}$ and then update the $\boldsymbol{\theta}$, $\boldsymbol{\beta}$ and $\boldsymbol{P}_{h,0}$.

Silhouette Constraint. Third, we want the contour projection of hand and object models to match the visual cues.

Hence, we introduce a binary silhouette cost. We first acquire the hand and object's binary mask ($\mathcal{B}_h$ and $\mathcal{B}_o$) from the recorded images. This process is automatic. We filter out the background pixels through green-screen and depth image matting. The remaining foreground pixels are the union of $\mathcal{B}_h$ and $\mathcal{B}_o$. Then, we render the 3D hand and object mesh on an image as silhouette and penalize the per-pixel misalignment between the rendered silhouette and the binary mask.

$$E_{\text{sh}} = \sum_{\text{all pix.}} \underbrace{f(\boldsymbol{\mathcal{V}}_o \cup \boldsymbol{\mathcal{V}}_h)}_{\text{detached}} \cap BCE\Big\{f(\boldsymbol{\mathcal{V}}_o \cup \boldsymbol{\mathcal{V}}_h), (\mathcal{B}_h \cup \mathcal{B}_o)\Big\} \tag{9}$$

In this equation, the $f(\cdot)$ is a differentiable rendering function [26]; the $(\boldsymbol{\mathcal{V}}_o \cup \boldsymbol{\mathcal{V}}_h)$ is the composited mesh model of hand $\boldsymbol{\mathcal{V}}_h$ and object $\boldsymbol{\mathcal{V}}_o$; the $f(\boldsymbol{\mathcal{V}}_o \cup \boldsymbol{\mathcal{V}}_h)$ is the rendered silhouette image of hand and object model; the $(\mathcal{B}_h \cup \mathcal{B}_o)$ is the union of hand and object binary mask; and the $BCE\{,\}$ is the binary cross entropy loss function; The gradients from E_{sh} will back propagate to $\boldsymbol{\mathcal{V}}_h$ and then update the $\boldsymbol{\theta}$, $\boldsymbol{\beta}$ and $\boldsymbol{P}_{h,0}$.

Anatomical Constraint. Forth, we want the MANO hand pose to satisfy the anatomical constraints of human hand. Hence, we borrow the axial adaptations from Yang *et al.* [57] and constrain the rotation axes and angles.

$$E_{\text{anat}} = \sum_{j \in \text{all}} \Big(\boldsymbol{a}_j \cdot \mathbf{n}_j^t + \max\big((\phi_j - \frac{\pi}{2}), 0\big)\Big) + \sum_{j \notin \text{MCP}} \boldsymbol{a}_j \cdot \mathbf{n}_j^s, \tag{10}$$

where the $\boldsymbol{a}_j$ and ϕ_j denote the axial and angular components of the j-th joint's rotation, the $\mathbf{n}_j^t$ and $\mathbf{n}_j^s$ are the predefined *twist* and *splay* direction, and"MCP" indicates the five Metacarpal joints. The gradients from E_{anat} will back propagate to each joint's axis-angle and then update the $\boldsymbol{\theta}$.

Temporal Smoothing. The above cost functions can only improve the per-frame precision of 3D hand annotations. However, frame-by-frame smoothness is also critical to improving our annotation quality. Hence, We want the solved 3D hand poses to be continuous in the time domain. We adopt a low-pass filter (*e.g.* Kalman Filters) to post-process the poses $\boldsymbol{\theta}$ and wrist positions $\boldsymbol{P}_{h,0}$ across the entire image sequence.

C. More Dataset Analysis

Hand Pose Distribution. We project the interacting hand poses into an embedded space yield from t-SNE [56]. The poses that transferred from the same *OakInk*-Core pose are painted in the same color. From the box in Fig. 9, we can see that the similar interacting hand poses with different objects are mapped to adjacent in the embedded 2D space. From the circles in Fig. 9, we can conclude that the different grasping types are away from each other.

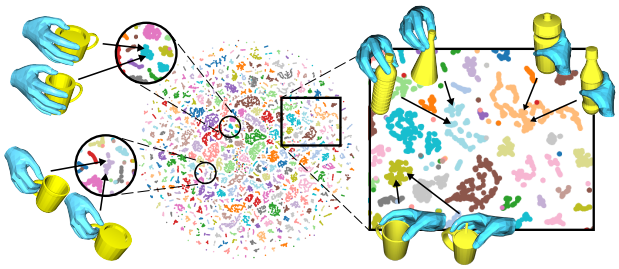

Figure 9. t-SNE embedding of hand poses. We randomly select 20 colors to visualize the clustered poses.

