TACO: Benchmarking Generalizable Bimanual Tool-ACtion-Object Understanding —Supplementary Material

Contents

A. Interaction Field Estimation

With accurate object models and MANO [13] meshes captured in our dataset, we benchmark estimating interaction fields of hands and objects from color images. Apart from recovering hand-object interaction fields in existing work [4], our task also involves estimating those between tools and target objects.

Problem formulation: Representing the left hand, right hand, tool, and target object with l, r, t, and o, the task is to estimate six interaction fields between hands and objects ($F^{r\to t}$, $F^{t\to r}$, $F^{l\to o}$, $F^{o\to l}$, $F^{t\to o}$, $F^{o\to t}$) from a given RGB frame, where field $F^{a\to b}$ is defined as the distance to the nearest vertex in mesh b for all vertices in mesh a.

Evaluation metrics: Following [4], we use the Mean Distance Error to evaluate the precision of predicted interaction fields, and the Acceleration Error to measure the smoothness of those estimates.

Baselines, results and analysis: We set up two baseline methods based on InterField-SF [4]. The first one (InterField-SF separated) takes an image and two meshes as input (e.g. the right hand and the tool) and estimates the two fields between them. The second one (InterField-SF concatenated) incorporates the image with meshes of both hands, the tool, and the target object, and predicts six interaction fields altogether. Table 1 shows the quantitative results of the two methods. The primary distinctions among test sets lie in the selection of tools and actions, with a relatively weaker correlation to target objects. The results suggest that these variations exert a more pronounced influence on RT and TR. Within the fields of RT and TR, the method performances on S1 significantly surpass those on S3, while the latter outperforms both S2 and S4. This indicates that the methods face challenges in generalizing to unseen geometries. The incorporation of seen geometries with unseen actions (S3) also introduces additional complexities. These challenges in generalization to both unseen geometries and actions underscore the need for further exploration and refinement of the proposed methods.

B. Data Capturing Details

B.1. Camera Calibration

Camera intrinsic calibration. We use a traditional method that places a checkerboard in the camera view with known scales of grids and estimates the camera intrinsic matrix and distortion using OpenCV functions.

Camera extrinsic calibration. After acquiring the camera intrinsic, we perform a semi-automatic process for calibrating the camera extrinsic before data capturing. As shown in Figure 1, we first place 12 markers in the scene. Benefiting from our mocap system, we can obtain accurate marker positions in the world coordinate system with errors less than 1mm. We then manually annotate the pixel coordinate of each marker in the color image, and compute the optimal camera extrinsic minimizing re-projection error of markers. We solve this Perspective-n-Point (PnP) problem

Test Set	Method	Mean Distance Error (mm, ↓)						Acceleration Error (m/s^2, ↓)					
		RT	TR	LO	OL	TO	OT	RT	TR	LO	OL	TO	OT
S1	InterField-SF (separated)	8.4	8.7	10.4	21.5	**11.6**	**15.7**	**8.7**	**8.3**	**10.3**	12.3	**13.7**	**16.3**
	InterField-SF (concatenated)	**8.1**	**8.5**	**10.1**	**21.3**	12.7	17.3	9.0	8.6	10.4	**11.9**	14.6	17.7
S2	InterField-SF (separated)	**14.9**	32.5	14.7	**18.9**	15.6	**12.5**	**10.5**	11.4	**12.0**	**10.7**	**13.8**	**11.6**
	InterField-SF (concatenated)	15.1	**31.5**	**14.6**	19.1	**15.2**	12.8	10.7	**11.3**	12.2	10.9	14.4	12.6
S3	InterField-SF (separated)	13.6	17.2	**15.1**	**26.6**	**13.0**	**14.0**	**9.8**	9.1	11.2	**10.8**	**13.0**	**13.4**
	InterField-SF (concatenated)	**13.4**	**16.9**	15.2	27.1	14.3	14.2	9.9	**9.0**	**11.1**	11.1	13.9	14.3
S4	InterField-SF (separated)	**13.9**	**34.2**	**12.5**	**19.0**	**15.0**	**12.5**	12.6	**9.6**	**10.7**	**10.3**	**14.5**	**13.3**
	InterField-SF (concatenated)	14.0	35.4	12.6	19.3	15.2	13.0	**10.6**	10.0	11.2	**10.3**	15.4	14.5

Table 1. Results on interaction field estimation [4], where R denotes right hand, T denotes tool, L denotes left hand and O denotes target object (e.g. RT means right-hand-to-tool). Methods are examined via Mean Distance Error and Acceleration Error.

with OpenCV algorithms.

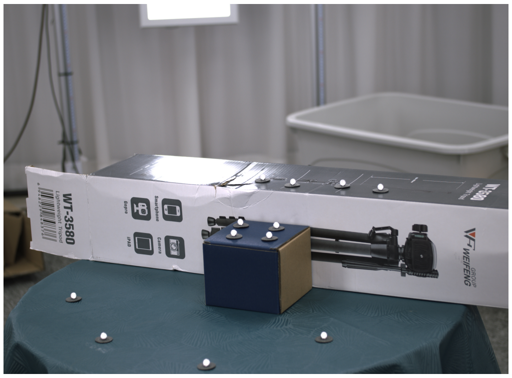

Figure 1. Calibrating the camera extrinsic.

B.2. Time Synchronization

We provide time-synchronized data from the different sensor modalities. Our 12 industrial FLIR cameras receive signals from the same signal generator through audio cables. To synchronize industrial cameras with our mocap system and Realsense L515 camera, we record UTC timestamps for each frame captured by different cameras and perform nearest-neighbor matching among timestamps. The maximal time difference between matched signals is 17ms.

