Supplementary Material for ParaHome: Parameterizing Everyday Home Activities Towards 3D Generative Modeling of Human-Object Interactions

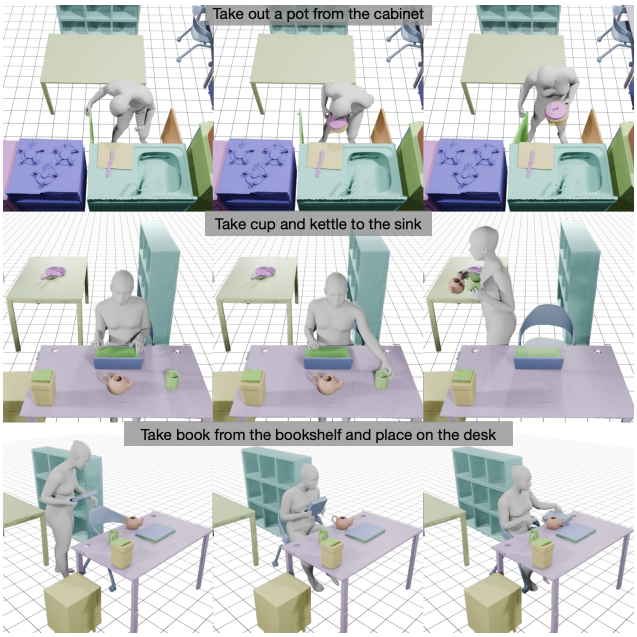


Figure 1. Rendered scenes and text annotation for each scene from example scenario.

1. Dataset Details

1.1. Dataset Contents

Scanned Object Mesh. We obtain high-quality 3D mesh scans of all objects placed in our system via an Einstar3D scanner [5]. We scan each object at least twice to reduce the unscanned areas or holes by changing the orientations of the objects (e.g., up-side-down), and fuse the scanned meshes via manual alignments. They are zero-centered and scaled to a metric scale.

Object Articulation Information. Objects with articulation contain axis a_e. If the part has a revolute joint, we include pivot point p_e additionally. These are defined in the object canonical space and are utilized in getting each object part-transformation toward the camera space.

Object Position and Orientation in the Camera Space. Each object's spatial information is computed using the per-frame transformation of markers attached to each object.

Relative Orientation of Hand/Body Joints. Orientation of each hand and body joints with respect to their parent joints is recorded and processed via a motion capture system.

3D Hand/Body Joint Positions in the Camera Space. With the positions of markers attached to the body in the mocap space acquired via body alignment protocol, translation and orientation of body to camera space are obtained using the positions of corresponding markers in the camera spaces. We compute the positions of two hands and body using the obtained translation and orientation.

Text Annotation for Each Action. For each capture, participants receive verbal instructions detailing the actions they will perform. These instructions specify which objects to interact with and how to interact with them, as illustrated in Fig. 1. The instructions are recorded and synchronized with the motion data. Additionally, we manually inspect the instrument to create more accurate text annotations, ensuring they are reliably mapped to each action.

Per-frame Contact Information. At each frame where contact between Left/Right/Body and object occurs, the corresponding frame and object category/body part information is recorded.

1.2. Dataset Comparison

As shown, our ParaHome dataset is the comprehensive dataset which captures all authentic and dynamic human-object interaction scenarios in a natural room environment. Compared to virtual setup using object placeholders [13] or the simulator environment, our data collection pipeline features more precise and reliable capture during manipulation, resulting in lower temporal jitter and reduced rate of object movement without hand contact. The impact of our dataset's higher accuracy on the downstream task is evaluated and discussed in Sec. 5.2. Our dataset includes dexterous body motion and movement of all objects in the scene and encompasses various manipulation motions involving articulated objects and multiple objects even in concurrent usage scenarios. Our capture scenarios feature natural and sequential manipulations like cooking, resulting in longer semantically concurrent action sequences (e.g. placing a pot on a stove and turning it on), as shown in our supplementary video. And such logically connected interactions are hard to find in a less structured setting and with randomly performed

