OAKINK2 : A Dataset of Embodied Hands-Object Manipulation in Long-Horizon Complex Task Completion

Xinyu Zhan[1,⋆] Lixin Yang[1,⋆] Yifei Zhao[1] Kangrui Mao[1]
Hanlin Xu[1] Zenan Lin[2] Kailin Li[1] Cewu Lu[1†]
[1]Shanghai Jiao Tong University, [2]South China University of Technology
{kelvin34501, siriusyang, yifei_zhao, karrymao, henry_xu}@sjtu.edu.cn,
auzenanlin@mail.scut.edu.cn, {kailinli, lucewu}@sjtu.edu.cn

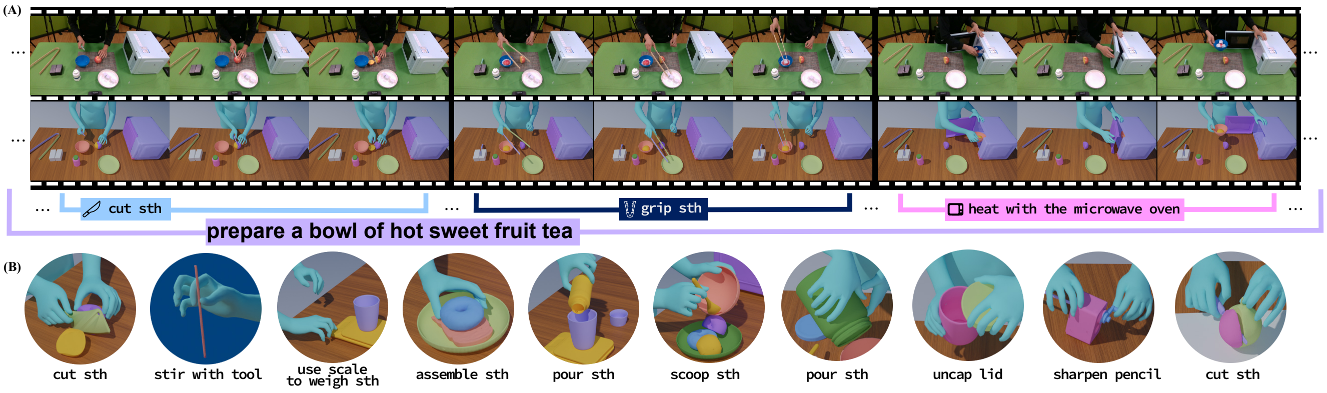


Figure 1. **OAKINK2**: **(A)** shows the captured image streams and the pose annotations, as well as the decomposition of *Combined Tasks* into interdependent *Primitive Tasks*; **(B)** shows several examples of hand-object interactions in *Primitive Tasks*. (Zoom in 🔍 for details)

Abstract

We present **OAKINK2***, a dataset capturing human interaction with multiple objects for further understanding of embodied hand-object manipulation in long-horizon complex task completion.* OAKINK2 *features the demonstrations for* ***Primitive Tasks*** *as minimal interactions fulfilling object affordance attributes, and the demonstrations for* ***Combined Tasks*** *as a combination of Primitive Tasks with certain* ***dependencies****. With the provided multi-view image streams and fine-grained pose annotations for the human body, hands and various interacting objects,* OAKINK2 *supports applications such as hand-object reconstruction; motion synthesis; scene interpretation, complex task target parsing, and human demo replication and combination in the scope of complex manipulation task completion. Our datasets and models will be released to the public.*

1. Introduction

Learning how humans achieve specific task objectives through diverse object manipulation behaviors has been a long-standing challenge. Recent data-driven approaches have made significant progress on this topic, including hand-object pose estimation [1, 4, 10, 18, 20–22, 31, 34], interaction synthesis [8, 14, 26, 50, 55], and action imitation [46, 47]. However, the gap still exists for current methods to achieve a human-level understanding on object manipulation for complex task completion. In particular, humans possess a remarkable capacity to learn from personal experiences, allowing them to interact with specific objects in an appropriate sequence to achieve desired outcomes. This inspires us to focus on embodied hands-object interaction that completes long-horizon complex manipulation tasks.

Tracing prior research, the advancement of data-driven methods is inseparable from the emergence of a series of solid hand-object interaction datasets: ObMan [20], YCBAfford [8], HO3D [17], ContactPose [3], GRAB [49], DexYCB [6], H2O [28], DexMV [47], HOI4D [35], ARCTIC [11], ContactArt [61], AssemblyHands [42], AffordPose [24] and OakInk [57]. A noteworthy example among both types of datasets is OakInk [57]. OakInk analyzed objects' affordance attributes and collected interaction demonstrations based on intents involving these attributes. Despite the success, these datasets do not sufficiently reflect the *orders* and *dependencies* between object affordance fulfillment and the *interactions between objects* to fulfill their affordances. This leads us to break down a complex manipulation task into multiple minimal affordance-fulfilling tasks known as *Primitive Tasks*. Each task is associated with the affordance attributes of the objects and intrinsically contains interaction relationships between multiple objects.

Another form of the data source is human-centric video datasets, represented by EPIC-KITCHENS-100 [9], Ego4D [15], and HAKE [32]. These large-scale datasets are collections of common manipulation tasks and are accompanied by detailed action segment annotations. However, void of 3D grounding for interacting hands and objects impedes the application of these datasets in 3D understanding. Despite this limitation, these video datasets show incorporating task-related annotations to hand-object interaction datasets can assist in understanding and completing complex manipulation tasks in 3D space.

To these ends, we present **OAKINK2**, extending the methodology of OakInk [57] to a further front: capturing long-horizon hand-object interactions in multiple scenarios and analyzing the object affordance dependency and interaction relationships. OAKINK2 is a large-scale hand-object interaction dataset containing human demonstrations for complex manipulation task completion, with multi-view image streams and paired pose annotations in 3D space for interacting hands and objects. OAKINK2 features the annotations of objects' affordance attributes, the demonstrations of *Primitive Tasks* to fulfill these affordances, and the demonstrations of *Combined Tasks* that are composed of *Primitive Tasks* with certain dependencies.

To construct OAKINK2, we first select object clusters from object repositories based on these objects' coexistence and construct corresponding interaction scenarios for these clusters. In each scenario, annotators propose complex manipulation task targets and pick involved object categories. Once the scenarios and task targets are ready, the selected objects are inspected for their affordance attributes using the annotation paradigm described in OakInk. We then design *Primitive Tasks* as *minimal* interactions to fulfill object affordance attributes, establishing mapping relationships between these tasks and affordance attributes. Lastly we acquire *Combined Tasks* to link long-horizon complex manipulation tasks within each scenario to embodied hands-object interactions. As the name implies, they are *Primitive Tasks* combinations that describe the dependencies of the constituent *Primitive Tasks* and constrain the execution orders. With all these formulations in place, we construct OAKINK2, a dataset that reflects the relationship between object affordances in embodied hand-object interactions that complete complex manipulation tasks.

OAKINK2 is composed of four interaction scenarios and 75 objects in total. We invite 9 subjects to interact within the constructed hand-object interaction scenarios and record 627 sequences of real-world bimanual hand-object interaction sequences, where 264 of these sequences are for *Combined Tasks*. These sequences contain 4.01M frames from four different views, including three allocentric views and one egocentric view. The dataset is captured on a data capture platform composed of a MoCap system and a multi-camera system. The pose of the subjects' body, hands, and the objects involved are solved from the captured optical markers. In each captured sequence, the poses and articulation parameters of all objects directly involved in the interaction process are captured and solved, thus the implicit interaction relationships between these objects are also contained in the annotations. In each sequence for *Combined Tasks*, the dependencies between *Primitive Tasks* are expressed with directed acyclic graphs and are included in the annotations with widely used PDDL [38] specifications. OAKINK2 intends to facilitate a variety of tasks: (1) conventional vision tasks like hand-object reconstruction, action recognition, and motion synthesis; and (2) task and motion planning aimed at the completion of intricate manipulation assignments.