C. Data Annotating Details

C.1. Details on Object Pose Optimization

We attach four markers with a radius of 4mm to the surface of each object and obtain the object pose by capturing marker positions by the optical mocap system. To reuse the markers and optimization results, we mark a target position on the object surface for each marker and attach markers to these fixed positions before data collection. For each object, we formulate the attached four markers as a rigid body B and optimize the relative 6D pose $T = [R, t]$ from the 3D object model to B, where $R \in SO(3)$ denotes 3D rotation, and $t \in \mathbb{R}^3$ indicates 3D translation. Since markers actually contact the object surface without interpenetration, we first design contact loss $L_c(q, P)$ and penetration loss $L_p(q, P)$ as:

$$\begin{aligned} L_c(q,P) &= \|q - p_{i^*}\|_2, \\ L_p(q,P) &= \max(-\vec{n}_{i^*}^T(q - p_{i^*}), 0), \end{aligned} \tag{1}$$

where $q \in \mathbb{R}^3$ is a query point, $P = \{p_i \in \mathbb{R}^3\}_{i=1}^{|P|}$ is a point cloud, $i^* = \underset{1 \leq i \leq |P|}{\arg\min} \|q - p_i\|_2$ denotes the index of the closest point in P to q, and $\vec{n}_i$ denotes the normal of the point p_i. We then incorporate the two loss functions and compute the optimal relative pose T^* via the following function:

$$\begin{aligned} T^* = \underset{R,t}{\arg\max} \sum_{k=1}^{K} (L_c(Rq_k + t, P) + L_p(Rq_k + t, P)) \\ [L_c(Rq_k + t, P) < \alpha], \end{aligned} \tag{2}$$

where K is the number of markers, $q_k \in \mathbb{R}^3$ is the marker position in the coordinate system of B, P is the vertices of the object model, and α=1cm is a threshold selecting markers near the object. Given a manual initialization of T, we use the Adam optimizer to find T^* with learning rate 1e-4. In practice, we attach 10 additional markers to the object surface (K=14) to improve the robustness of the optimization, meanwhile using only four of them to track the object during data capturing.

C.2. Details on 3D Hand Keypoint Localization

For the initial frame of the entire sequence, we employ a pre-trained YOLOv3 [12] to obtain the bounding boxes for both left and right hands. For subsequent frames, we leverage the Track-Anything Model [17] along with the optimized hand pose from the preceding frame to generate masks for both hands and compute bounding boxes based on these masks. Then, we crop out sub-images containing only one hand according to these bounding boxes. The resulting sub-images undergo processing via the single-hand pose estimation model MMPose [2] to determine 2D keypoint positions $K_{2D_c}[i]$ for each hand in each camera view. In $K_{2D_c}[i]$, $c \in C$ denotes the set of all allocentric cameras, and $1 \leq i \leq 21$ represents the 21 joints on the hand.

Given that not all positions are accurate, we employ RANSAC [5] to filter out imprecise 2D positions. In every iteration of RANSAC, two 2D keypoint positions $K_{2D_{c_1}}[i]$ and $K_{2D_{c_2}}[i]$ are chosen from two randomly selected different camera views c_1 and c_2. Based on positions $K_{2D_{c_1}}[i]$ and $K_{2D_{c_2}}[i]$, we can calculate their corresponding 3D points $K_{3D}[i]_{<c_1,c_2>}$ in the world coordinate system via triangulation. Subsequently, we project this 3D point onto camera planes and calculate the number of 2D keypoints within 30 pixels around the projected point. After all iterations, the 3D point $K_{3D}[i]_{<c_1,c_2>^*}$ with the highest number of surrounding 2D keypoints is selected as the 3D keypoint $K_{3D}[i]$. This process is defined as

$$\begin{aligned} K_{3D}[i] = &\mathop{\arg\max}_{K_{3D}[i]_{<c_1,c_2>}} \sum_{c=1}^{12} \\ &around_{2D}(proj_c(K_{3D}[i]_{<c_1,c_2>}), K_{2D_c}[i], 30), \end{aligned} \tag{3}$$

where $proj_c(\cdot)$ project $K_{3D}[i]$ onto camera c, and $around_{2D}(\cdot)$ calculates the distance between two points, outputting 1 if the distance is less than 30 pixels and 0 otherwise. In the selected iteration, the 2D keypoints $K_{2D_c}[i]$ that are more than 30 pixels away from the projected point will be deemed invalid and will be excluded from the subsequent optimization stage as $valid_c[i] = around_{2D}(proj_c(K_{3D}[i]), K_{2D_c}[i], 30)$.

C.3. Details on Hand Pose Optimization

We adopt MANO [13] to formulate a 3D hand mesh as $\Theta_h = \{\theta, \beta, t\}$, where $\theta \in \mathbb{R}^{48}$, $\beta \in \mathbb{R}^{10}$, and $t \in \mathbb{R}^3$ represent hand pose, hand shape, and wrist position, respectively. For each participant, the shape parameters β are precomputed based on specially collected data with only two hands and remain fixed in the subsequent hand pose optimization process. The MANO model maps Θ_h to a 3D hand mesh $\{J, V\} = MANO(\Theta_h)$, where $J \in \mathbb{R}^{778\times3}$ and $J \in \mathbb{R}^{21\times3}$ represent vertices and joints on hand, respectively. we first fit a MANO model by minimizing the following loss function:

$$\begin{aligned} \hat{\Theta}_h = &\mathop{\arg\min}_{\Theta_h} (\lambda_{2D}\mathcal{L}_{2D} + \lambda_{3D}\mathcal{L}_{3D} + \lambda_{angle}\mathcal{L}_{angle} + \\ &\lambda_{tc}\mathcal{L}_{tc}), \end{aligned} \tag{4}$$

where $\mathcal{L}_{2D}$ and $\mathcal{L}_{3D}$ encourages the MANO hand joints to align with the 2D and 3D keypoints, $\mathcal{L}_{angle}$ ensures a natural hand pose, and $\mathcal{L}_{tc}$ promotes temporal smoothness. Utilizing object pose, we then refine hand pose by minimizing the following loss function:

$$\begin{aligned} \hat{\Theta}_h = &\mathop{\arg\min}_{\Theta_h} (\lambda_{2D}\mathcal{L}_{2D} + \lambda_{3D}\mathcal{L}_{3D} + \lambda_{angle}\mathcal{L}_{angle} + \\ &\lambda_{tc}\mathcal{L}_{tc} + \lambda_p\mathcal{L}_p + \lambda_a\mathcal{L}_a), \end{aligned} \tag{5}$$

where $\mathcal{L}_p$ prevents hand-object interpenetration, and $\mathcal{L}_a$ encourages hand-object contact.

2D joint loss $\mathcal{L}_{2D}$. The 2D joint loss term is defined as

$$\mathcal{L}_{2D} = \sum_{c=1}^{12}\sum_{i=1}^{21} valid_c[i] \left\| proj_c(J[i]) - K_{2D_c}[i] \right\|^2, \tag{6}$$

where $J[i]$ denotes the i^{th} 3D hand joint position, the $proj_c(\cdot)$ operator projects it onto camera c, $K_{2D_c}[i]$ is the i^{th} 2D keypoint position of hand in the camera view c, and $valid_c[i]$ which is determined in RANSAC indicates whether $K_{2D_c}[i]$ is a valid value.