Dataset	hours #	subject #	object #	body	hand #	contact	obj. 6d.	obj. artic.	multi obj.	setup
GRAB [25]	3.8	10	51	✓	2	✓	✓	✗	✗	standing
BEHAVE [2]	4.2	8	20	✓	-	✓	✓	✗	✗	portable
InterCap [11]	0.6	10	10	✓	2	✓	✓	✗	✗	portable
FHPA [7]	0.9	6	26	✗	1	✗	✓	✗	✗	room
H2O [14]	1.1	4	8	✗	2	✗	✓	✗	✗	table
H2O-3D [9]	-	5	10	✗	2	✗	✓	✗	✗	table
HOI4D [17]	22.2	9	800(16)	✗	1	✗	✓	✓	✗	room
Chairs [12]	17.3	46	81	✓	2	-	✓	✓	✗	standing
ARCTIC [6]	1.2	10	11	✓	2	✓	✓	✓	✗	standing
NeuralDome [27]	4.6	10	23	✓	2	✓	✓	✓	✗	standing
OAKINK2 [26]	12.38	9	75	✓	2	✓	✓	✓	✓	table
TACO [18]	2.53	14	196(20)	✗	2	✓	✓	✗	✓	table
TRUMANS [13]	15	7	20(placeholders)	✓	2	✓	✓	✓	✓	room
Ours	8.1	38	22	✓	2	✓	✓	✓	✓	room

Table 1. Comparison of existing human-object interaction datasets

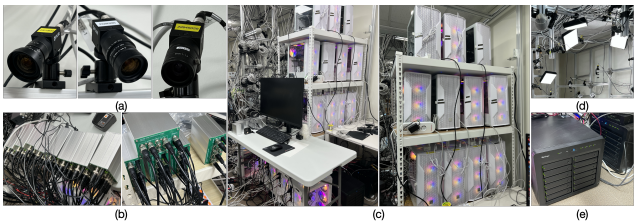


Figure 2. System Devices. (a) RGB cameras with 3 types of lenses. (b) Signal distributers (c) Desktop machines (d) LED Lights (e) NAS storage systems

actions. Furthermore, we collected data from 38 participants, capturing a wide range of motion styles across individuals.

2. System Details

2.1. Using ArUco Markers

Even though several works [6, 12] utilized IR markers for motion tracking, we find using ArUco markers to be more suitable in our capture system. We aim to capture in a broader spatial spectrum(i.e. entire room setting filled with objects) involving multiple interactions in a single capture time. Such environments filled with multiple furniture/objects and hand-object interactions involving multiple direct contacts, cause a significant occlusion as simulated in Sec.4.1 of our main manuscript. Even though ArUco markers have its downsides in corrupting RGB data and influencing natural human motions, using RGB data is not within our interest as mentioned Sec 6. of main manuscript and we empirically placed markers to minimize such interruptions.

2.2. Hardware Details

In order to cover the entire volume of the room and to reduce occlusion issues, we install 70 RGB industrial cameras, BFLY-31S4C-C. The cameras capture videos at 30Hz in 2048×1536 resolution. We set the exposure time at $3msec$, which shows a good balance between low-motion blur and sufficient brightness. We use three types of lenses (thirty $3mm$ lenses, twenty $5mm$ lenses, and twenty $6mm$ lenses), where the wide-angle lens ($3mm$) is helpful in capturing wide area. We calibrate the cameras using Structure-from-Motion via COLMAP [24] with multiple randomly patterned fabrics placed in our system. We provide pre-calibrated initial intrinsic parameters for the three types of lenses derived from 2 or 3 samples of lenses for better convergence in camera pose estimation. We scale the calibrated 3D space into a real-world metric (in meters) by locating checkerboards with known sizes during camera calibration.

All cameras, the motion capture suit, and gloves are synced and gen-locked via a common square wave signal that comes from the motion capture device to synchronize two heterogeneous systems, which is crucial to precise HOI captures. To deliver the sync signals to a large number of cameras, we utilized 11 signal distributors in a hierarchical manner, each of which can be connected to 8 cameras via GPIO cables. We use 1 master and 18 slave desktop machines to control the cameras and process captured records. Each slave machine is connected to 3 or 4 cameras via Ethernet cables and equipped with a 4-port 1G ethernet board, and 2 SSDs with a capacity of 500GB and 1TB each. 15 LED lights ($4500lm$) are installed to provide sufficient illumination. Pictures of our system devices are shown in Fig. 2.

To capture both body motion and subtle hand motions, we use IMU-based motion capture equipments, Xsens motion suit [21] and Manus hand gloves [20]. The body motion system captures the motions at 60Hz.