To summarize, we present **OAKINK2**, a large-scale dataset of embodied hand interactions with multiple objects for human completion of long-horizon complex manipulation tasks. OAKINK2 features *Primitive Tasks* demonstrations as minimal interactions fulfilling object affordance attributes and *Combined Tasks* demonstrations along with their decomposition into interdependent *Primitive Tasks*.

2. Related Works

Hand-Object Interaction Datasets. The recent research community has witnessed the emergence of numerous datasets on hand-object interactions. Earlier datasets [3, 8, 20] focused on static hand-object interactions with limited diversity. More recent datasets [6, 11, 17, 28, 49, 57] captured dynamic hand-object interactions covering the approaching, placement, and affordance fulfillment processes. These include datasets focused on embodied interactions [11, 49], bimanual interactions [11, 28] and interactions with articulated bodies [11, 61]. We pay particular attention to interaction datasets related to object affordances. [8] expressed affordances in grasp type labels. [3, 11, 49] collected intention labels for interactions. [24, 57] studied object affordance-based hand-object interaction and collected object segmentations and affordance labels. While [24] only annotated static interactions and [57] collected data in intention-oriented manner, our proposed OAKINK2 captures both human demonstrations for minimal interactions fulfilling object affordance attributes as *Primitive Tasks*, and demonstrations for *Combined Tasks* where object affordance attributes fulfilled in specific order constrained by their dependencies.

Another data source of hand-object interactions is large-scale 2D video datasets accompanied by action annotations. This includes EPIC-KITCHEN-100 [9] and EGO4D [15] as large-scale egocentric video datasets, HAKE [32] for object concept learning and HA-ViD [60] with temporal annotations for primitive tasks and atomic actions. Though these datasets provided spatial annotations like labeled 2D boxes, a lack of 3D annotations hinders the direct understanding of the 3D scene in interaction with current learning methods. Recent work like EPIC-Fields [52] explored providing 3D reconstructions algorithmically on existing video datasets. Our proposed OakInk2 provides video demonstrations along with the corresponding 3D grounding information: embodied hands and objects pose and shape annotations. With this 3D information OakInk2 can seamlessly support current learning methods for hand-object interactions while possessing task-level annotations.

Datasets and Benchmarks for long-horizon complex manipulation task completion. Multiple types of datasets and benchmarks for solving long-horizon complex manipulation tasks have been constructed in recent years. Long-horizon tasks in [12, 29, 45] were more inclined to mobile embodied manipulation, *i.e.* navigation first then manipulation. [16, 39, 41, 53] focused on completing manipulation tasks under complex objective constraints, often embedded in text descriptions. Despite the complex constraints and trajectories, these long-horizon manipulation tasks usually only covered one affordance of an object instance.

OakInk2 considers 'long-horizon' as the fulfillment of more than one object affordance attribute in a given manipulation task. As a result, OakInk2 places more emphasis on the fulfillment order of these object affordances and their interdependencies.

3. Construction of OakInk2

OakInk2 features embodied hands-object interaction for long-horizon complex task completion. We first introduce how the interactions are acquired in Sec. 3.1. We then provide information about the data capture setup in Sec. 3.2 and procedures to annotate the dataset in Sec. 3.3. We introduce how execution paths of certain complex *Combined Tasks* are narrated in language in Sec. 3.4.

3.1. Interaction Acquisition

This subsection introduces the three-stage interaction acquisition process (as illustrated in Fig. 2).

3.1.1 Scenario Construction

From the objects we have collected (Fig. 2 **-1.A**), we select four clusters of objects that frequently co-exist in the interaction process and can be accurately tracked in the data capture platform. The selected objects come from four possible sources: (1) ShapeNet [5] models; (2) ContactDB [2] objects; (3) objects included in OakInk [57]; (4) original objects collected from vendors. Based on these selected object clusters (Fig. 2 **-1.B**), we construct four interaction scenarios. Each scenario has its unique characteristic and corresponds to a set of complex manipulation tasks with targets. The scenarios are: (1) kitchen table, (2) study room table, (3) demo chem lab, (4) bathroom table. These scenarios come up with their own task targets for hand-object interaction. We invite annotators (👥) to propose complex manipulation task targets based on the selected cluster of the objects in the scenario (Fig. 2 **-1.C**).

3.1.2 *Primitive Tasks* Acquisition

In the second stage, we ask annotators to evaluate the objects within the scenario and assign affordance attributes to them. Each affordance attribute contains a specific part segmentation and a phrase tuple (Fig. 2 **-2.A**). ***Primitive Tasks*** is designed as *minimal* interactions that fulfill those object affordance attribute. Here *minimal* indicates the designed tasks are required to fully complete the functionality of the object affordance attribute without any redundant interaction process. The affordance attributes of objects in the scenarios previously constructed are attached following the protocol in OakInk [57]. We invite annotators to attach the affordance attributes to the object, where each attribute contains a segmentation of a specific part along with one phrase tuple describing its function. We then concretize *Primitive Tasks* based on these affordance attributes, specifying the starting condition, ending condition, and the in-between hand-object interaction process. For example, a *knife* has been attached to the affordance attribute $\langle$*cut*, *something*$\rangle$, and the part segmentation for its *blade*. Then a *Primitive Task*, *cut something*, is designed to implement this affordance attribute (Fig. 2 **-2.B**). The task requires the subject to move the *blade* of the *knife* to completely pass through the object to be cut so that the separated parts could be detached. The task will be considered incomplete if the *blade* only touches the surface or stops halfway inside the object.

3.1.3 *Combined Tasks* Decomposition

In the third stage, we proceed to design and decompose ***Combined Tasks*** that implement complex manipulation task targets through **long-horizon** hand-object interaction processes. Each manipulation target is paired with specific object instances, combining the starting conditions and instantiated task objectives to form concrete *Combined Tasks* (Fig. 2 **-3.A**). We collect all available affordance attributes of the involved objects and their associated *Primitive Tasks*, leading to a set of available *Primitive Tasks*. We notice that the ordering of *Primitive Tasks* completion is important for the *Combined Task* completion, indicating the existence

