End-to-End HOI Reconstruction Transformer with Graph-based Encoding

Supplementary Material

This supplementary material elaborates on the proposed HOI-TG regarding methodology details and experimental results. Such information includes grid sampling and positional encoding, the construction of graph adjacency weight matrix for objects, and expanding loss functions. The results cover the comparison with StackFLOW [17], the variants of removing the annotated segmentation masks and a replacement by DetectronV2 [41] masks, and the location of graph residual blocks. Then, we comprehensively compare the computational efficiency and discuss the limitations of our HOI-TG. To further validate the effectiveness of our method, we present more qualitative comparisons in the end. Along with this material, we also provide the code for reproducibility.

6. More details of our HOI-TG

6.1. Grid sampling and positional encoding

We introduce grid sampling and positional encoding to provide the encoder with more 3D informative input. First, we project the human joints $\mathbf{M}_j^{\text{init}}$, human mesh $\mathbf{M}_h^{\text{init}}$ and object mesh $\mathbf{M}_o^{\text{init}}$ obtained in the init stage onto the 2D image plane using the camera parameters predicted in the init stage. Then, we apply bilinear interpolation on $\mathbf{F}$ to obtain the feature corresponding to each vertex. Finally, we concatenate the interpolated feature and the 3D coordinates of each vertex to produce our 3D queries.

6.2. Graph adjacency matrix for objects

For the adjacency matrix of objects with different templates, assuming that the 3D coordinate point of an object template is $P \in \mathbb{R}^{n\times 3}$, we first calculate the distance $d(p_i, p_j)$ between each two points,

$$d(p_i, p_j) = \|p_i - p_j\|_2 \tag{5}$$

Then for each point, we select the K points with the closest distance as its neighbors and calculate the un-normalized adjacency matrix $\bar{\mathbf{A}}^{'}$,

$$\bar{\mathbf{A}}^{'}(i,j) = \begin{cases} d(p_i, p_j) & \text{if } p_j \text{ is one of the } K \text{ neighbors of } p_i, \\ 0 & \text{otherwise.} \end{cases} \tag{6}$$

K represents the pre-set number of neighbors. We use the distance as the weight for neighbor nodes and 0 for non-neighbor nodes. Finally, $\bar{\mathbf{A}}^{'}$ is normalized to obtain the adjacency matrix $\bar{\mathbf{A}}$.

6.3. Loss functions and implementation details.

Our loss function consists of three parts, which are $\mathcal{L}_{\text{human}}$, $\mathcal{L}_{\text{hbox}}$ focusing on human reconstruction, and $\mathcal{L}_{\text{object}}$ focusing on object reconstruction:

$$\mathcal{L} = \mathcal{L}_{\text{human}} + \mathcal{L}_{\text{object}} + \mathcal{L}_{\text{hbox}}. \tag{7}$$

The $\mathcal{L}_{\text{hbox}}$ represents the L1 loss of the predicted hand bounding box and GT. We follow previous works [30, 31] for the design.

$\mathcal{L}_{\text{human}}$ is defined as:

$$\mathcal{L}_{\text{human}} = \mathcal{L}_{\text{human}}^{\text{ms-vertex}} + \mathcal{L}_{\text{human}}^{\text{param}} + \mathcal{L}_{\text{joint}} + \mathcal{L}_{\text{edge}}. \tag{8}$$

Human vertex multi-scale loss $\mathcal{L}_{\text{human}}^{\text{ms-vertex}}$: We upsample $\mathbf{M}_h \in \mathbb{R}^{431\times 3}$ twice to get $\mathbf{M}_h^{*} \in \mathbb{R}^{1723\times 3}$ and $\mathbf{M}_h^{**} \in \mathbb{R}^{6890\times 3}$. $\mathcal{L}_{\text{human}}^{\text{ms-vertex}}$ represents the L1 loss between the multi-scale human vertices($\mathbf{M}_h$, $\mathbf{M}_h^{*}$ and $\mathbf{M}_h^{**}$) and GT vertices.

Human parameters loss $\mathcal{L}_{\text{human}}^{\text{param}}$: $\mathcal{L}_{\text{human}}^{\text{param}}$ represents the L1 loss between the predicted parameters (human body mesh θ_{body} and human hand mesh θ_{hand}) in the init stage and GT parameters.