Object	Part1	Part2
Sink	revolute	revolute
Laptop	revolute	-
Drawer	sliding	sliding
Gas stove	revolute	revolute
Microwave	revolute	-
Trashbin	revolute	-
Washing machine	revolute	-
Refrigerator	revolute	revolute

Table 2. Part information of articulated objects

3. Data Acquisition

3.1. Modeling Object Articulations.

To capture the movement of articulated objects, we model each object as a parametric 3D model by defining the object-specific articulated motion parameters. This modeling requires scanning individual parts separately and compositing them in a canonical space by defining axis direction, pivot points, revolute joints, and so on, based on the object types. During HOI captures, we track the motion of each part via our marker system (e.g. monitor of a laptop and the base), from which we compute the articulated motion parameters. In this subsection, we describe the process of modeling articulated objects as parametric 3D models. Articulation information of each object with multiple parts is shown in Tab. 2.

To find axis $\mathbf{a}_e$ and pivot point $\mathbf{p}_e$ of the articulated objects, we capture markers attached to each object part at different part states separately and acquire each marker corners in the ParaHome space as $\{m_i(t)\}_{t=1}^n$. Prior to applying algorithm, we transform marker corners $\{m_i(t)\}_{t=1}^n$ back to object canonical space with $T_{mar\rightarrow obj}^{-1}$ and utilize transformed marker corners in the canonical space $\{m_i'(t)\}_{t=1}^n$. For the sliding joint, axis a_e can easily be calculated using marker corners at time t and t' as:

$$\mathbf{a}_e = \frac{m_i'(t) - m_i'(t')}{\|m_i'(t) - m_i'(t')\|}$$

In case object part has a revolute joint, we start initializing an axis $\mathbf{a}_e$ and each relative state $\Delta s_e(t,t') = |s_e(t) - s_e(t')|$ between time t and t' (for the target articulated object captured at different n number of states, time t and time t' satisfies $t \neq t'$ and $t, t' \in \{1, 2, \cdots, n\}$). Then we apply optimization with marker corners toward all possible pairs of times t and t'. Let f be a map defining rotation transformation with respect to pivot and given axis-angle and denote as $\mathbf{T}_{t'\rightarrow t} = f(a_e, \Delta s_e(t,t'), p_e)$. Then for a set of all possible time pairs $\mathbf{P}$, the optimization target for axis a_e, relative state Δs_e and pivot p_e is defined as:

$$\underset{a_e, \Delta s_e(t,t'), p_e}{\arg\min} \sum_{(t,t') \in \mathbf{P}} \|m_i'(t) - \mathbf{T}_{t'\rightarrow t} m_i'(t')\|^2$$

Since initial axis $\mathbf{a}_e$ and pivot $\mathbf{p}_e$ are initialized in the object canonical space, we directly utilize acquired information to derive transformations using detected markers for each capture data.

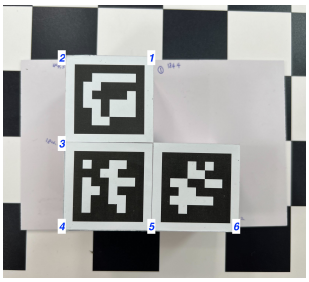
3.2. Body Alignment Detail (Sec 3.4 in Main Paper)

In this subsection, we provide additional details of our spatial alignment process between a multiview camera system and wearable motion capture systems, described in Sec. 3.2 in our main manuscript.

To resolve the issue of imperfect body and hand skeleton scale from the wearable motion capture system, we attach 3 or 4 ArUco markers to each near-rigid body part (torso, hands, upper arms, lower arms, upper legs, lower legs) to assign correspondences. During alignment capture, participants perform the range-of-motion movement by rotating their arms and legs while pinned or bent, particularly twisting their wrists to locate each hand wrist. With the captured data, we optimize body skeleton configuration $\mathcal{B} = \{\mathcal{O}\}$ and body markers locations $\mathcal{M}^b$ via gradient decent with a learning rate of 0.008 for 50 epochs. Specifically for weights of body and foot, $\lambda_b = 100, \lambda_f = 5000$ are used. In the case that alignment is not well optimized, we additionally penalize excessive length change in spines and difference in skeleton lengths between the left and right sides of the body by adding an extra regularization term. Once the alignment procedure is finished, we remove markers (all from the upper legs, one for each upper arm, lower arm, and lower leg) to minimize interference with the movements of the participant. The selection of remaining markers is determined based on their importance during captures, where we assess their importance by evaluating whether their absence would compromise the accuracy of body positioning in the camera space. Check our supplementary video for an example of body alignment motion.