Contact Distribution. We provide the contact heatmaps on example objects that reveal the frequencies of contact among all interactions. Fig. 10 shows such heatmaps on six *Oak* base categories. We see that the "hot" area (red) that denotes the high frequency of contact is consistent with the object affordance we described.

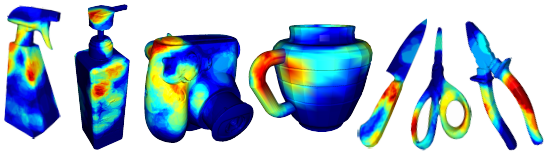

Figure 10. Heatmaps of contact frequency on object surface.

D. Implementation: IntGen and HoverGen

The architecture of IntGen (Fig. 12) and HoverGen (Fig. 13) model are modified from the original GrabNet [52] (Fig. 11) design.

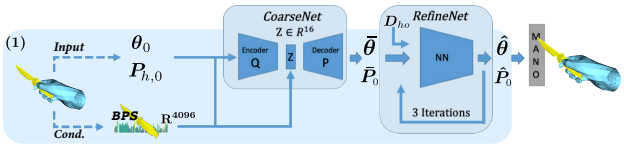

Figure 11. **GrabNet** [52]: the original design.

In the IntGen task, we select three intents: *use*, *hold* and *hand-out*, map the intents' word string to a real-valued word vector, and train the networks with the intent vector as the additional input. During training, poses within different intents will be mapped to different areas in the latent pose space: $\mathcal{Z} \in \mathbb{R}^{16}$. The training loss functions in IntGen are identical to those in GrabNet, including standard conditional VAE losses (KL divergence and weight regularization), mesh reconstruction losses (hand vertices and mesh edge loss), and physical quality losses (penetration and contact loss). We train the IntGen on category-level data in *OakInk*-Shape, including the *mug*, *camera*, *trigger sprayer* and *lotion bottle*. The training process lasts 1,000 epochs, with the mini batch size 32 and initial learning rate

of 1×10^{-3}. The learning rate decays by a factor of 0.5 at every 200 epochs.

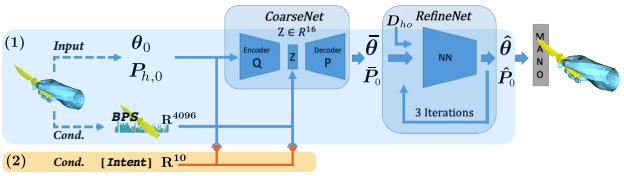

Figure 12. **IntGen**: the intent-based grasp generation network design.

As shown in Fig. 13, in the HoverGen task, we provide the root rotation: $\boldsymbol{\theta}_0^\star$ and root position: $\boldsymbol{P}_{h,0}^\star$ of the giver's hand as the additional inputs for CoarseNet, and the Chamfer distance: $\boldsymbol{D_{hh}} \in \mathbb{R}^{778}$ from the original giver's hand to the predicted receiver's hand as an additional input for RefineNet. As a result, the HoverGen model learns a receiving hand's embedding space, $\mathcal{Z}$, conditioned on the object shape and the giver's hand root pose. At inference time, given an unseen object shape and the giver's hand root pose: $(\boldsymbol{\theta}_0^\star, \boldsymbol{P}_{h,0}^\star)$, we sample a vector from $\mathcal{Z}$ and decode a receiver hand pose to complete a human-to-human handover. The training loss in HoverGen model includes all the losses in IntGen model, plus an L1 loss on the Chamfer distance $\boldsymbol{D_{hh}}$ w.r.t. the ground-truth $\hat{\boldsymbol{D}}_{\boldsymbol{hh}}$, a Chamfer distance from the original giver's hand to the ground-truth receiver's hand. We train the HoverGen model 1,000 epochs with a mini-batch size 256 and an initial learning rate of 1×10^{-3}, decaying a half at every 200 epochs.