3D joint loss $\mathcal{L}_{3D}$. The 3D joint loss term is defined as

$$\mathcal{L}_{3D} = \sum_{i=1}^{21} \left\| J[i] - K_{3D}[i] \right\|^2 \tag{7}$$

where J_i denotes the i^{th} 3D hand joint position and $K_{3D}[i]$ is the i^{th} 3D keypoint position fused by 2D keypoint positions from 12 allocentric views in RANSAC. 2D joint loss $\mathcal{L}_{2D}$ and 3D joint loss $\mathcal{L}_{3D}$ provide the most direct supervision for hand pose, aligning the MANO hand with the positions of keypoints.

Angle constraint loss $\mathcal{L}_{angle}$. The angle constraint loss term imposes restrictions on the permissible angles for the rotation of 15 joints, thus preventing undue distortion of the fingers and ensuring a natural hand pose. In the MANO model, the hand pose parameter $\theta \in \mathbb{R}^{48}$, which can be conceptualized as $\theta \in \mathbb{R}^{16\times3}$, signifies 16 axis-angle representations. Among these, 1 axis-angle corresponds to the global rotation of the hand, while the remaining 15 axis-angle represent rotations of 15 joints on the hand. The angle constraint loss term is defined following [19] as

$$\mathcal{L}_{angle} = \sum_{i=1}^{45} \max\left(\underline{\theta_i} - \theta[i], 0\right) + \max\left(\theta[i] - \overline{\theta_i}, 0\right), \tag{8}$$

where $\underline{\theta_i}$ and $\overline{\theta_i}$ denote the upper and lower bounds, respectively, for the i^{th} joint angle parameter $\theta[i]$.

Temporal consistency loss $\mathcal{L}_{tc}$. Due to noise in the data and the randomness in the output of the hand pose estimation model for each frame, the hand pose in the video may exhibit a noticeable degree of jitter. While other loss terms are applied to individual frames, this loss term considers adjacent frames, helping to alleviate the jitter in the hand pose. We draw inspiration from [19] and define temporal consistency loss term as

$$\mathcal{L}_{tc} = \sum_{i\in\mathcal{I}} \left(\left\| \Delta_t^i \right\|^2 + \left\| \Delta_\theta^i - \Delta_\theta^{i-1} \right\|^2 \right), \tag{9}$$

where $\Delta_t^i = t^i - t^{i-1}$ and $\Delta_\theta^i = \theta^i - \theta^{i-1}$. $\mathcal{I}$ represents the index number within the entire sequence, excluding the initial frame.

Attraction loss $\mathcal{L}_a$. During the optimization process, there might be insufficient contact between the hand and the object. The attraction loss term encourages the hands near the object to make sufficient contact with it and is defined as

$$\mathcal{L}_a = \sum_{i=1}^{778} around_{3D}(\mathrm{V_h}[i], \mathrm{V_o}\left[i^*\right], 0.01)\left\|\mathrm{V_h}[i] - \mathrm{V_o}\left[i^*\right]\right\|^2, \tag{10}$$

where $\mathrm{V_h}[i]$ is the i^{th} vertex on hand mesh, $\mathrm{V_o}\left[i^*\right]$ is the vertex on the object closest to $\mathrm{V_h}[i]$, and $around_{3D}(\cdot)$ calculates the distance between two points, outputting 1 if the distance is less than 0.01 meter and 0 otherwise. This loss term is computed twice: once between the right hand and the tool, and once between the left hand and the object.

Penetration loss $\mathcal{L}_p$. During the optimization process, there is a possibility of interpenetration between the hand and the object. This is evidently unrealistic in real-world scenarios. Therefore, a loss term is introduced to mitigate such interpenetration. Similar to [6], we define penetration loss term as

$$\mathcal{L}_p = \sum_{i=1}^{778} \max\left(-\mathbf{n}_{\mathrm{o}}\left(\mathrm{V_o}\left[i^*\right]\right)^T\left(\mathrm{V_h}[i] - \mathrm{V_o}\left[i^*\right]\right), 0\right), \tag{11}$$

where the $\mathbf{n}_{\mathrm{o}}(\cdot)$ operator computes the normal for a vertex, and $\mathrm{V_h}[i]$ represent the i^{th} vertex on the hand mesh, and $\mathrm{V_o}\left[i^*\right]$ denotes the vertex on the object closest to $\mathrm{V_h}[i]$. This loss term is computed twice: once between the right hand and the tool, and once between the left hand and the object.

D. Detailed Statistics on TACO

Object diversities. Figure 2 shows the 20 object categories in our dataset. The categories are chosen from everyday hand-object interaction scenarios, and each object has a proper scale that can be manipulated stably by a single hand. Among these categories, 17 categories are utilized as tools during interaction, and 9 categories are treated as target objects. Figure 3 illustrates 12 object instances from the *brush* category, indicating the diversity of object geometries.

Interaction diversities. As a knowledge base supporting generalizable studies on novel tool-action-object triplets, TACO includes using different tools and target objects to perform the same action types. Figure 4 and 5 show percentages of tool and target object usages in different action types, respectively. All 15 action types involve interaction demonstrations from various kinds of target object categories, while 12 out of them are performed by multiple tool categories.

Motion speed. Table 2 shows statistics on the speed of hand-object motions from different action types. v_r, v_j, v, and ω represent the velocity of the hand wrist, the average velocity of the MANO hand joints, the linear velocity of the object, and the angular velocity of the object, respectively. Compared to target objects, tools always dominate in interaction and have a significantly fast motion speed (16.6 cm/s and 71.3 °/s on average), indicating the difficulties of forecasting and synthesizing their motions. Since all manipu-

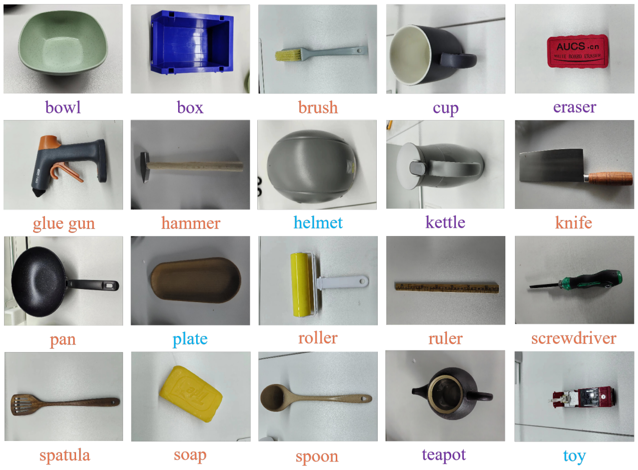

Figure 2. Visualization of 20 object categories in TACO, including 17 tool categories (shown in purple and brown) and 9 target object categories (shown in purple and blue).