3.3. Hand Calibration Structure and Protocol (Sec 3.4 in Main Paper)

As human usually handle objects with their fingers, fingertips play an important role during interaction. We made the calibration structure to better locate fingertips and find the hand skeletons and relative locations between the attached hand markers to each wrist. The hand calibration structure is composed of three cubes with ArUco markers and the ordered 3D corner vertices of the structure are defined as $C = \{\mathbf{c}_i \in \mathbb{R}^3\}_{i=1}^6$ as shown in Fig. 3. During the hand calibration procedure, we request each participant to touch the calibration structure's corners with their fingertips. We instruct them to touch specified multiple corners at each step



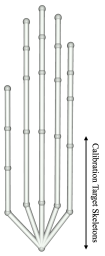


Figure 3. (Left) The hand calibration structure to precisely measure hand skeleton configuration and to find the relative locations of hand markers attached to the wrist in hand-centric coordinate (Right) Hand skeleton and Calibration targets

Corner #	Hand Side	Seq1	Seq2	Seq3	Seq4
1,2	Right	1, 2	1, 3	1, 4	1, 5
1,3	Right	1, 2	1, 3	1, 4	1, 5
2,4	Right	1, 2	1, 3	1, 4	1, 5
5,2	Right	1, 2	1, 3	1, 4	1, 5
6,2	Right	1, 2	1, 3	1, 4	1, 5
6,3,2	Right	1, 2, 3	1, 3, 4	1, 4, 5	-
2,1	Left	1, 2	1, 3	1, 4	1, 5
3,1	Left	1, 2	1, 3	1, 4	1, 5
4,2	Left	1, 2	1, 3	1, 4	1, 5
5,2	Left	1, 2	1, 3	1, 4	1, 5
6,2	Left	1, 2	1, 3	1, 4	1, 5
2,5,6	Left	1, 2, 3	1, 3, 4	1, 4, 5	-

Table 3. Hand Calibration Protocol

using two or three fingertips. A participant undergoes 23 steps of such touching processes per-hand. The Tab. 3 comprises hand calibration instructions for subjects to follow. Corner # is a set of two or three target corner numbers of the calibration structure which the subject should contact with their fingertips. Also, the orders of fingers to touch the target corners are specified with numbers corresponding to each finger, which are (1:Thumb, 2:Index, 3:Middle, 4:Ring, 5:Little). An example of the hand calibration procedure is shown in our supplementary video.

3.4. Implementation Details on Hand Calibration

Here we describe the details of the hand calibration method. As described in section 3.4, optimization parameters are hand skeleton configuration $\mathcal{H} = \{\mathcal{S}^h, \mathcal{O}^h\}$ and positions of 3D markers in the local hand-centric coordinate $\mathcal{M}^h$. We empirically decide the general target range of the optimization skeleton to the palm area shown in Fig. 3 and add constraints that limit the skeleton scales(s_i) for each skeleton segment i between $0.8 \leq s_i \leq 1.2$, and additional skeleton offset value(δ_j) for target joints j with $|\delta| \leq 0.01$ in meter scale to avoid unnatural deformation of hand skeleton. The location of hand markers $\mathcal{M}^h$ is optimized through a total of 150 iterations. The skeleton scale and additional offset are optimized starting from 50 and 100 iterations each. We use three losses, L_{tip} to measure the Euclidean distance

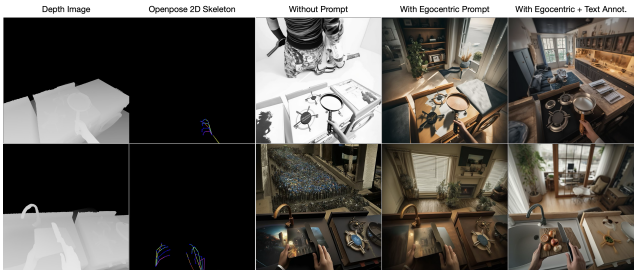


Figure 4. Synthesized RGBs and Comparison between with or without text annotation.

from the hand tips to paired corners, L_{wrist} to measure the distance from the wrist location from body motion capture device and the wrist position computed by hand marker position and L_{pen} to measure penetration of hand to the calibration structure. The penetration loss is computed by a cosine similarity between the calibration structure's normal vector and the target corner-to-hand tip vector. In summed loss $\lambda_t L_{tip} + \lambda_w \mathcal{L}_{wrist} + \lambda_p \mathcal{L}_{pen}$, losses are weighted equally by $\lambda_t = 1, \lambda_w = 1, \lambda_p = 1$. But they are manually adjusted based on the touch accuracy and body calibration accuracy per participant. After the alignment process, the average Euclidean distance between the corner and the target fingertip results in 0.83 (in centimeters).