Human joint loss $\mathcal{L}_{\text{joint}}$: Our model has three joint outputs in total: i) the 3D joint predicted in the init stage, ii) the init 3D joint and init 2D joint obtained by the SMPLH model through the human parameters predicted in the init stage, iii) the 3D joint and 2D joint coordinates we reconstruct with the transformer. We all utilize L1 loss to minimize the loss between them and the corresponding GT.

Human edge length consistency loss $\mathcal{L}_{\text{edge}}$: $\mathcal{L}_{\text{edge}}$ is the L1 loss between up-sampled predicted human mesh $\mathbf{M}_h^{**}$ edges and GT edges.

$\mathcal{L}_{\text{object}}$ is defined as:

$$\mathcal{L}_{\text{object}} = \mathcal{L}_{\text{object}}^{\text{vertex}} + \mathcal{L}_{\text{object}}^{\text{param}}. \tag{9}$$

Object vertices loss $\mathcal{L}_{\text{object}}^{\text{vertex}}$: $\mathcal{L}_{\text{object}}^{\text{vertex}}$ is the L1 loss between the reconstruction object vertices $\mathbf{M}_o$ and GT.

Object parameters loss $\mathcal{L}_{\text{object}}^{\text{param}}$: $\mathcal{L}_{\text{object}}^{\text{param}}$ is the L1 loss between the init object parameters ($\mathbf{R}_{\text{init}}$ and $\mathbf{T}_{\text{init}}$) and GT.

Implementation details. We train our HOI-TG framework using the Adam optimizer with an initial learning rate of 1×10^{-4} for both the transformer and the ResNet50 backbone. The pipeline is trained for 50 epochs, with the learning rate decaying by 0.1 after 30 epochs. All transformer weights are randomly initialized, except that the ResNet backbone is initialized with weights from Hand4Whole [30]. We set the mini-batch size to 16 on an NVIDIA A100 80GB GPU.

Method	$CD_{human}\downarrow$	$CD_{object}\downarrow$	$Contact_p\uparrow$	$Contact_r\uparrow$
StackFLOW (w/o post-optim.)	5.98	12.6	0.429	0.521
StackFLOW (w post-optim.)	6.27	11.5	0.465	0.538
Ours	**4.87**	**7.49**	**0.647**	**0.539**

Table 5. Comparison with StackFLOW [17] on the BEHAVE [2] dataset. We use the officially released checkpoints by StackFLOW [17] for the comparison.

7. More experimental results

7.1. Comparison with StackFLOW

Table 5 presents a comparison between our HOI-TG and StackFLOW [17]. StackFLOW infers the posterior distribution of spatial relationships between people and objects from the input image and utilizes GT offsets to optimize their positions and postures during the inference stage, namely post-optimization. Although StackFLOW provides experimental results on the BEHAVE dataset, it splits the dataset differently from other approaches [31, 42, 44]. Specifically, StackFLOW samples more instances for testing within the BEHAVE dataset's test set. For a fair comparison, we report results on the intersection of its test split and the generally used test set. Our method outperforms StackFLOW, regardless of whether it includes the post-optimization stage. The results show the effectiveness of our implicit contact modeling over the explicit human-object offset constraint.

7.2. Segmentation

Following CONTHO [31], we use human and object segmentations provided by the datasets as the inputs. For completeness, we investigate the necessity of the segmentation masks. As shown in Tab. 6, without segmentation, the reconstruction results for both humans and objects are subpar, particularly for objects. Our analysis reveals that this is primarily due to information interference from the background image. When the object's color closely matches the background, the model struggles to accurately discern the object's depth position, which significantly hinders human-object interaction (HOI) reconstruction. By incorporating segmentation, the model can more effectively identify the relative depth positions of humans and objects, resulting in improved reconstruction accuracy.

	$CD_{human}\downarrow$	$CD_{object}\downarrow$	$Contact_p\uparrow$	$Contact_r\uparrow$
CONTHO	4.99	8.42	0.628	0.496
CONTHO (w/o seg.)	6.16	19.23	0.440	0.348
Ours	**4.59**	**8.00**	**0.662**	**0.554**
Ours (w/o seg.)	5.67	19.39	0.473	0.446

Table 6. Ablation study of segmentation in the inputs.