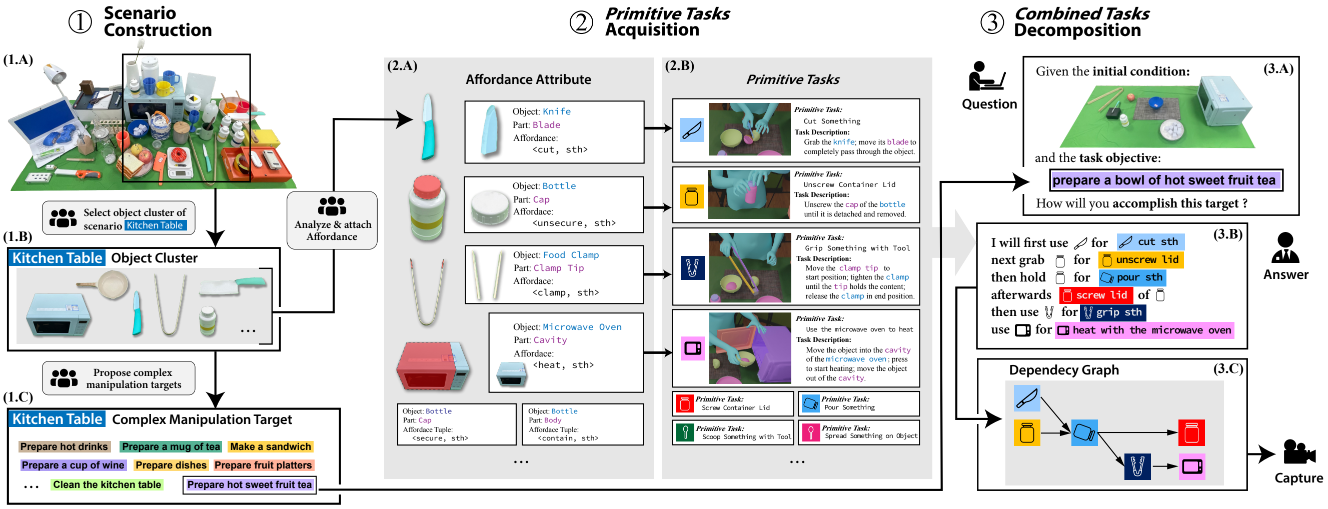


Figure 2. **Three-stage interaction acquisition process.** In ① **Scenario Construction**, annotators (👥) are tasked with creating clusters of objects **(1.A)**, envisioning interaction scenarios **(1.B)**, and suggesting complex manipulations targets **(1.C)**. In ② ***Primitive Tasks* Acquisition**, annotators analysis and attach affordances to the objects **(2.A)**, and propose an *Primitive Task* to encapsulates an affordance **(2.B)**. In ③ ***Combined Tasks* Decomposition**, experts (🧑‍💻) frame the task with a specific outline **(3.A)**, inviting subjects (👤) to map out its execution using the established *Primitive Tasks* **(3.B)**, and breakdown their responses into the dependency graph **(3.C)**.

of *dependencies* between these *Primitive Task* components within the *Combined Task*. We deploy a special capture protocol to acquire the decomposition and dependency information. Before the capture, the expert (🧑‍💻) instantiates the target with specific description, and then ask the subject (👤) to describe the ordering of the completion of available *Primitive Tasks* to complete the *Combined Tasks* (Fig. 2 -**3.B**). The ordering is recorded and inspected by the expert to derive the *dependencies* between the *Primitive Tasks* components. Single *dependency* relationship is represented as a directional link between two tasks, and all *dependency* relationships effective in the *Combined Task* are organized in a directional acyclic graph (Fig. 2 -**3.C**). An example of a *Combined Tasks* is to *prepare a cup of sweet water*. Water in *bottle* needs to be transferred to the target container *teacup*. Then the subject has to *unscrew* the lid of the *bottle* first, then to *pour* the content.

3.2. Capture Setup

The data capture platform contains two major components: the optical MoCap system for collecting pose information and the multi-camera system for capturing visual information within the capture volume. The MoCap system uses 12 Optitrack Prime 13W infrared cameras to track the position of the surface markers on the subject's upper body, left and right hand, and the objects on the table. The MoCap system runs at 120 hz. The multi-camera system consists of 4 commodity RGB cameras, 3 of which are from allocentric views and 1 is from the egocentric view. The multi-camera system runs at 30 fps. We synchronize all sensors and calibrate the relative transforms between these two systems.

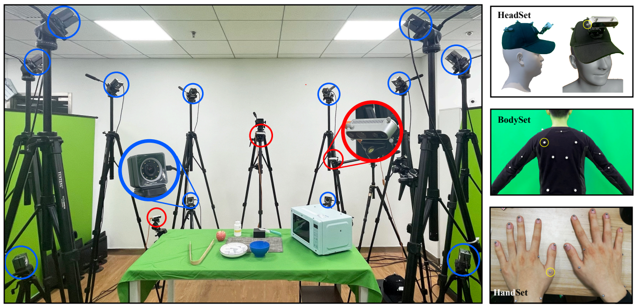


Figure 3. **Capture setup.** The left part shows the captured volume. MoCap cameras are circled in blue and RGB cameras in red. The right part shows in order the head marker-set for the head pose and the egocentric camera, the upper body and hand marker-set.

3.3. Annotation Pipeline

We begin with the cleaned reflective marker positions in the capture volume. The procedure to clean the captured marker positions is described in Sup. Mat. The marker positions are mostly provided by the MoCap system and complemented with the triangulation results of 2D key points of multi-view images when the MoCap system fails to track the markers. Based on these marker positions we obtain the poses of objects and humans.

Object Pose. Poses of rigid bodies are directly solved as long as three or more markers are tracked. The base parts of articulated bodies are handled similarly to rigid bodies. The articulated parts are divided into two categories. If the part is large enough to attach enough markers without blocking the interaction then it will be handled like rigid bodies. Otherwise, a marker is attached to that part. The marker's position is calibrated in the object's canonical coordinate frame, as well as the articulation description (*e.g*. revolution axis or prismatic axis). The parameter of the articulation joint is determined by minimizing the squared difference between the observed marker position and the recovered marker position in the objects' canonical frame.

Human Pose and Surface. The annotation of human pose and surface relies on SMPL-X [44]. Using full-body models provides extra information like arm poses and head poses, and is also helpful to attain realistic hand-wrist articulation as reported in [11]. MANO [48] parameters are fit to the hand part of SMPL-X.

To actually acquire human pose and surface, we employ a two-stage fitting approach inspired by the application of the MoSH++ algorithm in [11, 36, 49]. In the first stage, we use the captured markers when the subject in T-pose to fit the subject's SMPL-X *shape* parameter $\bar{\beta}$ and each marker's location $P_{\mathcal{M}}^{(c)}$ in the canonical space of SMPL-X. From stage one's optimization result, we can determine the correspondence $\mathcal{C}(\cdot)$ from the subject's surface markers to the vertices of the SMPL-X model.

In the second stage, we fit the subject's pose $\theta = \{\theta_t\}$ throughout the interaction process based on the shape $\bar{\beta}$ and marker correspondence $\mathcal{C}(\cdot)$ obtained in the first stage. Then the subject's surface is reconstructed with the SMPL-X body model provided the *pose* and *shape* parameters. Other body representations like MANO are derived from this result. We implement the two-stage fitting pipeline on PyTorch for its automatic differentiation support and use the common gradient descent based algorithm to solve for both stages. Details are provided in Sup. Mat.

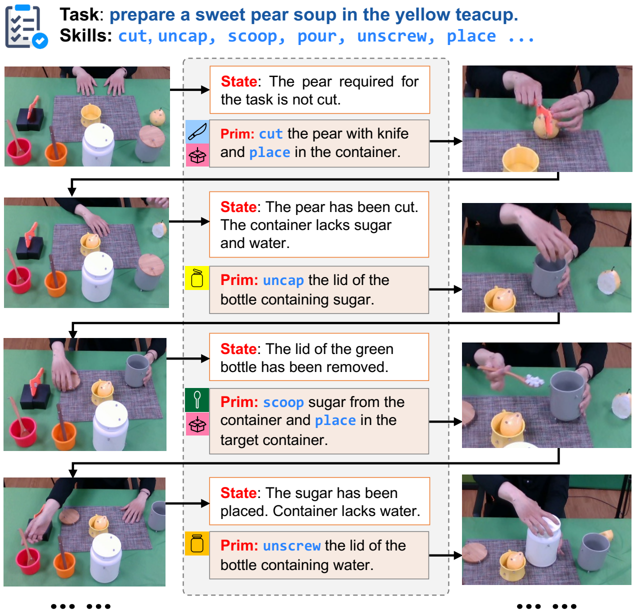
3.4. Narrations of Task Execution Path

We narrate the current state of the scene together with the subsequent *Primitive Tasks* to be performed on a selected set of complex *Combined Tasks* along the execution path of their constituent *Primitive Tasks*. We ask annotators to summarize the remaining tasks that have yet to be completed to achieve the current manipulation target, given the scene after the execution of the previous *Primitive Tasks* and the upcoming *Primitive Tasks* to be executed next. Subsequently, annotators will provide descriptions of the details required for the next step in executing the *Primitive Tasks*. The text obtained from annotators will be refined by GPT-4 [43] and used as the narration for the execution path of complex tasks. The process is illustrated in Fig. 4.