3.5. Fitting Human Body Model

We illustrate details on fitting SMPL-X [19] pose parameters to Xsens skeletons using the optimized shape parameters. For SMPL-X hand pose, we devise an optimization scheme which runs FABRIK solver [1] initially to get SMPL-X hand skeleton retargetted to Xsens hand and optimize each hand joint pose directly to fit into each retargetted joint positions. For body pose, we do not consider global orientation and translation in acquiring body pose for simplicity. As an input data representation, we split each sequence with a window size of 60, and reform body joint rotations except for hands in 6D representations [29], thus $x_{in} \in \mathbb{R}^{60\times21\times6}$ per batch. For the model, we use a variation of Temporal Convolution Network [15] for the encoder and decoder. During training, we define the default reconstruction loss, $\mathcal{L}_{\text{recon}}$, for joint rotation and further incorporate the end effector loss, $\mathcal{L}_{\text{end}}$. This additional loss includes the SMPL-X vertices of the hands and legs, as well as the wrist, foot, and hand tip joints. To regulate any present noise in the motion, we also add joint velocity loss as an regularization. Then the loss sums up to:

$$\mathcal{L} = \lambda_{recon}\mathcal{L}_{recon} + \lambda_{end}\mathcal{L}_{end} + \lambda_{vel}\mathcal{L}_{vel}$$

After training, we extract windows of all sequences with step size of 30, and initialize a latent code z_{pose} by feeding the encoder with the Xsens joint rotation data by roughly matching joint category between two different skeletons(i.e.

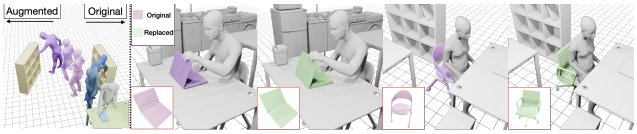


Figure 5. Augmented data visualization of layout change (Left) and asset replacement (Right)

Dataset	Method	AUC@IoU$_{25}$↑	AUC@IoU$_{50}$↑	AUC@IoU$_{75}$↑
ROPE [28]	IST-Net [16]	28.7	10.6	0.5
ROPE [28]	GenPose++ [28]	**39.9**	**19.1**	**2.0**
ParaHome$_{all}$	GenPose++ [28]	26.4	10.3	0.6
ParaHome$_{rigid}$	GenPose++ [28]	29.7	12.2	0.9

Table 4. Quantitative comparison of category-level object pose estimation on ROPE [28] and ParaHome synthetic data. Since the two datasets differ in the presence of articulation, we divide ParaHome data into two subsets, all including articulation objects and $rigid$ with only rigid objects.

jLeftT4Shoulder of Xsens to left collar of SMPL-X). Then we optimize z_{pose} by feeding into the trained decoder to fit with Xsens skeleton wrist, hand tip, ankle, and foot joints. Thus denoting a set of paired Xsens and SMPL-X target joints as $\mathcal{J}$, we formulate the optimization problem as:

$$z^*_{\text{pose}} = \arg\min_{z_{\text{pose}}} \sum_{j_{\text{xsens}}, j_{\text{smplx}} \in \mathcal{J}} \| j_{\text{xsens}} - j_{\text{smplx}} \|^2$$

After optimization, we use decoded output body pose using z^*_{pose}. Since we sample sequences as 60-length windows with step size of 30, there exists discrepancies in body poses where contiguous windows overlap. We use *slerp* to compensate for such discontinuities for each joint pose parameters.

3.6. Extending Dataset

To provide variations in our dataset, augmenting dataset is possible in two ways: (1) Changing object layout and (2) Asset replacement with ShapeNet/SAPIEN. For layout changes, human motion has to be modified accordingly considering target displacement. We utilize MDM model trained with AMASS and ParaHome for motion in-betweening from source pose to target pose in different location. For asset replacement, we perform ICP to replace target object. The augmented scenes are shown in Fig. 5.