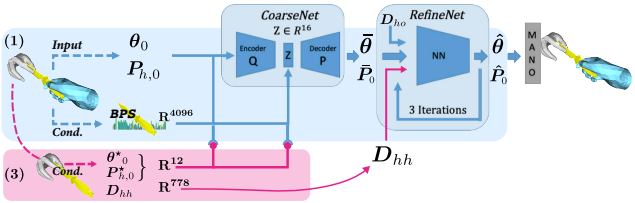

Figure 13. **HoverGen**: the handover generation network design. The giver's hand is paint in **gray** and the receiver's hand in **blue**.

E. Perceptual Survey for Generation Tasks

To investigate the general audience's opinion about the predicted pose of the generation tasks: GrabNet, IntGen, and HoverGen, we conducted three perceptual surveys on the Amazon Mechanical Turk (AMT). In each survey, we show four views of each predicted hand-object interaction and ask the audiences to give their opinion about a statement (*e.g.* "*the hand is interacting naturally with the object*"). The audiences are asked to rate the statement with a 5-level Likert scale ("strongly agree" corresponds to grade 5 and "strongly disagree" corresponds to grade 1). The layout of the perceptual surveys on GrabNet, IntGen, and HoverGen are shown in Fig. 15.

F. Additional Benchmark Results

F.1. Hand Mesh Recovery: Other Splits

Apart from the default split **`SP0`** (split by views) in the main text, we also provide another two data splits and the HMR benchmark results for *OakInk*-Image.

- **`SP1` (subjects split).** (train/val/test: 6/1/5). We split the *OakInk*-Img by subjects. The subjects recorded in the test split will not appear in the train split.
- **`SP2` (objects split).** (train/test: 70%/5%/25%). We split the *OakInk*-Img by objects. The objects that have been grasped in the test split will not appear in the train split.

Splits	Methods	MPJPE↓ (AUC↑)	MPVPE↓
`SP1`	I2L-MeshNet [37]	18.04 (*0.641*)	18.08
	HandTailor [33]	15.72 (*0.792*)	16.31
`SP2`	I2L-MeshNet [37]	15.79 (*0.733*)	15.87
	HandTailor [33]	14.14 (*0.846*)	14.81

Table 8. **HMR results in *mm***. AUC are shown in parentheses.

F.2. Unseen Out-of-domain Object

We refer the objects in *OakInk*-Shape test set as unseen in-domain objects, indicating that they may have similar counterparts included in the training set. In this part, we are also interested in the performance of our generation tasks on the unseen **out-of-domain** objects. We choose the Stanford bunny, a general 3D test model, as an illustrative prototype of **out-of-domain** object. We test the GrabNet and HoverGen model on the Stanford bunny and provide the generated grasps and receiving poses in Fig. 14. Both the GrabNet and HoverGen model are trained on our *OakInk*-Shape training set. The results show that through training on the *OakInk*-Shape, GrabNet and HoverGen can synthesize realistic and prehensile interactions for general objects.

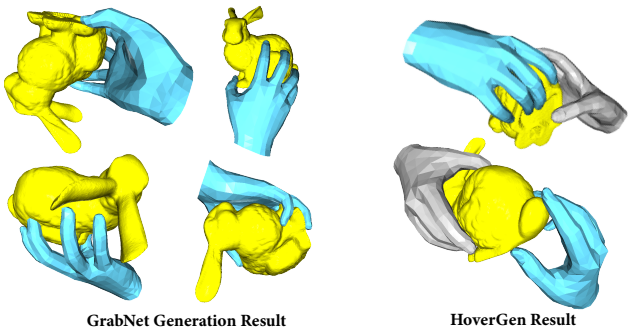

Figure 14. Generation results on **unseen out-of-domain** objects.

F.3. More Visualization

We provide more qualitative results of the benchmark results of HMR task in Fig. 16, HOPE task in Fig. 17, GraspGen (GrabNet) in Fig. 18: top, IntGen in Fig. 18: middle, and HoverGen in Fig. 18: bottom.