4. The OAKINK2 Dataset

4.1. Data and Annotation

OAKINK2 provides multi-view RGB frames along with the calibrated camera parameters to support vision-based perception methods. We collect four 30 fps image streams with resolution 848 × 480, including three allocentric views and one egocentric view. The annotation of OAKINK2 covers the conventional 3D annotation reflecting the interaction process, and the task-related information reflecting task dependencies. The dataset contains annotations for the body, hands, and objects (with articulation parameters if applicable). For the task-related part, the annotations offered include: the affordance attributes, expressed as part segmentation with attached phrase tuple; *Primitive Tasks* corresponded to the affordance attributes;*Combined Tasks* with the description of task targets, initial conditions, the dependency graph of constituent *Primitive Tasks* and the subject's completion sequence; PDDL specifications [38] based on the *Combined Tasks* descriptions. Fig. 5 illustrates a selection of scanned objects models in OAKINK2, emphasizing the separable or articulation parts of the objects. Fig. 6 shows several scene initiations paired with complex manipulation targets within three types of scenarios. Fig. 7 presents a visualization of the annotation quality in OAKINK2. The body, hands, and objects are blended on the original images for visual inspection. Evaluations of the dataset annotation are detailed in Sup. Mat. Image-based subsets undergo cross-dataset validation, while the shape-based subsets are examined for their physical property integrity.



Task: prepare a sweet pear soup in the yellow teacup.
Skills: cut, uncap, scoop, pour, unscrew, place ...

State: The pear required for the task is not cut.

Prim: cut the pear with knife and place in the container.

State: The pear has been cut. The container lacks sugar and water.

Prim: uncap the lid of the bottle containing sugar.

State: The lid of the green bottle has been removed.

Prim: scoop sugar from the container and place in the target container.

State: The sugar has been placed. Container lacks water.

Prim: unscrew the lid of the bottle containing water.

Figure 4. **Narration of Task Execution Path.** The left column shows the current state of the scene. The center column shows the narration dialog retrieved from annotators. The right column shows the upcoming *Primitive Task* to be executed.

4.2. Dataset Statistics

OAKINK2 sets up four scenarios of hand-object interaction with a total number of 38 long-horizon complex manipu-

Dataset	image mod.	resolution	#frame	#views	#subj	#obj	3D gnd.	real / syn.	label method	hand pose	obj pose	afford. inter.	dynamic inter.	long-horizon	task decomp.
EPIC-KITCHEN-100 [9]	✓	–	–	1	37	–	✗	–	–	✗	✗	✗	✗	✓	✓
EGO4D [15]	✓	–	–	1	931	–	✗	–	–	✗	✗	✗	✗	✓	✓
HA-ViD [60]	✓	1280 × 720	1.5M	3	30	40	✗	–	–	✗	✗	✗	✗	✓	✓
FPHAB [13]	✓	1920 × 1080	105K	1	6	4	✓	real	mocap	✓	✓	✓	✓	✗	✗
ObMan [20]	✓	256 × 256	154K	1	20	3K	✓	syn	simulate	✓	✓	✗	✗	✗	✗
YCBAfford [8]	✓	–	133K	1	1	21	✓	syn	manual	✓	✗	✗	✗	✗	✗
HO3D [17]	✓	640 × 480	78K	1-5	10	10	✓	real	auto	✓	✓	✗	✓	✗	✗
ContactPose [3]	✓	960 × 540	2.99M	3	50	25	✓	real	auto	✓	✓	✓	✗	✗	✗
GRAB [49]	✗	–	1.62M	–	10	51	✓	real	mocap	✓	✓	✓	✓	✗	✗
DexYCB [6]	✓	640 × 480	582K	8	10	20	✓	real	crowd	✓	✓	✗	✓	✗	✗
H2O [28]	✓	1280 × 720	571K	5	4	8	✓	real	auto	✓	✓	✓	✓	✗	✗
HOI4D [35]	✓	1280 × 800	3M	1	9	1000	✓	real	crowd	✓	✓	✓	✓	✓	✗
ARCTIC [11]	✓	2800 × 2000	2.1M	9	10	11	✓	real	mocap	✓	✓	✓	✓	✗	✗
ContactArt [61]	✓	–	332K	–	–	80	✓	real	transfer	✓	✓	✗	✓	✗	✗
AssemblyHands [42]	✓	1920 × 1080	3.03M	12	34	–	✓	real	semi-auto	✓	✗	✓	✓	✓	✓
AffordPose [24]	✗	–	–	–	–	641	✓	syn	manual	✓	✓	✓	✗	✗	✗
OakInk-Image [57]	✓	848 × 480	230K	4	12	100	✓	real	crowd	✓	✓	✓	✓	✗	✗
OakInk-Shape [57]	✗	–	–	–	–	1700	✓	real	transfer	✓	✓	✓	✗	✗	✗
OakInk2	✓	848 × 480	**4.01M**	4	9	75	✓	real	mocap	✓	✓	✓	✓	✓	✓

Table 1. **A cross-comparison among various public datasets.**

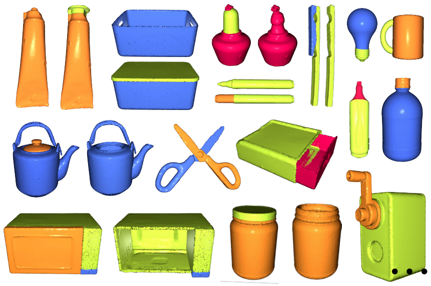


Figure 5. **Scanned object models.**

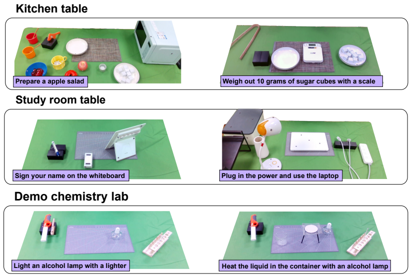

Kitchen table
Prepare a apple salad
Weigh out 10 grams of sugar cubes with a scale
Study room table
Sign your name on the whiteboard
Plug in the power and use the laptop
Demo chemistry lab
Light an alcohol lamp with a lighter
Heat the liquid in the container with an alcohol lamp

Figure 6. **Scene initiation.**

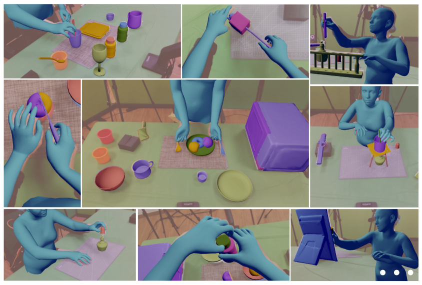


Figure 7. **Quality visualization.**

OakInk2-H-SV	Comprising single-view images for hand reconstruction selected when hands are *visible*.
OakInk2-H-MV	For multi-view hand reconstruction, assembling views where hands is *visible* in most cameras.
OakInk2-HO	For hand-held object reconstruction, including views of *visible* hands with a *grasped* object.
OakInk2-O	For object pose estimation, featuring views with at least one *visible* object.
OakInk2-Grasp	Encompassing frames that depict a *grasped* object.
OakInk2-Motion-Grab	Collating frames that capture the *approach* and *grasp* stages in a *Primitive Task*.
OakInk2-Motion-Task	Amassing frames that document the span from object *grasp* to task *completion*.