3.7. Synthesizing Realistic RGB

Utilizing the ParaHome dataset, which provides diverse and rich 3D motion data, we generate RGB images from various viewpoints, all aligned with 3D annotations. We employ a diffusion-based image synthesis model [3] combined with ControlNet to create 2D RGB images consistent with the 3D data. Human-object interaction scenes are rendered from multiple perspectives, including egocentric, high-angle, and front-facing views. Rendered depth maps and OpenPose [4] joint information are integrated, along with text prompts to enhance image quality and alignment with the original data. The impact of including text information is shown in Fig. 4, implying improved alignment and realism. Quantitative results on off-the-shelf 3D estimation models are presented in Table 4 and Table 5. For object 6D estimation, our synthetic data achieves accuracy comparable to the ROPE dataset, demonstrating the realism of the generated images for model to detect. Specifically, occlusions from hand interactions and complex object articulation in the ParaHome dataset result in lower accuracy, as shown in Fig.6, suggesting future potential improvement. For 3D hand pose estimation, the synthetic data performs competitively with other datasets quantitatively, though occlusions during manipulation lead to slightly reduced accuracy compared to HO3D[8] and FreiHAND [30], as illustrated in Fig. 7.

Dataset	Method	PA-MPJPE↓	PA-MPVPE↓	F@5↑	F@15↑
FreiHAND [30]	HaMeR [23]	6.0	5.7	0.785	0.990
HO3D [8]	Pose2Mesh [23]	12.5	12.7	0.441	0.909
HO3D [8]	HaMeR [23]	7.7	7.9	0.635	0.980
ParaHome(Ours)	HaMeR [23]	9.47	9.46	0.25	0.85

Table 5. Quantitative comparison of 3D hand pose reconstruction on FreiHand, HO3D and ParaHome synthetic data. PA-MPVPE and PA-MPJPE are meausred in mm.

Figure 6. Rendered results of 6d reconstruction model on synthesized data. (Left) Successful cases. (2) Failure cases due to the occlusion.

Figure 7. Rendered results of the 3D hand reconstruction model on synthesized data. (Left) Successful cases. (Right) Cases with large 3D keypoints loss due to occlusion.

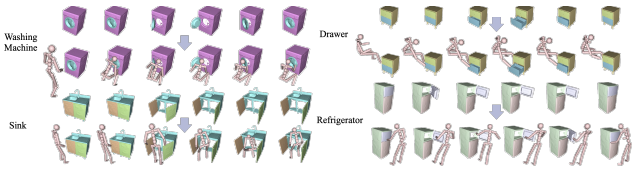


Figure 8. Synthesized body motions conditioned by sequences of object state

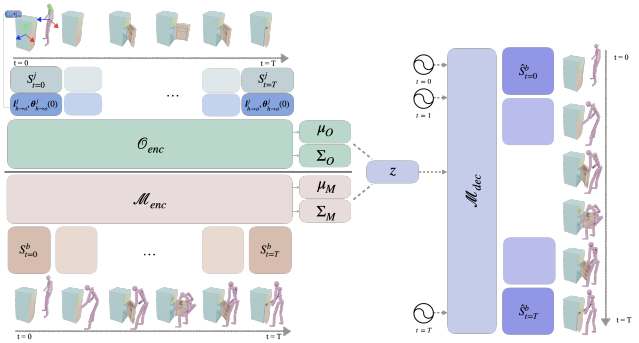


Figure 9. Model Architecture estimating human motion based on object states

4. Sequence Example Visualization

4.1. Sequence Visualization

Sampled data from our collected datasets are shown in Fig. 11. Corresponding text annotations for actions are provided under the caption.

5. Experiments

5.1. Synthesizing Body Motion for Desired Object Manipulation

Train Details: The goal of our model shown in Fig. 9 is to synthesize a plausible 3D human motion conditioned with sequences of object state at a range of times. We represent target object status at each time t as $\mathbf{S}_{to}(t) = \{\phi^j(t)\}$ using joints state $\phi^j(t) \in \mathbb{R}^2$. We represent body pose at time t as $\mathbf{S}_p(t) = \{X_t, \Delta p, \Delta r\}$ using body pose X_t, root's linear velocity Δp and angular velocity Δr. We test with two types of body pose representation: the person root-centered skeleton representation [10], and the SMPL-X [22] body pose. In training, we use AdamW optimizer and LR=$1e-4$ with 1500 epochs and batch size 32.