G. Discussion on Personally Identifiable Data

We collect hand-object interaction data on 12 human subjects recruited through a third-party crowd-sourcing company. In the collection process, their actions will be recorded in video sequences by the MulCam system. We ensure that the data collecting process meets the ethics requirements through the following announcements:

- The third-party crowd-sourcing company warrants appropriate IRB approval (or equivalent, based on local government requirements) are obtained. The company name and warranties are withheld based on the anonymous submission guidelines.
- All the subjects involved in data collection are required to sign a contract with the third-party crowd-sourcing company, involving permission on the portrait usage, the acknowledgment of data usage, and payment policy. During the data collecting process, all subjects are paid by the hour.
- All the subjects involved in the data collecting process acknowledge that the collected data will only be intended for academic and permitted commercial usages.
- We ensure all the subjects involved in the data collecting process are willing to share the personal-related data, including actions, skin tones, body/hand shapes, *etc*.
- We require all the subjects not to dress in revealing or offensive clothes during the data collection process.
- Upon the release of the dataset, we will desensitize all samples in the dataset by blurring the subjects' faces (if any), tattoos, rings, or any other accessories that may reveal the subjects' identity.

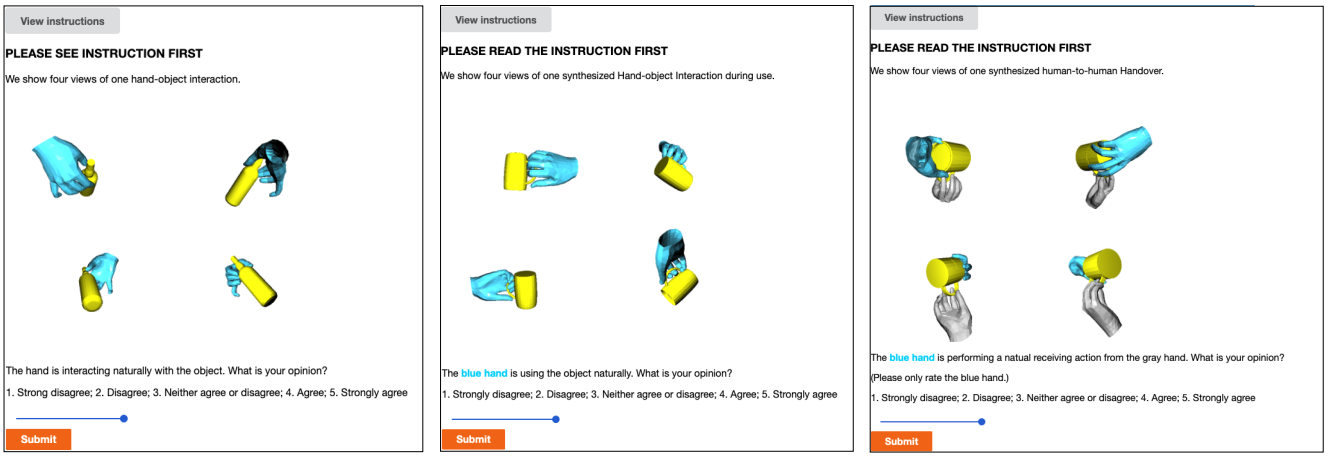

Figure 15. **The layout of three perceptual surveys on AMT**.
Left: GrabNet (statement: *the hand is interacting naturally with the object*);
Middle: IntGen (statement: *the blue hand is using the object naturally*);
Right: HoverGen (statement: *the blue hand is performing a natural receiving action from the gray hand*)