Except using the segmentation masks provided by the datasets, we can also extract the masks with off-the-shelf segmentation models such as DetectronV2 [41]. This way, the only input will be the RGB image. We evaluate the extra time cost of obtaining the masks by DetectronV2 in Tab. 7. After incorporating human and object segmentation into the pipeline, the reasoning time slightly increases from 0.208 to 0.264 seconds. Benefiting from the extracted masks, our model achieves much better HOI reconstruction results than the 'w/o seg.' variant. Since the extracted segmentations have not undergone manual correction, some inaccuracies account for a slight decrease in global mesh reconstruction.

	$CD_{human}\downarrow$	$CD_{object}\downarrow$	$Contact_p\uparrow$	$Contact_r\uparrow$	Time(s)
Ours	**4.59**	**8.00**	0.662	**0.554**	**0.208**
Ours (w Det.)	4.66	8.10	**0.664**	0.550	0.264

Table 7. Comparison of segmentations provided by the dataset and extracted by DetectronV2.

7.3. Location of graph residual blocks

We integrate the Human Graph Residual Block in all three transformer encoder blocks. As for the proposed Object Graph Residual Block, we investigate the optimal location in Tab. 8. The results indicate that: i) Incorporating the object graph residual block at any layer positively contributes to human and object reconstruction. ii) Adding a graph convolutional network (GCN) to the first and second blocks yields more significant improvements in reconstruction. This suggests that the self-attention mechanism at higher layers struggles to distinguish clear boundaries between human and object features. Therefore, our HOI-TG only equips the second transformer encoder block with the Object Graph Residual Block.

8. Generalization to in-the-wild images.

For generalization, Fig. 7 shows the estimated HOI of InterCap samples using HOI-TG trained on the BEHAVE dataset and applying HOI-TG to images in the wild. For the first row, we use the model trained on the BEHAVE [2] dataset directly for testing on the InterCap [16] dataset. For the second row, we directly use the model trained on the BEHAVE [2] dataset to test the reconstruction results of in-the-wild images.The results indicate that HOI-TG possesses a certain degree of generalization ability. However, given that

Human Graph Residual Block			Object Graph Residual Block						
Block1	Block2	Block3	Block1	Block2	Block3	$CD_{human}\downarrow$	$CD_{object}\downarrow$	$Contact_p\uparrow$	$Contact_r\uparrow$
✗	✗	✗	✗	✗	✗	4.73	8.55	0.606	0.559
✓	✓	✓	✗	✗	✗	4.61	8.11	0.651	0.539
✓	✓	✓	✓	✗	✗	4.62	8.01	0.638	**0.591**
✓	✓	✓	✗	✓	✗	**4.59**	**8.00**	**0.662**	0.554
✓	✓	✓	✗	✗	✓	4.68	8.43	0.643	0.534
✓	✓	✓	✓	✓	✓	4.62	8.05	0.644	0.573

Table 8. Ablation study of the location of Object Graph Residual Block.

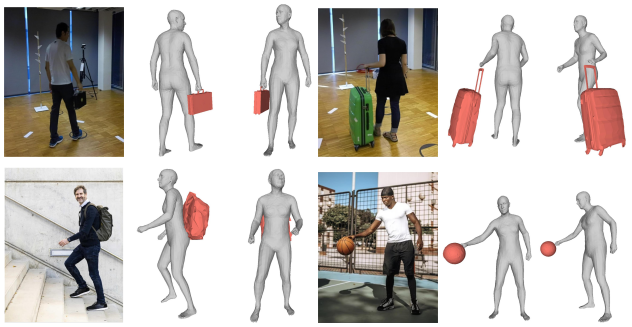

Figure 7. Estimated HOI of InterCap samples using HOI-TG trained on BEHAVE (row 1) and of images in the wild (row 2).

our method is data-driven, the generalization capability of HOI reconstruction in complex scenes still requires further investigation.