Additional Results: We train our baseline model for four objects including a refrigerator, drawer, washing machine, and sink with window sizes 30, 60, and 90. The quantification results are shown in Table 6. As shown in the result table, as the window size decreased, the accuracy increased in most items and root-centered body skeleton representation results in better accuracy for pose-dependent attributes (rc-joints, rc-wrists, glb-joints) but SMPL-X notation results in better global orientation and root position. Additional examples of visualization are shown in Fig. 8 and our supplementary video.

Object	body repr.	window	MPE↓(cm) rc-joints	rc-wrists	glb-root	glb-joints	MOE↓	Foot skating↓
Refrigerator	root centric	30	0.58	1.36	0.68	1.15	0.22	0.98
		60	0.76	1.73	2.47	3.22	0.43	1.05
		90	0.97	2.09	5.60	6.57	0.75	1.74
	SMPL-X	30	1.67	2.69	0.68	2.32	0.12	1.58
		60	1.93	2.94	1.83	3.95	0.24	1.50
		90	3.49	5.26	5.32	8.87	0.53	2.67
Drawer	root centric	30	1.51	2.10	0.81	1.86	0.13	1.04
		60	1.52	1.74	1.02	2.26	0.24	0.80
		90	2.61	2.90	3.92	5.88	0.66	1.58
	SMPL-X	30	3.43	4.04	0.40	3.87	0.11	1.38
		60	3.27	3.83	0.96	4.17	0.12	1.04
		90	2.78	3.39	2.73	5.20	0.18	1.10
Washing Machine	root centric	30	0.50	0.89	0.35	0.77	0.15	0.94
		60	0.55	1.06	1.44	2.14	0.48	1.35
		90	1.29	2.62	9.03	11.46	0.92	2.60
	SMPL-X	30	1.94	2.53	0.42	2.24	0.09	1.04
		60	3.68	4.29	2.53	6.59	0.44	2.72
		90	6.03	7.55	6.01	13.29	0.60	2.68
Sink	root centric	30	0.58	0.98	0.42	0.82	0.16	0.79
		60	0.61	0.94	1.02	1.47	0.29	0.81
		90	1.21	1.93	3.83	4.84	0.76	1.67
	SMPL-X	30	2.27	2.91	0.52	2.64	0.10	0.90
		60	2.64	3.12	1.26	3.79	0.18	0.87
		90	2.60	3.12	2.16	4.80	0.27	1.01

Table 6. Quantitative results of the ParaHome task.

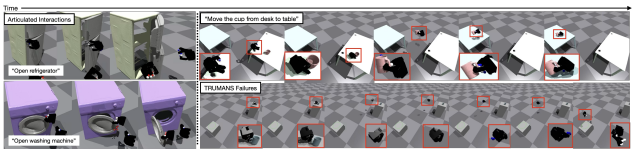


Figure 10. Hand retargetting results with articulated objects and performing task

5.2. Manipulation in Physics Engine

We qualitatively compare TRUMANS [13] and our dataset in IsaacGym by loading the 3D assets and retargeting hand motions into Allegro hand (via Dex-Retargeting). Our dataset demonstrates significantly better feasibility, as seen in Fig. 10 for a "moving cup" example, whereas the similar action of TRUMANS fails. We further successfully tested more complicated actions involving articulated objects of our dataset (e.g opening doors of refrigerator and washing-machine). Formulating physical interaction using simulator should be one of the key applications, and tracking accuracy matters when it comes to retargetting joint motions to other actors in the environment, which our dataset has superiority over other datasets.

References

[1] Andreas Aristidou, Yiorgos Chrysanthou, and Joan Lasenby. Extending FABRIK with model constraints. *Comput. Animat. Virtual Worlds*, 2016. 4

[2] Bharat Lal Bhatnagar, Xianghui Xie, Ilya A Petrov, Cristian Sminchisescu, Christian Theobalt, and Gerard Pons-Moll. Behave: Dataset and method for tracking human object interactions. *CVPR*, 2022. 2

[3] Black Forest Lab, 2024. https://huggingface.co/black-forest-labs. 5

[4] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. 2017. 5

[5] Einstar, 2023. https://www.einstar.com/. 1

[6] 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. 2023. 2

[7] 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. *CVPR*, 2018. 2

[8] Shreyas Hampali, Mahdi Rad, Markus Oberweger, and Vincent Lepetit. Honnotate: A method for 3d annotation of hand and object poses. *CVPR*, 2020. 5