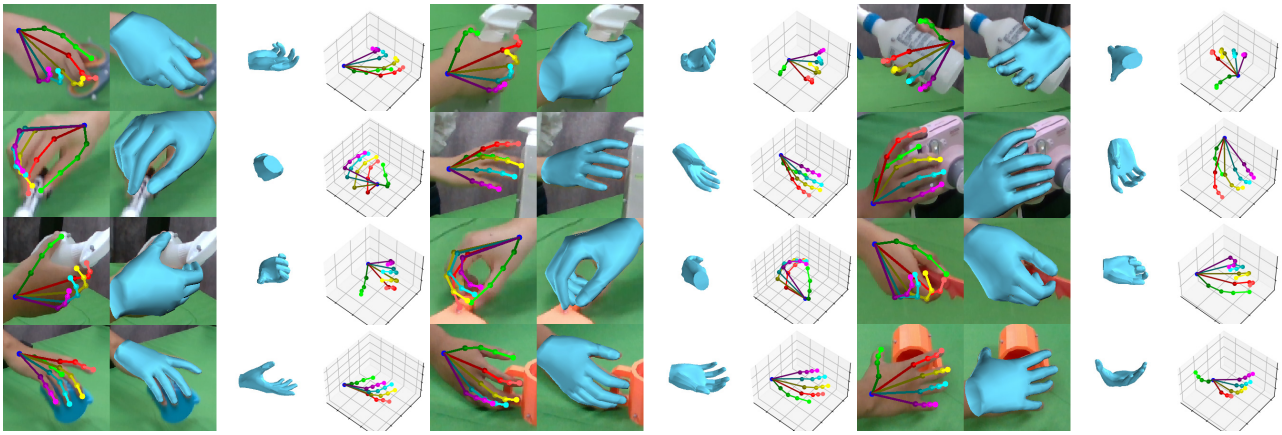

Figure 16. **More qualitative results** on HMR task.

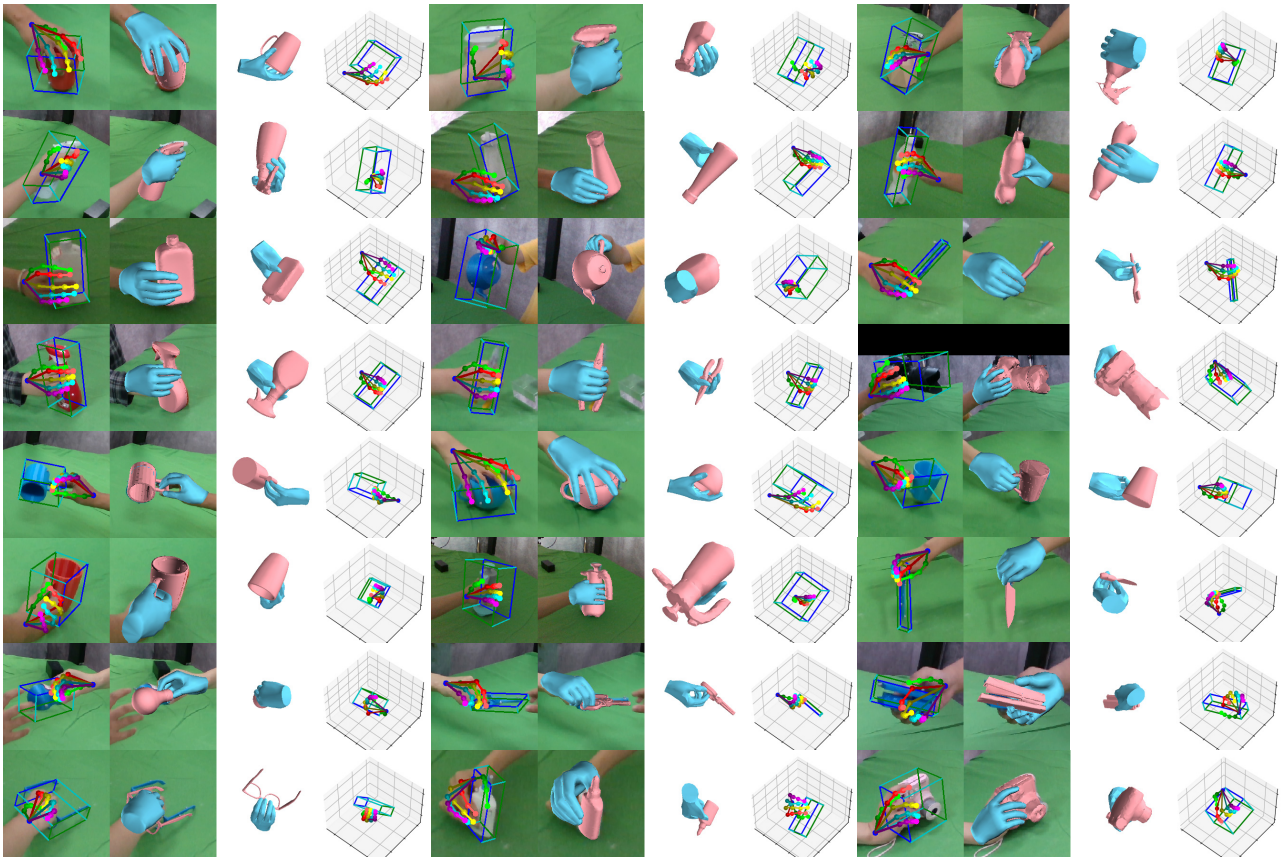

Figure 17. **More qualitative results** on HOPE task.