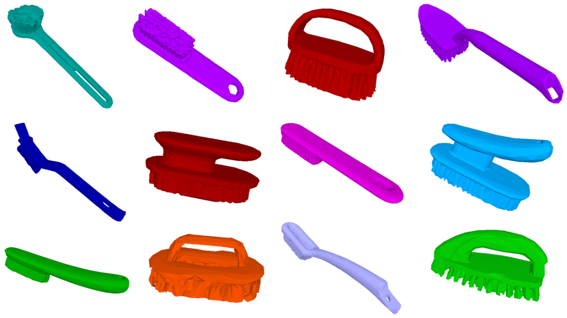

Figure 3. Visualization of 12 brushes in TACO.

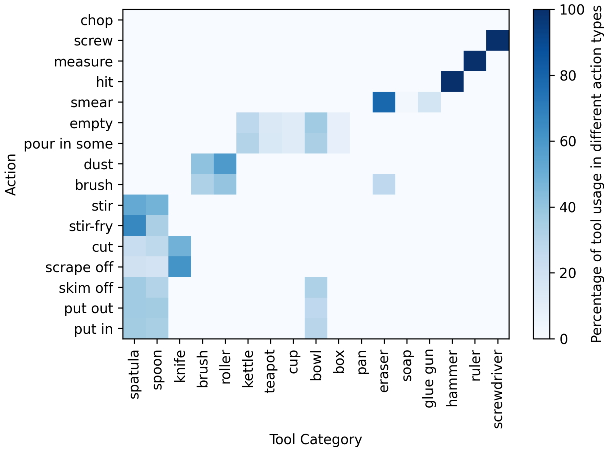

Figure 4. Percentage of tool usage for each action type.

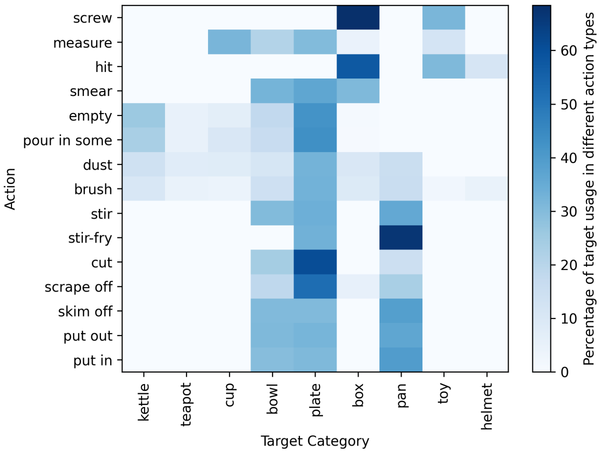

Figure 5. Percentage of target object usage for each action type.

lation behaviors are performed by right-handed individuals, the right hand commonly controls the tool and thus consistently moves faster than the left hand among different action types.

Hand pose distribution. Figure 6 illustrates the T-SNE visualization of hand poses from TACO and HO3D [6]. The distribution of hand poses from TACO mostly differs from that of HO3D due to the different human behaviors.

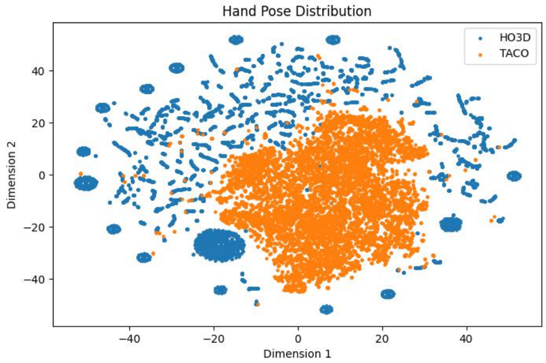

Figure 6. T-SNE visualization of hand poses from TACO and HO3D.

E. Details on Marker Removal Evaluation

Data processing. For each image from the raw videos and the marker-removed ones, we first render the spheres on the image to obtain their 2D mask and crop an image patch with the boundary the same as the mask. We then scale the image patch in equal proportions and place it at the center of a 512x512 image with black background color. Finally, a Gaussian kernel with σ=1.0 is utilized to augment the 512x512 image as the network input.

Network training. For an image input $I \in \mathbb{R}^{512\times512\times3}$, a U-Net [14] is used to estimate the heatmap $H \in \mathbb{R}^{512\times512}$ that indicates the probability that each pixel belongs to the inpainted image regions. The loss function is the mean-square error comparing the estimated heatmaps against the ground truth ones. The network is trained by an Adam [8] optimizer with a learning rate of 5e-4.

F. Details on Evaluation Metrics

Evaluating compositional action recognition. Following existing action recognition work [11, 18], we use Top-1 Accuracy and Top-5 Accuracy to evaluate whether the ground truth action label appears in the top-1 or top-5 predictions with the highest probabilities presented by the method.

Evaluating generalizable hand-object motion forecasting. As a prediction task, motion forecasting approaches are assessed by measuring differences between predictions and ground truths. Following the line of human-object motion forecasting studies [1, 3, 15, 16], we represent a hand as a 3D skeleton $J \in \mathbb{R}^{21\times3}$ with 21 joints and use Mean Per Joint Position Error $J_e = \frac{1}{21M}\sum_{k=1}^{M}\sum_{i=1}^{21}\|\hat{J}_{k,i} - \bar{J}_{k,i}\|_2$ to measure hand predictions, where M is the number of predicted frames, $\hat{J}$ is hand skeleton predictions, and $\bar{J}$ is ground truth values. Since objects are rigid bodies, the translation error T_e and rotation error R_e are defined as:

$$\begin{aligned} T_e &= \frac{1}{M}\sum_{k=1}^{M}\|\hat{t}_k - \bar{t}_k\|_2, \\ R_e &= \frac{1}{M}\sum_{k=1}^{M}\arccos\frac{\mathbf{Tr}(\hat{R}_k^T\bar{R}_k) - 1}{2}, \end{aligned} \tag{12}$$

where $\hat{t} \in \mathbb{R}^3$ and $\hat{R} \in \mathbb{R}^{3\times3}$ are predicted object translation vectors and rotation matrices, $\bar{t}$ and $\bar{R}$ are ground-truth ones, and $\mathbf{Tr}$ denotes the trace of a matrix.