9. Efficiency and limitations

9.1. Computational efficiency

We summarize the parameters, running time, and performance of different methods in Tab. 9. The data related to PHOSA [47] and CHORE [42] are provided by CONTHO [31] and StackFLOW [17]. For StackFLOW [17], CONTHO [31], and our HOI-TG, we evaluate the speed using the same environment on a single RTX 4090 GPU, employing a batch size of 1 for multiple inferences and calculate the average running time. We only consider the inference time, excluding data preprocessing. According to Tab. 9, our HOI-TG outperforms previous methods in human-object interaction reconstruction while achieving the fastest inference speed. Both the optimization-based method PHOSA [47] and the neural field-based model CHORE [42] require significantly longer inference time. The post-optimization process of StackFLOW [17] also consumes quite a lot of time. In contrast, our HOI-TG can infer much faster and improves both global mesh reconstruction and local contact modeling. The results indicate that we have developed a more elegant and powerful architecture than previous approaches.

Method	Params(M)	Time(s)	Chamfer Dist↓	F1 Score↑
PHOSA	-	165.3	19.395	0.317
CHORE	18.19	312.2	8.120	0.523
StackFLOW	83.43	15.67	8.885	0.499
CONTHO	82.80	0.218	6.705	0.554
Ours	122.81	**0.208**	**6.295**	**0.603**

Table 9. Comparison of model efficiency and performance on BEHAVE [2] dataset.

9.2. Limitations

This section discusses cases where our HOI-TG fails to produce satisfactory reconstruction results and analyze the reason.

Lying poses: Our model may not perform well on complex or rare postures like lying down. As shown in the first row of Fig. 8, our model cannot accurately predict the posture of arms and legs and even cause mesh penetration. We consider that is mainly because actions such as lying down cause most body parts to be self-occluded. Such self-occlusion poses a big challenge for the ResNet50 backbone in producing meaningful initial human mesh. As a result, our model may fail to distinguish among different human vertices and predict inaccurate human mesh and object pose.

Purely symmetric objects: Accurately reconstructing highly symmetrical objects such as spherical and square has always been challenging in HOI reconstruction. The difficulty of capturing the object details may result in inaccurate rotation prediction. As shown in the second row of Fig. 8, our model cannot accurately estimate the correct rotation posture of the yoga ball.

10. More visual comparison results

We provide more HOI reconstruction results in the BEHAVE [2] and InterCap [16] datasets in Figs. 9 and 10. Regarding complex interactive actions, HOI-TG surpasses

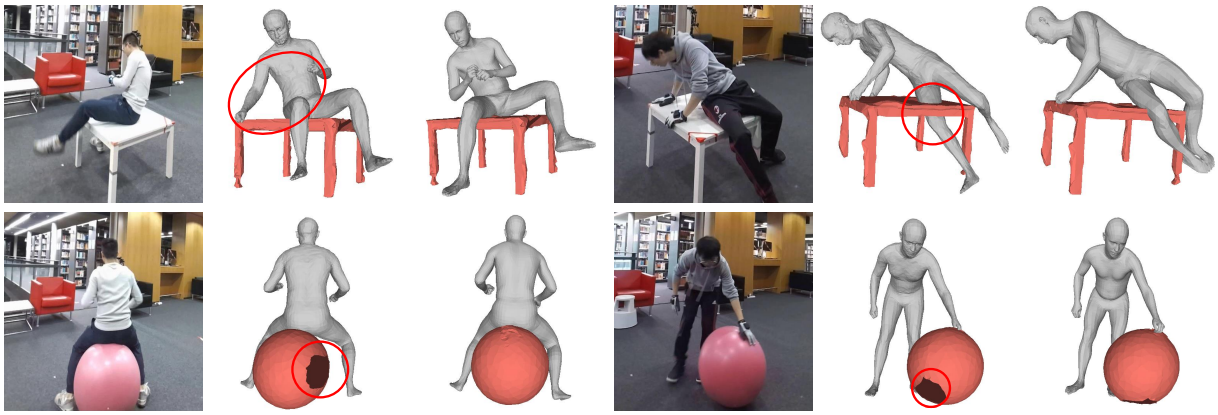

Figure 8. Failure cases of our HOI-TG. We highlight the region with red circles.

CONTHO [31] in both human mesh reconstruction and object posture estimation. Our model demonstrates an advantage in dealing with mesh penetration and inaccurate object posture. It also achieves higher reconstruction accuracy for human-object interactions without physical contact.