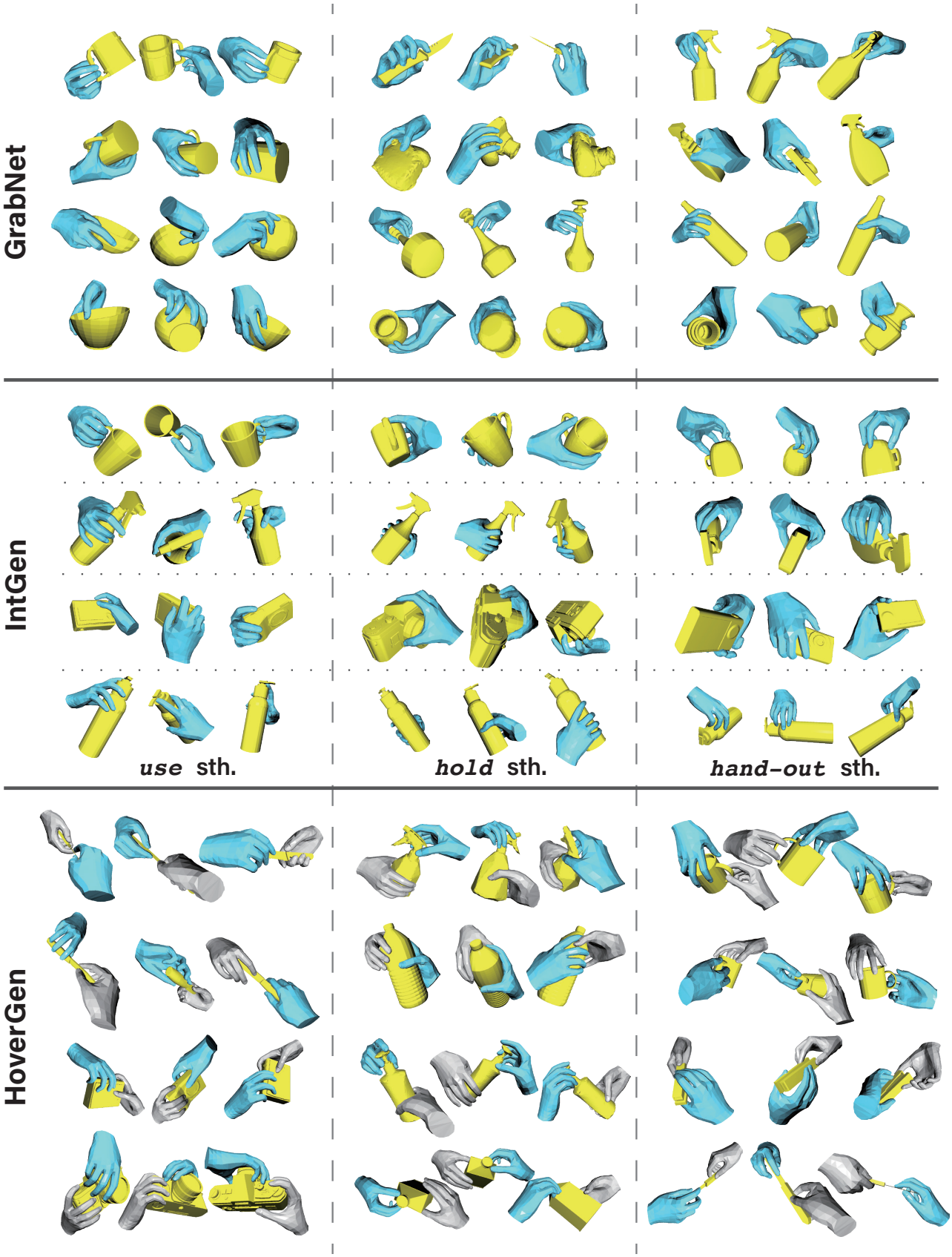

Figure 18. **More qualitative results** on GrabNet, IntGen and HoverGen predictions.