Evaluating cooperative grasp synthesis. As a generative task, the benchmark should examine the physical plausibility and reality of synthesized hand meshes. For assessing physical plausibility, the contact ratio (*Con. R*) indicates the proportion of results that are in contact with the tool, while the interpenetration volume (*Pen. V*) denotes the average volume that is occupied by both the generated hand and the tool and is computed by voxelizing hand-object meshes to 1mm cubes and counting the intersecting ones. The collision ratio (*Col. R*) examines conflicts between generated hands and the environment, computing the probability of results penetrating the target object and the left hand. To evaluate whether results are realistic, we first present an interaction feature extractor (Figure 7) that encodes hand-object vertices to a 64-dimensional feature f and obtain ground truth feature distribution $\bar{D} = \{\bar{f}_i\}$ by applying it to real interaction snapshots. We then replace the vertices of the right-hand mesh with those from synthesized ones and

Action	Right hand		Left hand		Tool		Target object	
	v_r(cm/s)	v_j(cm/s)	v_r(cm/s)	v_j(cm/s)	v(cm/s)	ω(°/s)	v(cm/s)	ω(°/s)
Put in	17.2(±11.5)	20.0(±13.2)	9.4(±10.5)	10.4(±11.8)	18.9(±20.9)	88.1(±116.3)	3.7(±6.6)	14.1(±35.3)
Put out	17.9(±11.6)	20.5(±13.0)	10.5(±10.5)	11.7(±11.7)	18.7(±20.9)	90.8(±129.1)	4.4(±7.5)	18.2(±67.0)
Skim off	15.4(±10.7)	18.0(±12.5)	9.4(±10.3)	10.2(±11.4)	17.1(±19.9)	70.8(±98.9)	3.7(±6.4)	15.0(±41.5)
Scrape off	13.4(±10.4)	15.4(±11.6)	9.6(±10.0)	10.7(±11.6)	16.7(±19.0)	69.5(±98.9)	4.0(±7.4)	19.0(±47.8)
Cut	13.7(±11.5)	15.5(±13.0)	8.4(±9.8)	9.5(±11.3)	15.4(±17.9)	70.1(±115.3)	3.1(±6.1)	13.3(±32.5)
Stir-fry	17.0(±12.2)	19.3(±13.3)	10.7(±9.7)	12.1(±11.7)	21.4(±20.5)	77.0(±88.2)	8.2(±15.5)	23.1(±63.7)
Stir	16.8(±12.6)	18.9(±13.7)	9.0(±10.0)	10.3(±11.2)	20.1(±20.3)	74.0(±114.6)	4.6(±7.4)	15.6(±29.5)
Brush	15.3(±12.2)	17.7(±14.5)	11.0(±10.4)	12.2(±11.9)	19.3(±22.3)	71.5(±167.6)	8.1(±11.4)	33.9(±55.6)
Dust	13.9(±9.5)	17.2(±11.4)	10.8(±10.4)	12.4(±12.1)	19.6(±21.9)	88.5(±121.1)	7.1(±10.6)	31.6(±49.7)
Pour in some	15.0(±13.0)	16.3(±14.2)	9.0(±10.8)	10.0(±12.7)	8.8(±13.2)	39.4(±77.9)	2.8(±5.2)	11.1(±27.4)
Empty	14.9(±13.6)	16.1(±14.7)	8.5(±10.6)	9.4(±12.5)	10.2(±14.2)	45.5(±83.49)	2.9(±5.4)	12.3(±40.0)
Smear	12.4(±12.5)	15.8(±15.2)	9.9(±11.4)	11.4(±12.3)	16.5(±19.5)	60.1(±161.6)	7.4(±10.8)	32.7(±59.5)
Hit	14.1(±11.1)	17.4(±13.1)	9.3(±9.6)	9.9(±9.9)	18.7(±23.9)	67.6(±114.1)	5.7(±10.1)	21.9(±40.0)
Measure	15.3(±11.3)	16.0(±12.4)	15.9(±13.8)	17.8(±16.7)	12.0(±14.8)	56.9(±102.0)	2.0(±5.6)	10.6(±24.2)
Screw	12.5(±11.7)	15.7(±13.0)	10.8(±12.2)	11.7(±13.0)	11.6(±18.5)	182.5(±278.0)	4.8(±8.7)	20.7(±37.1)
Overall	15.0(±11.7)	17.2(±13.4)	9.9(±10.6)	11.1(±12.1)	16.6(±19.9)	71.3(±125.9)	5.0(±8.8)	20.9(±46.9)

Table 2. Average hand and object motion speed for each specific action type.

obtain another feature distribution $\hat{D} = \{\hat{f}_i\}$. Finally, the Fréchet Inception Distance (FID) score (*FID*) is computed on $\hat{D}$ and $\bar{D}$ measuring the dissimilarity between them. The interaction feature extractor is supervised-trained by decoding the one-hot action label from f via a fully connected layer.

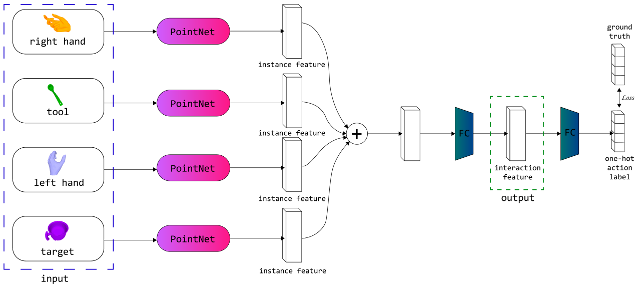

Figure 7. Network structure of our interaction feature extractor. Given vertices of hand-object meshes, the network first utilizes PointNet [10] to encode vertices of each mesh to a 128-dimensional feature, respectively, and then adds the four features together and acquires the interaction feature via a fully connected layer.

G. Baseline Designs for Interactive Grasp Synthesis

We modify baseline approaches [7, 9] to integrate the interaction environment (the left hand and the target object) into the network structure. We directly regard the interaction environment as additional conditions for CVAE, and apply existing point cloud encoders to transfer its point clouds to feature vectors. Figure 8 compares our modified HALO-VAE [7] structure with HALO-VAE^{-}.