Figure 9. Qualitative comparison of 3D human and object reconstruction with CONTHO [31] on BEHAVE [2] dataset.

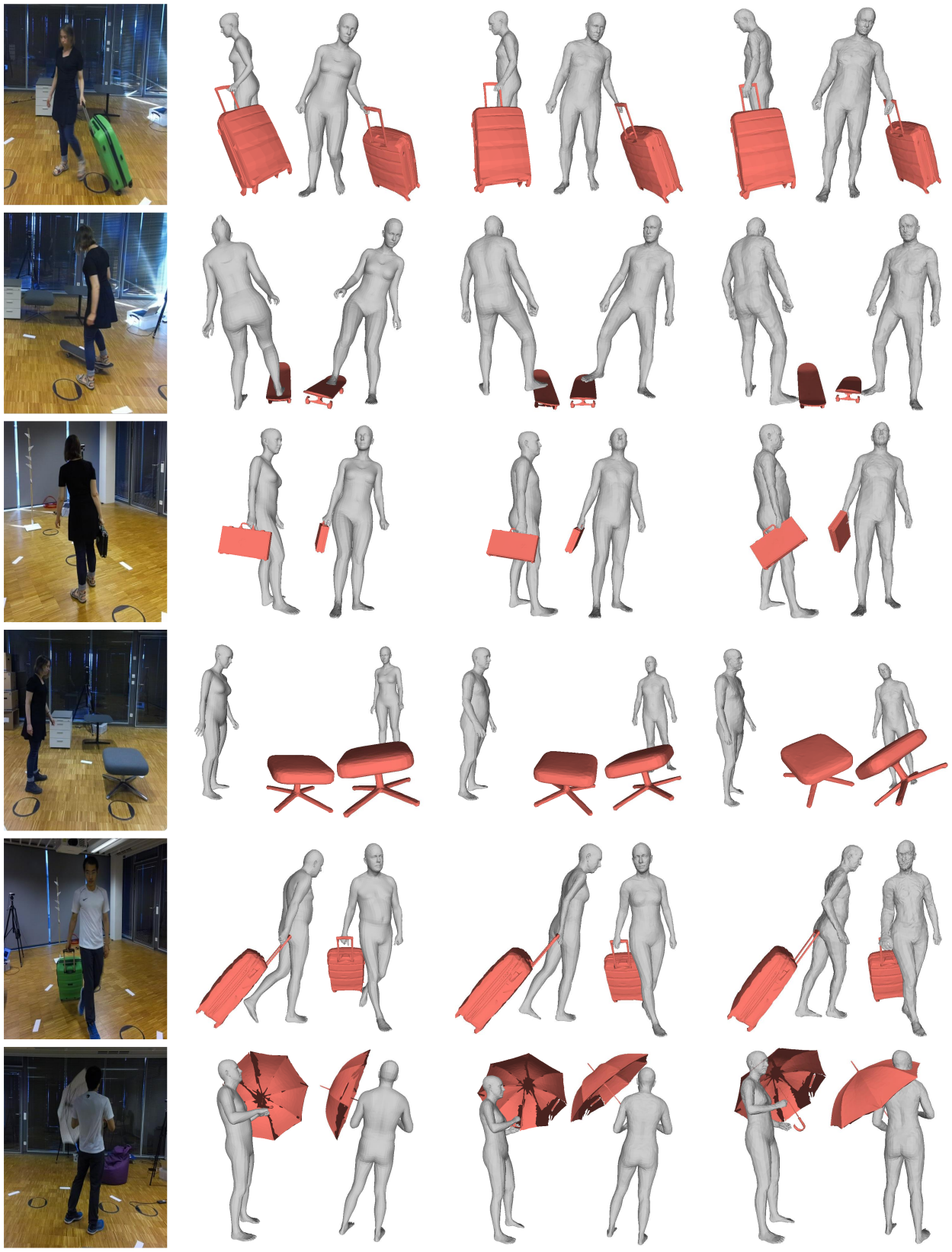

Figure 10. Qualitative comparison of 3D human and object reconstruction with CONTHO [31] on InterCap [16] dataset.