Table 2. **Definitions of the task-specific subsets.**

lation targets, which instantiates to 150 *Combined Tasks*. OakInk2 contains in total 75 objects and 51 affordance attributes. 51 affordance attributes map to 51 types of *Primitive Tasks*. OakInk2 contains 627 sequences of bimanual dexterous hand-object interaction in total. 363 of these are for *Primitive Tasks* and 264 for *Combined Tasks*. In total, OakInk2 contains 4.01M image frames.

We compare OakInk2 to multiple existing hand-object interaction datasets in Tab. 1. Compared with existing human-centric video datasets [9, 15] which are larger in scale and more diverse, OakInk2 provides 3D grounding information of embodied hand-object interaction for current learning methods in the form of hand and object pose annotations in 3D space. As for recent hand-object interaction datasets like [11, 35], OakInk2 has comparable scale and contains more frames to cover long-horizon hand-object interactions, and provides extra task decomposition information for complex manipulation task completion. Compared with its previous counterpart OakInk [57], OakInk2 is much larger in scale and covers affordance-based interactions, extending the methodology to a further front.

4.3. Task-specified Subsets

Since OakInk2 is intended to support a variety of tasks, we have curated specialized subsets through diverse sample selection specific to each task type. Segmentation masks for bodies, hands, and objects are rendered per frame, individually and in combination. An instance is deemed *visible* in a frame if the ratio of the combined mask to any individual mask surpasses a set threshold. Similarly, an object is classified as *grasped* if it is maintained within a minimal distance of $\leq 5\,\mathrm{mm}$ to the hands and a height displacement from its initial state of $\geq 5\,\mathrm{mm}$. The subsets are listed in Tab. 2. Refer to the Sup. Mat for details on the features and statistics of all available subsets.

5. Tasks and Benchmarks

We show in this section that OAKINK2 supports reconstruction tasks (Sec. 5.1) and motion generation tasks (Sec. 5.2).

5.1. Reconstruction

Motivation. Recovering 3D information from images remains one of the core tasks of computer vision. Given that OAKINK2 annotate 3D pose and shape information during the completion of complex manipulation tasks, we evaluate the performance of existing reconstruction methods on the collected data. In numerous tasks of reconstructing 3D information from interaction scenes, we have selected the following two tasks to serve as benchmarks: (1) Single-view Hand Mesh Recovery; (2) Multi-view Hand Mesh Recovery. Evaluation on Single-view Hand Mesh Recovery measures the potential capabilities of existing methods in extracting 3D hand information in relative coordinate systems from images or videos on an internet scale. Evaluation on Multi-view Hand Mesh Recovery measures the potential capabilities of existing methods in recovering accurate hand pose and shape in constrained capture scenarios containing multiple pre-calibrated cameras.

Task Definition. In general, the Hand Mesh Recovery task is to estimate the 3D hand poses $\boldsymbol{P}_h$ and shapes $\boldsymbol{V}_h$ during the interaction process from the captured images $\boldsymbol{\mathcal{I}} = \{\mathcal{I}\}$. In single-view settings, the image input $\{\mathcal{I}_v\}$ will only contain one view v from all views, egocentric or allocentric. Then the task is to learn a parameterized model that predicts the distribution: $P_\Phi(\boldsymbol{P}_h, \boldsymbol{V}_h | \boldsymbol{\mathcal{I}} = \{\mathcal{I}_v\})$.

In multi-view settings, the image input will contain multiple views $\boldsymbol{\mathcal{I}} = \{\mathcal{I}_{v_1}, \cdots, \mathcal{I}_{v_n}\}$, together with the camera calibration parameters. Then the task is to learn a parameterized model that predicts: $P_\Phi(\boldsymbol{P}_h, \boldsymbol{V}_h | \{\mathcal{I}_{v_1}, \cdots, \mathcal{I}_{v_n}\})$.

Evaluation metrics and Baselines. We evaluate three categories of metrics: (1) mean per joint position error (**MPJPE**); (2) mean per vertex position error (**MPVPE**); and (3) percentages of correct keypoints under the curve within range (**AUC**). These metrics are evaluated in different frame systems. In single-view settings, these metrics will be evaluated in wrist(root)-relative (**RR**) systems, frame systems after procustes analysis (**PA**) and frame systems of the camera space. In multi-view settings, these metrics will be additionally evaluated in frame systems of the world space. For Single-view Hand Mesh Recovery, we use METRO [33] and Residual Log-likelihood Estimation (RLE) [30] on OAKINK2-H-SV subset as baselines. For Multi-view Hand Mesh Recovery, we use POEM [58] and its variants on OAKINK2-H-MV as baselines. Detailed benchmark results are provided Sup. Mat.

5.2. Complex Task Completion

Motivation. OAKINK2 brings in a new aspect – decomposing Complex *Combined tasks* into paths of *Primitives*. Each primitive is associated with a diverse array of image-textual descriptions, a feature that greatly facilitates the inverse process of Complex Task Completion (CTC). CTC involves generating complex manipulation sequences based on textual instruction. To effectively accomplish the CTC task, OakInk2 supplies annotations for each phase, empowering neural networks to adeptly interpret scenes, accurately localize objects, parse Complex *Combined Tasks* into *Primitives*, and replicate human demonstrations corresponding to each *Primitive*. CTC has immediate applications in AR and human-robot interaction, offering novel capabilities such as the automated scripting of digital human behaviors and providing adaptive assistance in household activities, enriching user convenience.

Task Definition. The Complex Task Completion task is to generate human motion trajectories based on a textual description of the scene $\mathbf{w}_s$ and the complex manipulation task objective $\mathbf{w}_g$, involved object models $\boldsymbol{\mathcal{M}}$ and their poses $\boldsymbol{T}_m$ in the initial scene, as well as text $\{\mathbf{w}_o\}$ describing the state of each object in the initial scene. The goal is to generate a human motion trajectory that can accomplish the task objective given the provided dependency constraints provided in OAKINK2.

End-to-end generation in this particular setup exerts great challenges that surpass the capabilities of current learning methods. The recent breakthroughs in foundation models [7, 23, 27, 40] including Vision Language Models (VLM) and open-vocabulary vision models allow us to utilize them as oracles, indicating that certain parts in the generation process could be assumed to be solved by these oracles. Here we assume the motion trajectory of objects from their source locations to target locations for task completion can be derived from the oracles. Based on this assumption, we adopt a multi-staged problem modeling approach. Specifically, we follow these two stages:

(1) Utilize demonstration trajectories from the primitive tasks provided by OAKINK2. For each primitive task, we learn a trajectory generation model and formulate it into a unified API format.