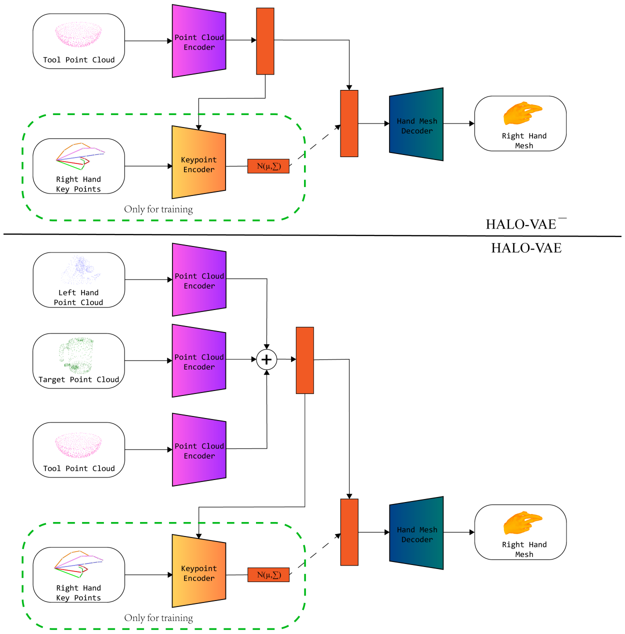

Figure 8. Comparison of HALO-VAE^{-} and our modified HALO-VAE [7].

H. Qualitative Results on Hand-object Motion Forecasting

Figure 9 shows the qualitative results of CAHMP [3]. Although CAHMP achieves the best performance among the four baseline methods, it commonly fails to forecast fast movements (Figure 9 (a),(b)) from the right hand and the tool, and encounters difficulty understanding human interaction intentions (Figure 9 (c),(d)). Please see our supplementary video for more visualizations.

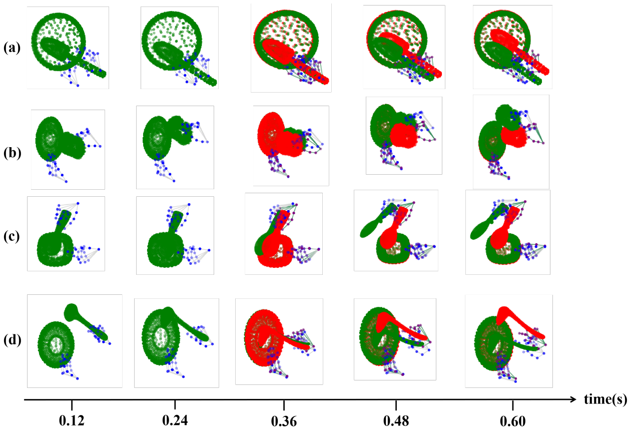

Figure 9. Qualitative results on hand-object motions predicted by CAHMP [3]. The green and blue points denote the ground truth, while the red and purple ones indicate the predicted motions. We show five frames at 0.12, 0.24, 0.36, 0.48, and 0.60s.

I. TACO Visualization

Figure 10 shows our 12 RGB frames from all third-person views and the RGB and depth images from our egocentric camera. Figures 11, 12, and 13 exhibit some examples of our hand-object meshes, hand-object segmentation, and marker-removed image patches, respectively. Please see our supplementary video for more data visualizations.

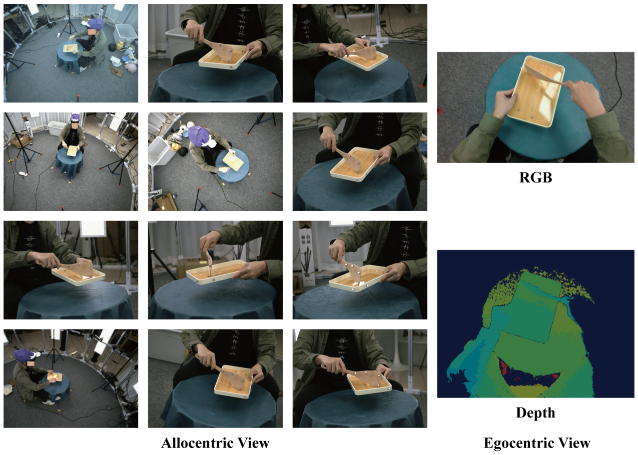

Figure 10. Visualization of allocentric and egocentric camera views. Our system involves 12 allocentric RGB cameras and one egocentric RGBD sensor.