[9] 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. *CVPR*, 2022. 2

[10] Daniel Holden, Jun Saito, and Taku Komura. A deep learning framework for character motion synthesis and editing. *ACM Transactions on Graphics (TOG)*, 2016. 6

[11] Yinghao Huang, Omid Taheri, Michael J Black, and Dimitrios Tzionas. Intercap: Joint markerless 3d tracking of humans and objects in interaction. *GCPR*, 2022. 2

[12] Nan Jiang, Tengyu Liu, Zhexuan Cao, Jieming Cui, Zhiyuan Zhang, Yixin Chen, He Wang, Yixin Zhu, and Siyuan Huang. Full-body articulated human-object interaction. In *ICCV*, 2023. 2

[13] Nan Jiang, Zhiyuan Zhang, Hongjie Li, Xiaoxuan Ma, Zan Wang, Yixin Chen, Tengyu Liu, Yixin Zhu, and Siyuan Huang. Scaling up dynamic human-scene interaction modeling. 2024. 1, 2, 6

[14] Taein Kwon, Bugra Tekin, Jan Stühmer, Federica Bogo, and Marc Pollefeys. H2o: Two hands manipulating objects for first person interaction recognition. *ICCV*, 2021. 2

[15] Colin Lea, Michael D Flynn, Rene Vidal, Austin Reiter, and Gregory D Hager. Temporal convolutional networks for action segmentation and detection. 2017. 4

[16] Jianhui Liu, Yukang Chen, Xiaoqing Ye, and Xiaojuan Qi. Prior-free category-level pose estimation with implicit space transformation. 2023. 5

[17] 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. *CVPR*, 2022. 2

[18] Yun Liu, Haolin Yang, Xu Si, Ling Liu, Zipeng Li, Yuxiang Zhang, Yebin Liu, and Li Yi. Taco: Benchmarking generalizable bimanual tool-action-object understanding. pages 21740–21751, 2024. 2

[19] Naureen Mahmood, Nima Ghorbani, Nikolaus F Troje, Gerard Pons-Moll, and Michael J Black. Amass: Archive of motion capture as surface shapes. *ICCV*, 2019. 4

[20] Manus, 2023. https://www.manus-meta.com/. 2

[21] Movella, 2023. https://base.xsens.com/. 2

[22] 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 *CVPR*, 2019. 6

[23] Georgios Pavlakos, Dandan Shan, Ilija Radosavovic, Angjoo Kanazawa, David Fouhey, and Jitendra Malik. Reconstructing hands in 3d with transformers. 2024. 5

[24] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *CVPR*, 2016. 2

[25] Omid Taheri, Nima Ghorbani, Michael J Black, and Dimitrios Tzionas. Grab: A dataset of whole-body human grasping of objects. *ECCV*, 2020. 2

[26] Xinyu Zhan, Lixin Yang, Yifei Zhao, Kangrui Mao, Hanlin Xu, Zenan Lin, Kailin Li, and Cewu Lu. Oakink2: A dataset of bimanual hands-object manipulation in complex task completion. 2024. 2

[27] Juze Zhang, Haimin Luo, Hongdi Yang, Xinru Xu, Qianyang Wu, Ye Shi, Jingyi Yu, Lan Xu, and Jingya Wang. Neuraldome: A neural modeling pipeline on multi-view human-object interactions. In *CVPR*, 2023. 2

[28] Mengchen Zhang, Tong Wu, Tai Wang, Tengfei Wang, Ziwei Liu, and Dahua Lin. Omni6d: Large-vocabulary 3d object dataset for category-level 6d object pose estimation. 2025. 5

[29] Yi Zhou, Connelly Barnes, Lu Jingwan, Yang Jimei, and Li Hao. On the continuity of rotation representations in neural networks. In *CVPR*, 2019. 4

[30] Christian Zimmermann, Duygu Ceylan, Jimei Yang, Bryan Russell, Max Argus, and Thomas Brox. Freihand: A dataset for markerless capture of hand pose and shape from single rgb images. *ICCV*, 2019. 5

Figure 11. Example scenes of ParaHome dataset and aligned text annotation (Column1) Move kettle and cup from desk to the sink. (Column2) Take laptop from the desk and move to the table. (Column3) Take pan from the gas stove to the table. (Column4) Put laundry in the washing machine. (Column 5) Throw away STH into trash can.