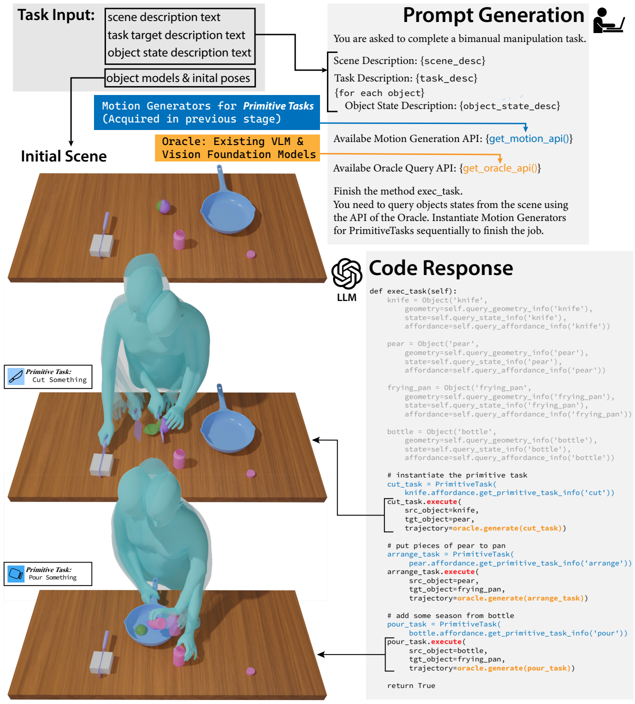
(2) Employ the retrieve and compose paradigm and leverage the task planning abilities of a pre-existing large language model (LLM). This enables us to generate query codes for object locations and attributes and execution codes of the pre-learned primitive tasks.

By combining these two stages as subtasks, we aim to find a potential approach to address the complexities involved in generating human motion trajectories for complex task completion.

Motion Generation. This subtask is to learn a *Primitive Task*-specific human motion generation model based on demonstrations included in OAKINK2. Given a *Primitive Task* $\mathbf{w}_p$, we assume that both the object m's starting position $\boldsymbol{T}^{(0)}$ and trajectory $\{\boldsymbol{T}^{(i)}\}$ during the interaction process are known. Then our objective is to generate a sequence of human poses $\{\boldsymbol{P}_h^{(j)}\}$ in such a way that the human approaches and grabs the object at its starting position, and proceeds to move the object along its trajectory to the final position.

We further split the subtask into three substages. In the first substage, a static grasp $\boldsymbol{P}_h^{(j_g)}$ is generated based on the object's initial pose $\{\boldsymbol{T}^{(0)}\}$ in its trajectory. Subsequently in the second substage, a human motion trajectory $\{\boldsymbol{P}_h^{(0)}, \cdots, \boldsymbol{P}_h^{(j_g)}\}$ is generated to reach the object in its initial location. In the final substage, we generate a human motion trajectory $\{\boldsymbol{P}_h^{(j_g+1)}, \cdots, \boldsymbol{P}_h^{(j_\text{end})}\}$ sufficient to complete the *Primitive Task* in accordance with the object's movement trajectory. By merging the trajectories from the second and third substages into $\{\boldsymbol{P}_h^{(0)}, \cdots, \boldsymbol{P}_h^{(j_g)}, \cdots, \boldsymbol{P}_h^{(j_\text{end})}\}$, we consequently generate the desired human posture trajectory that has completed the *Primitive Task*. We adopt the GNet and MNet in GOAL [50] model for the first and second substage, and a modified version in the final substage. We evaluate contact ratio (CR) and solid intersection volume (SIV) on first substage results and evaluate motion smoothness with Power Spectrum KL divergence of joints (PSKL-J). For the network frameworks and benchmark results, we refer to the Sup. Mat.

Primitive Planning. In this stage we leverage the task planning ability of GPT-4 [43] to generate programs that can be executed to generate multiple trajectories for underlying *Primitive Tasks* completion. We first embed the scene description $\mathbf{w}_g$ and each object's description $\{\mathbf{w}_o\}$ into the prompt based on manually designed templates, as well as the predefined function APIs including the oracles that detect object positions and synthesis object trajectories and the motion generators for *Primitive Tasks*. GPT-4 will respond to the prompt, and then we use a symbolic checker built upon the dependency information for *Primitive Task* completion in OAKINK2 to test whether the generated program completes the task without violation of constraints. If a successful program is acquired, it is executed to call in sequel multiple *Primitive Task* motion generators. These trajectories are connected by essential interpolation to connect the end states of the previous trajectory to the initial states of the next trajectory. This process is illustrated in Fig. 8.



You are asked to complete a bimanual manipulation task.

Scene Description: {scene_desc}
Task Description: {task_desc}
{for each object}
  Object State Description: {object_state_desc}

Availabe Motion Generation API: {get_motion_api()}

Availabe Oracle Query API: {get_oracle_api()}

Finish the method exec_task.
You need to query objects states from the scene using the API of the Oracle. Instantiate Motion Generators for PrimitiveTasks sequentially to finish the job.

```
def exec_task(self):
    knife = Object('knife',
        geometry=self.query_geometry_info('knife'),
        state=self.query_state_info('knife'),
        affordance=self.query_affordance_info('knife'))

    pear = Object('pear',
        geometry=self.query_geometry_info('pear'),
        state=self.query_state_info('pear'),
        affordance=self.query_affordance_info('pear'))

    frying_pan = Object('frying_pan',
        geometry=self.query_geometry_info('frying_pan'),
        state=self.query_state_info('frying_pan'),
        affordance=self.query_affordance_info('frying_pan'))

    bottle = Object('bottle',
        geometry=self.query_geometry_info('bottle'),
        state=self.query_state_info('bottle'),
        affordance=self.query_affordance_info('bottle'))

    # instantiate the primitive task
    cut_task = PrimitiveTask(
        knife.affordance.get_primitive_task_info('cut'))
    cut_task.execute(
        src_object=knife,
        tgt_object=pear,
        trajectory=oracle.generate(cut_task))

    # put pieces of pear to pan
    arrange_task = PrimitiveTask(
         pear.affordance.get_primitive_task_info('arrange'))
    arrange_task.execute(
         src_object=pear,
         tgt_object=frying_pan,
         trajectory=oracle.generate(arrange_task))

    # add some season from bottle
    pour_task = PrimitiveTask(
         bottle.affordance.get_primitive_task_info('pour'))
    pour_task.execute(
         src_object=bottle,
         tgt_object=frying_pan,
         trajectory=oracle.generate(pour_task))

    return True
```

Figure 8. **The diagram of Complex Task Completion.** An initial input populates a predefined template, generating a targeted prompt. GPT-4 responds with code that delineates the program's execution path. Within this code, blue snippet indicate motion generators for primitive tasks; the orange snippet marks the oracle that predicts object trajectories. A symbolic checker validates the code and generates human movements. These actions are integrated to produce the end outcome.

6. Future Works

OAKINK2 is a dataset packing a variety of hand-object interactions for human completion of intricate, long-horizon complex manipulation tasks. OAKINK2 incorporates *Primitive Tasks* demonstrations, characterized as minimal interactions that satisfy object affordance attributes, and *Combined Tasks* demonstrations, which also include their decomposition into interdependent *Primitive Tasks*.

First, we expect OAKINK2 to support large-scale language-manipulation pretraining [25, 59], improving the performance of numerous oracles involved in Complex Task Completion. In the longer term, we expect OAKINK2 can potentially support learning frameworks capable of end-to-end text-to-manipulation generation.

Second, OAKINK2 can empower various embodied manipulation tasks in the future by retargeting the collected demonstrations for *Primitive Tasks* to heterogeneous hands and platforms as [19, 46, 47, 54, 56] implied. The interaction scenarios constructed in OAKINK2 can also be transferred and integrated into existing simulation environments [37, 51] to support embodied learning of complex *Combined Tasks* completion.