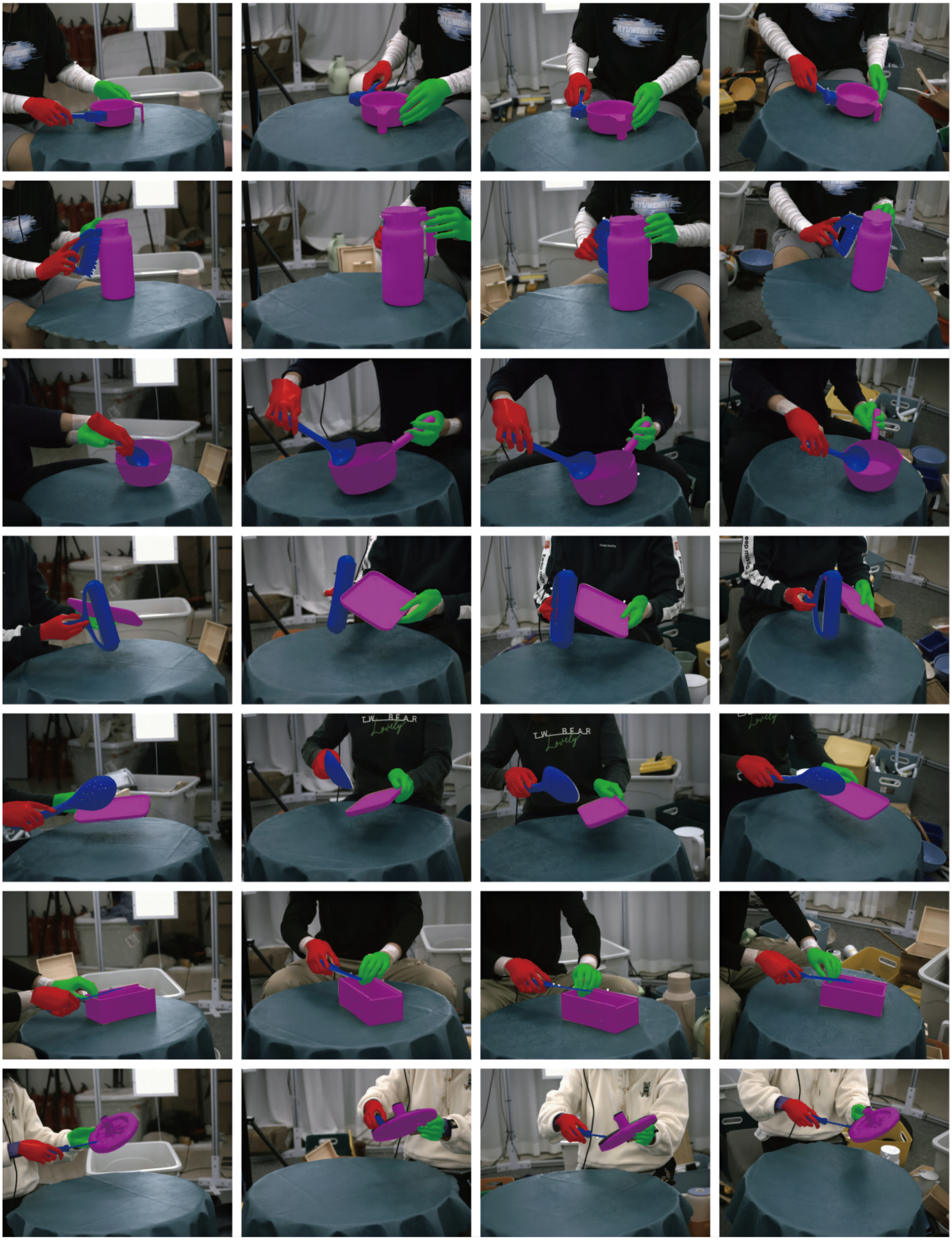

Figure 11. Visualization of hand-object meshes. We overlay the original color frames with rendered hand-object meshes.

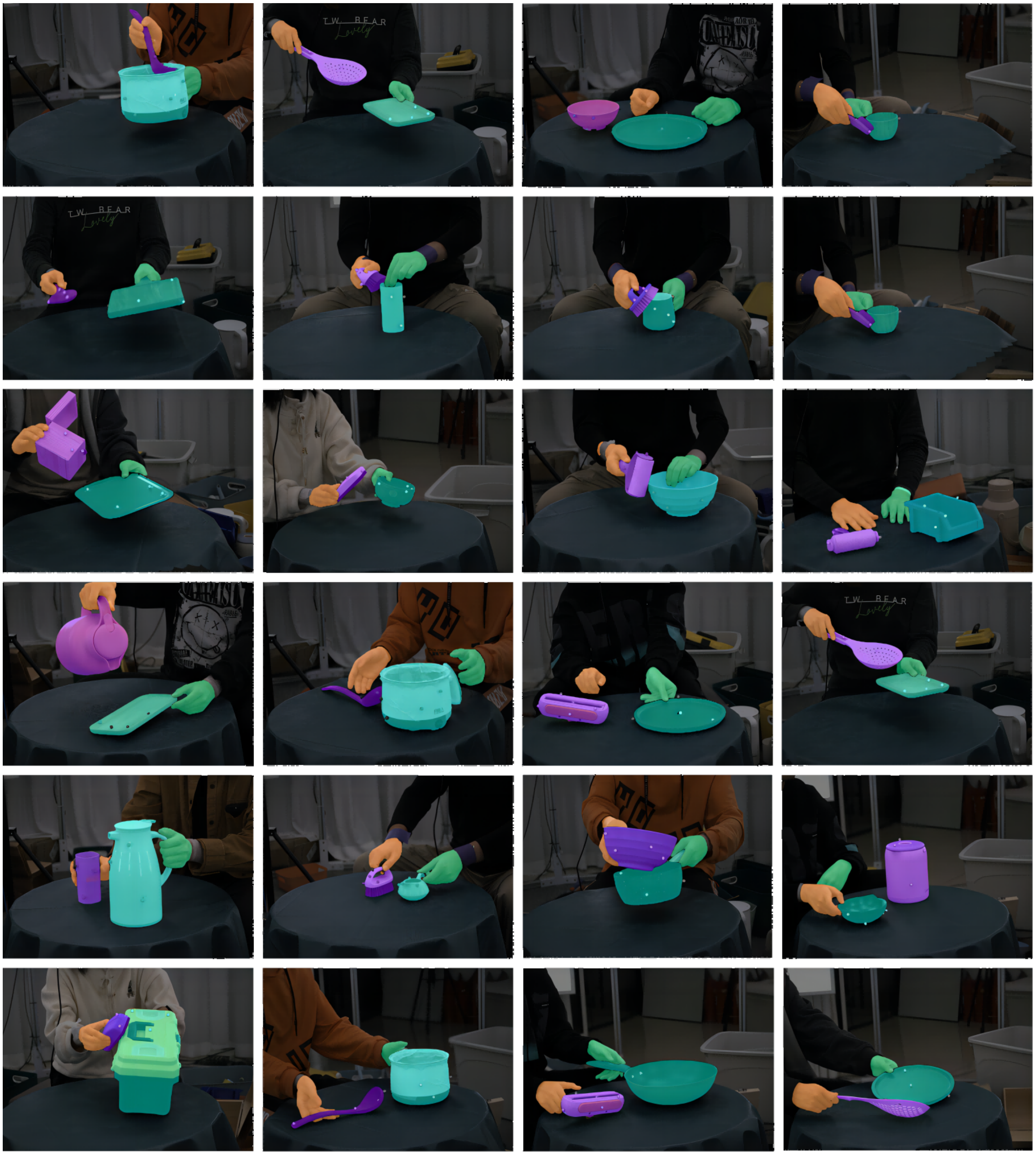

Figure 12. Visualization of automatic 2D hand-object segmentation.