References

[1] Ahmed Tawfik Aboukhadra, Jameel Malik, Ahmed Elhayek, Nadia Robertini, and Didier Stricker. THOR-Net: End-to-end graformer-based realistic two hands and object reconstruction with self-supervision. In *Winter Conference on Applications of Computer Vision (WACV)*, 2023. 1

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

[3] 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 *European Conference on Computer Vision (ECCV)*, 2020. 1, 2, 6

[4] Zhe Cao, Ilija Radosavovic, Angjoo Kanazawa, and Jitendra Malik. Reconstructing hand-object interactions in the wild. In *International Conference on Computer Vision (ICCV)*, 2021. 1

[5] 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. 3

[6] 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 *Computer Vision and Pattern Recognition (CVPR)*, 2021. 1, 2, 6

[7] Ho Kei Cheng and Alexander G Schwing. Xmem: Long-term video object segmentation with an atkinson-shiffrin memory model. In *European Conference on Computer Vision (ECCV)*, 2022. 7

[8] Enric Corona, Albert Pumarola, Guillem Alenya, Francesc Moreno-Noguer, and Grégory Rogez. Ganhand: Predicting human grasp affordances in multi-object scenes. In *Computer Vision and Pattern Recognition (CVPR)*, 2020. 1, 2, 6

[9] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Antonino Furnari, Evangelos Kazakos, Jian Ma, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, et al. Rescaling egocentric vision: Collection, pipeline and challenges for epic-kitchens-100. *International Journal of Computer Vision*, 2022. 2, 3, 6

[10] Bardia Doosti, Shujon Naha, Majid Mirbagheri, and David Crandall. HOPE-Net: A graph-based model for hand-object pose estimation. In *Computer Vision and Pattern Recognition (CVPR)*, 2020. 1

[11] Zicong Fan, Omid Taheri, Dimitrios Tzionas, Muhammed Kocabas, Manuel Kaufmann, Michael J Black, and Otmar Hilliges. ARCTIC: A dataset for dexterous bimanual hand-object manipulation. In *Computer Vision and Pattern Recognition (CVPR)*, 2023. 1, 2, 5, 6

[12] Haoyuan Fu, Wenqiang Xu, Han Xue, Huinan Yang, Ruolin Ye, Yongxi Huang, Zhendong Xue, Yanfeng Wang, and Cewu Lu. Rfuniverse: A physics-based action-centric interactive environment for everyday household tasks. *arXiv preprint arXiv:2202.00199*, 2022. 3

[13] 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 *Computer Vision and Pattern Recognition (CVPR)*, 2018. 6

[14] Anindita Ghosh, Rishabh Dabral, Vladislav Golyanik, Christian Theobalt, and Philipp Slusallek. IMoS: Intent-driven full-body motion synthesis for human-object interactions. In *Computer Graphics Forum*, 2023. 1

[15] Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, et al. Ego4d: Around the world in 3,000 hours of egocentric video. In *Computer Vision and Pattern Recognition (CVPR)*, 2022. 2, 3, 6

[16] Jiayuan Gu, Fanbo Xiang, Xuanlin Li, Zhan Ling, Xiqiang Liu, Tongzhou Mu, Yihe Tang, Stone Tao, Xinyue Wei, Yunchao Yao, et al. Maniskill2: A unified benchmark for generalizable manipulation skills. *arXiv preprint arXiv:2302.04659*, 2023. 3

[17] Shreyas Hampali, Mahdi Rad, Markus Oberweger, and Vincent Lepetit. Honnotate: A method for 3d annotation of hand and object poses. In *Computer Vision and Pattern Recognition (CVPR)*, 2020. 1, 2, 6

[18] Shreyas Hampali, Sayan Deb Sarkar, Mahdi Rad, and Vincent Lepetit. Keypoint transformer: Solving joint identification in challenging hands and object interactions for accurate 3d pose estimation. In *Computer Vision and Pattern Recognition (CVPR)*, 2022. 1

[19] Ankur Handa, Karl Van Wyk, Wei Yang, Jacky Liang, Yu-Wei Chao, Qian Wan, Stan Birchfield, Nathan Ratliff, and Dieter Fox. Dexpilot: Vision-based teleoperation of dexterous robotic hand-arm system. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9164–9170. IEEE, 2020. 8

[20] 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 *Computer Vision and Pattern Recognition (CVPR)*, 2019. 1, 2, 6

[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 *Computer Vision and Pattern Recognition (CVPR)*, 2020.

[22] Yana Hasson, Gül Varol, Ivan Laptev, and Cordelia Schmid. Towards unconstrained joint hand-object reconstruction from rgb videos. In *International Conference on 3D Vision (3DV)*, 2021. 1

[23] Wenlong Huang, Chen Wang, Ruohan Zhang, Yunzhu Li, Jiajun Wu, and Li Fei-Fei. Voxposer: Composable 3d value maps for robotic manipulation with language models. *arXiv preprint arXiv:2307.05973*, 2023. 7

[24] Juntao Jian, Xiuping Liu, Manyi Li, Ruizhen Hu, and Jian Liu. AffordPose: A large-scale dataset of hand-object interactions with affordance-driven hand pose. *arXiv preprint arXiv:2309.08942*, 2023. 1, 2, 6

[25] Biao Jiang, Xin Chen, Wen Liu, Jingyi Yu, Gang Yu, and Tao Chen. MotionGPT: Human motion as a foreign language. *arXiv preprint arXiv:2306.14795*, 2023. 8

[26] Hanwen Jiang, Shaowei Liu, Jiashun Wang, and Xiaolong Wang. Hand-object contact consistency reasoning for human grasps generation. In *International Conference on Computer Vision (ICCV)*, 2021. 1

[27] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. *arXiv preprint arXiv:2304.02643*, 2023. 7

[28] Taein Kwon, Bugra Tekin, Jan Stuhmer, Federica Bogo, and Marc Pollefeys. H2O: Two hands manipulating objects for first person interaction recognition. In *International Conference on Computer Vision (ICCV)*, 2021. 1, 2, 6

[29] Chengshu Li, Ruohan Zhang, Josiah Wong, Cem Gokmen, Sanjana Srivastava, Roberto Martín-Martín, Chen Wang, Gabrael Levine, Michael Lingelbach, Jiankai Sun, et al. Behavior-1k: A benchmark for embodied ai with 1,000 everyday activities and realistic simulation. In *Conference on Robot Learning*, pages 80–93. PMLR, 2023. 3

[30] Jiefeng Li, Siyuan Bian, Ailing Zeng, Can Wang, Bo Pang, Wentao Liu, and Cewu Lu. Human pose regression with residual log-likelihood estimation. In *International Conference on Computer Vision (ICCV)*, 2021. 7

[31] 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 *Computer Vision and Pattern Recognition (CVPR)*, 2022. 1

[32] Yong-Lu Li, Xinpeng Liu, Xiaoqian Wu, Yizhuo Li, Zuoyu Qiu, Liang Xu, Yue Xu, Hao-Shu Fang, and Cewu Lu. HAKE: a knowledge engine foundation for human activity understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2022. 2, 3

[33] Kevin Lin, Lijuan Wang, and Zicheng Liu. End-to-end human pose and mesh reconstruction with transformers. In *Computer Vision and Pattern Recognition (CVPR)*, 2021. 7

[34] Shaowei Liu, Hanwen Jiang, Jiarui Xu, Sifei Liu, and Xiaolong Wang. Semi-supervised 3d hand-object poses estimation with interactions in time. In *Computer Vision and Pattern Recognition (CVPR)*, 2021. 1

[35] Yunze Liu, Yun Liu, Che Jiang, Kangbo Lyu, Weikang Wan, Hao Shen, Boqiang Liang, Zhoujie Fu, He Wang, and Li Yi. HOI4D: A 4d egocentric dataset for category-level human-object interaction. In *Computer Vision and Pattern Recognition (CVPR)*, 2022. 1, 6

[36] Naureen Mahmood, Nima Ghorbani, Nikolaus F Troje, Gerard Pons-Moll, and Michael J Black. AMASS: Archive of motion capture as surface shapes. In *Computer Vision and Pattern Recognition (CVPR)*, 2019. 5

[37] Viktor Makoviychuk, Lukasz Wawrzyniak, Yunrong Guo, Michelle Lu, Kier Storey, Miles Macklin, David Hoeller, Nikita Rudin, Arthur Allshire, Ankur Handa, et al. Isaac gym: High performance gpu-based physics simulation for robot learning. *arXiv preprint arXiv:2108.10470*, 2021. 8

[38] Drew McDermott, Malik Ghallab, Adele Howe, Craig Knoblock, Ashwin Ram, Manuela Veloso, Daniel Weld, and David Wilkins. PDDL - the planning domain definition language. 1998. 2, 5

[39] Oier Mees, Lukas Hermann, Erick Rosete-Beas, and Wolfram Burgard. Calvin: A benchmark for language-conditioned policy learning for long-horizon robot manipulation tasks. *IEEE Robotics and Automation Letters*, 2022. 3

[40] Matthias Minderer, Alexey Gritsenko, Austin Stone, Maxim Neumann, Dirk Weissenborn, Alexey Dosovitskiy, Aravindh Mahendran, Anurag Arnab, Mostafa Dehghani, Zhuoran Shen, et al. Simple open-vocabulary object detection. In *European Conference on Computer Vision (ECCV)*, 2022. 7

[41] Tongzhou Mu, Zhan Ling, Fanbo Xiang, Derek Yang, Xuanlin Li, Stone Tao, Zhiao Huang, Zhiwei Jia, and Hao Su. Maniskill: Generalizable manipulation skill benchmark with large-scale demonstrations. *arXiv preprint arXiv:2107.14483*, 2021. 3

[42] Takehiko Ohkawa, Kun He, Fadime Sener, Tomas Hodan, Luan Tran, and Cem Keskin. AssemblyHands: Towards egocentric activity understanding via 3d hand pose estimation. In *Computer Vision and Pattern Recognition (CVPR)*, 2023. 1, 6

[43] OpenAI. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 5, 8

[44] Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed AA Osman, Dimitrios Tzionas, and Michael J Black. Expressive body capture: 3d hands, face, and body from a single image. In *Computer Vision and Pattern Recognition (CVPR)*, 2019. 5

[45] Xavier Puig, Eric Undersander, Andrew Szot, Mikael Dallaire Cote, Tsung-Yen Yang, Ruslan Partsey, Ruta Desai, Alexander William Clegg, Michal Hlavac, So Yeon Min, et al. Habitat 3.0: A co-habitat for humans, avatars and robots. *arXiv preprint arXiv:2310.13724*, 2023. 3

[46] Yuzhe Qin, Hao Su, and Xiaolong Wang. From one hand to multiple hands: Imitation learning for dexterous manipulation from single-camera teleoperation. *IEEE Robotics and Automation Letters*, 2022. 1, 8

[47] 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 *European Conference on Computer Vision (ECCV)*, 2022. 1, 8

[48] Javier Romero, Dimitris Tzionas, and Michael J Black. Embodied hands: Modeling and capturing hands and bodies together. *ACM Transactions on Graphics*, 36(6), 2017. 5

[49] Omid Taheri, Nima Ghorbani, Michael J Black, and Dimitrios Tzionas. GRAB: A dataset of whole-body human grasping of objects. In *European Conference on Computer Vision (ECCV)*, 2020. 1, 2, 5, 6

[50] Omid Taheri, Vasileios Choutas, Michael J Black, and Dimitrios Tzionas. GOAL: Generating 4d whole-body motion

for hand-object grasping. In *Computer Vision and Pattern Recognition (CVPR)*, 2022. 1, 8

[51] Emanuel Todorov, Tom Erez, and Yuval Tassa. MuJoCo: A physics engine for model-based control. *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012. 8

[52] Vadim Tschernezki, Ahmad Darkhalil, Zhifan Zhu, David Fouhey, Iro Laina, Diane Larlus, Dima Damen, and Andrea Vedaldi. EPIC Fields: Marrying 3d geometry and video understanding. *arXiv preprint arXiv:2306.08731*, 2023. 3

[53] Homer Walke, Kevin Black, Abraham Lee, Moo Jin Kim, Max Du, Chongyi Zheng, Tony Zhao, Philippe Hansen-Estruch, Quan Vuong, Andre He, et al. Bridgedata v2: A dataset for robot learning at scale. *arXiv preprint arXiv:2308.12952*, 2023. 3

[54] Weikang Wan, Haoran Geng, Yun Liu, Zikang Shan, Yaodong Yang, Li Yi, and He Wang. UniDexGrasp++: Improving dexterous grasping policy learning via geometry-aware curriculum and iterative generalist-specialist learning. *arXiv preprint arXiv:2304.00464*, 2023. 8

[55] Yan Wu, Jiahao Wang, Yan Zhang, Siwei Zhang, Otmar Hilliges, Fisher Yu, and Siyu Tang. SAGA: Stochastic whole-body grasping with contact. In *European Conference on Computer Vision (ECCV)*, 2022. 1

[56] Yinzhen Xu, Weikang Wan, Jialiang Zhang, Haoran Liu, Zikang Shan, Hao Shen, Ruicheng Wang, Haoran Geng, Yijia Weng, Jiayi Chen, et al. Unidexgrasp: Universal robotic dexterous grasping via learning diverse proposal generation and goal-conditioned policy. In *Computer Vision and Pattern Recognition (CVPR)*, 2023. 8

[57] Lixin Yang, Kailin Li, Xinyu Zhan, Fei Wu, Anran Xu, Liu Liu, and Cewu Lu. OakInk: A large-scale knowledge repository for understanding hand-object interaction. In *Computer Vision and Pattern Recognition (CVPR)*, 2022. 1, 2, 3, 6

[58] Lixin Yang, Jian Xu, Licheng Zhong, Xinyu Zhan, Zhicheng Wang, Kejian Wu, and Cewu Lu. Poem: Reconstructing hand in a point embedded multi-view stereo. In *Computer Vision and Pattern Recognition (CVPR)*, 2023. 7

[59] Yaqi Zhang, Di Huang, Bin Liu, Shixiang Tang, Yan Lu, Lu Chen, Lei Bai, Qi Chu, Nenghai Yu, and Wanli Ouyang. MotionGPT: Finetuned llms are general-purpose motion generators. *arXiv preprint arXiv:2306.10900*, 2023. 8

[60] Hao Zheng, Regina Lee, and Yuqian Lu. Ha-vid: A human assembly video dataset for comprehensive assembly knowledge understanding. *arXiv preprint arXiv:2307.05721*, 2023. 3, 6

[61] Zehao Zhu, Jiashun Wang, Yuzhe Qin, Deqing Sun, Varun Jampani, and Xiaolong Wang. ContactArt: Learning 3d interaction priors for category-level articulated object and hand poses estimation. *arXiv preprint arXiv:2305.01618*, 2023. 1, 2, 6