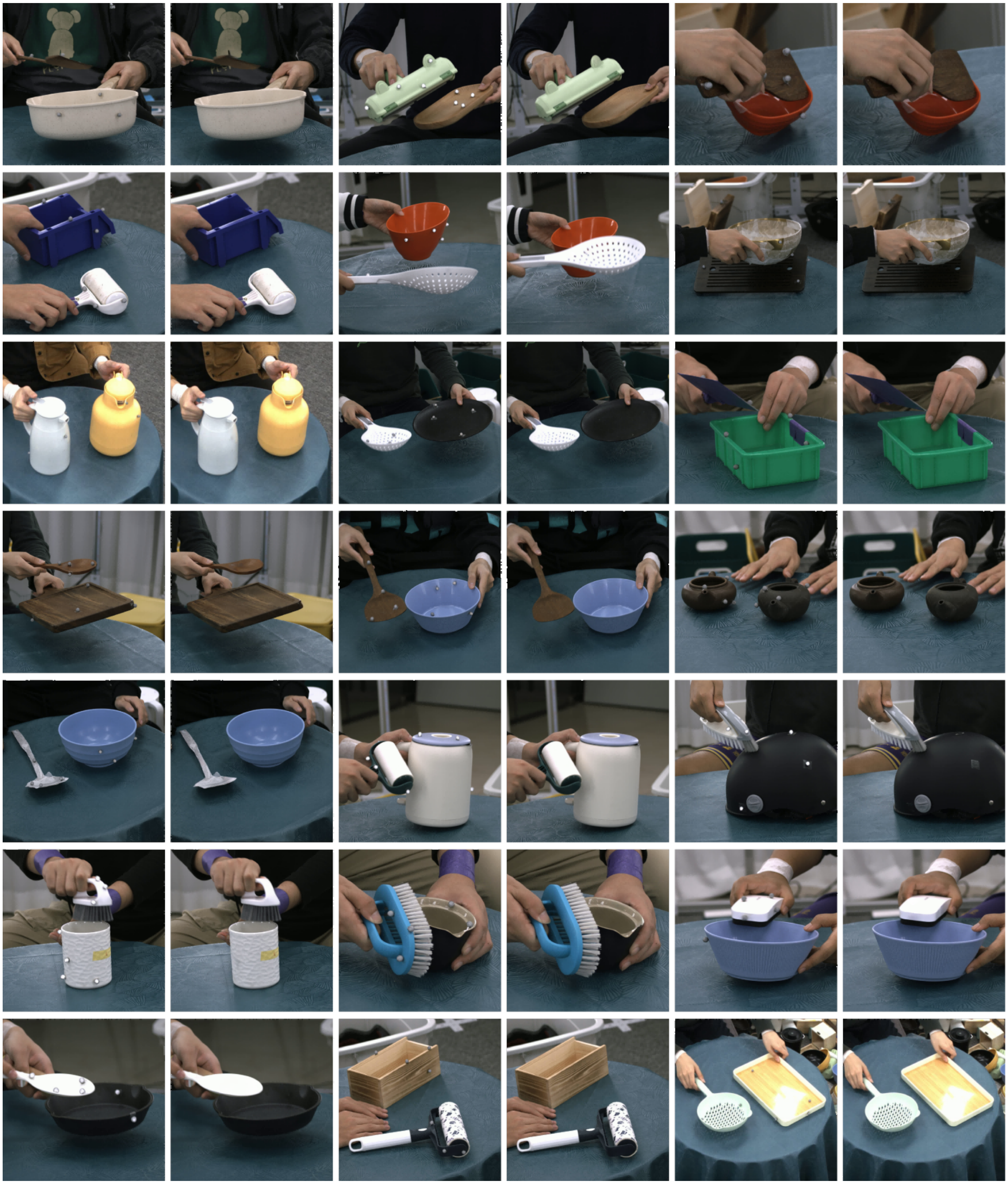

Figure 13. Visualization of original and marker-removed image patches.

References

[1] Vida Adeli, Mahsa Ehsanpour, Ian Reid, Juan Carlos Niebles, Silvio Savarese, Ehsan Adeli, and Hamid Rezatofighi. Tripod: Human trajectory and pose dynamics forecasting in the wild. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13390–13400, 2021. 5

[2] MMPose Contributors. Openmmlab pose estimation toolbox and benchmark. https://github.com/open-mmlab/mmpose, 2020. 2

[3] Enric Corona, Albert Pumarola, Guillem Alenya, and Francesc Moreno-Noguer. Context-aware human motion prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6992–7001, 2020. 5, 6, 7

[4] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12943–12954, 2023. 1, 2

[5] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981. 3

[6] Shreyas Hampali, Mahdi Rad, Markus Oberweger, and Vincent Lepetit. Honnotate: A method for 3d annotation of hand and object poses. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3196–3206, 2020. 4, 5

[7] Korrawe Karunratanakul, Adrian Spurr, Zicong Fan, Otmar Hilliges, and Siyu Tang. A skeleton-driven neural occupancy representation for articulated hands. In *2021 International Conference on 3D Vision (3DV)*, pages 11–21. IEEE, 2021. 6

[8] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[9] Shaowei Liu, Yang Zhou, Jimei Yang, Saurabh Gupta, and Shenlong Wang. Contactgen: Generative contact modeling for grasp generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 20609–20620, 2023. 6

[10] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017. 6

[11] Gorjan Radevski, Marie-Francine Moens, and Tinne Tuytelaars. Revisiting spatio-temporal layouts for compositional action recognition. *arXiv preprint arXiv:2111.01936*, 2021. 5

[12] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018. 2

[13] Javier Romero, Dimitrios Tzionas, and Michael J Black. Embodied hands: Modeling and capturing hands and bodies together. *arXiv preprint arXiv:2201.02610*, 2022. 1, 3

[14] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pages 234–241. Springer, 2015. 5

[15] Weilin Wan, Lei Yang, Lingjie Liu, Zhuoying Zhang, Ruixing Jia, Yi-King Choi, Jia Pan, Christian Theobalt, Taku Komura, and Wenping Wang. Learn to predict how humans manipulate large-sized objects from interactive motions. *IEEE Robotics and Automation Letters*, 7(2):4702–4709, 2022. 5

[16] Sirui Xu, Zhengyuan Li, Yu-Xiong Wang, and Liang-Yan Gui. Interdiff: Generating 3d human-object interactions with physics-informed diffusion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14928–14940, 2023. 5

[17] Jinyu Yang, Mingqi Gao, Zhe Li, Shang Gao, Fangjing Wang, and Feng Zheng. Track anything: Segment anything meets videos. *arXiv preprint arXiv:2304.11968*, 2023. 2

[18] Taojiannan Yang, Yi Zhu, Yusheng Xie, Aston Zhang, Chen Chen, and Mu Li. AIM: Adapting image models for efficient video action recognition. In *The Eleventh International Conference on Learning Representations*, 2023. 5

[19] Xingyi Zhou, Qingfu Wan, Wei Zhang, Xiangyang Xue, and Yichen Wei. Model-based deep hand pose estimation, 2016. 3