PICO: Reconstructing 3D People In Contact with Objects
Supplementary Material

Here we provide a detailed description of the PICO-db data collection (Sec. S.1), including implementation details, data statistics and quality control. In Sec. S.2, we discuss how we leverage vision large language models (VLLMs) to augment body contact prediction from DECO. In Sec. S.3, we provide implementation details about our PICO-fit reconstruction method, along with its quantitative evaluations and perceptual study. Finally, in Sec. S.4, we provide additional ablations, qualitative results and failure cases.

Video on our website. To crowd-source 3D contact annotation on both the body and the object using Amazon Mechanical Turk (AMT), we build a new annotation tool which we describe in detail in the following section. We recommend that readers view the provided supplemental video for an in-depth tour of our tool, its features and the annotation protocol. This is the same video we used for training AMT workers before qualifying them for this task.

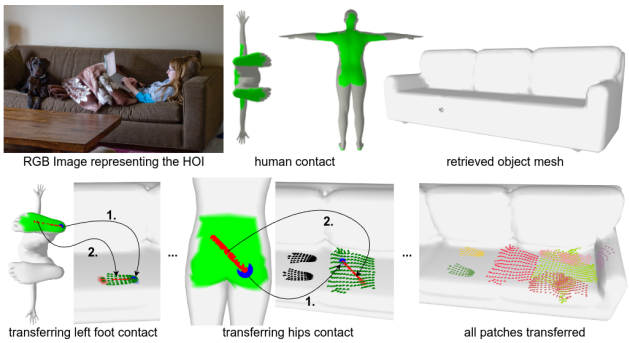

Figure S.1. **Top row:** Inputs to the PICO-db app. (1) RGB image. (2) DAMON body contacts. (3) The retrieved object mesh. **Bottom row:** Contact transfer via 2 clicks from the foot (left) and hips (center) onto the object. The first click specifies the axis start location (blue ball), and the second one the axis direction (red line). **Bottom row (right):** Resulting patches after annotation.

S.1. PICO-db data collection

S.1.1. Contact representation & projection details

DAMON's body contacts form neighboring-vertex patches. To represent such a patch, we compute the axis-based parameterization of "ContactEdit" [5] via three steps: (1) We synthesize an "axis," i.e., an open curve on the body composed of piece-wise shortest geodesics between constituent surface points. (2) For each patch vertex, we compute its closest axis point via short-time heat diffusion [13], and (3) its logarithmic map (`logmap`), i.e., its geodesic distance and direction w.r.t. its associated axis point [7].

The `logmap` helps transferring patches across meshes. That is, given an axis, we can reconstruct the patch via the inverse operation, the exponential map (`expmap`). Thus, transferring patches boils down to transferring only the axis. The axis can be completely unpacked on any surface given only the starting location and direction of the first geodesic.

Simply put, this lets us transfer body contact patches onto an object with just two clicks, which define the axis start location and direction respectively. Crucially, this also defines bijective point correspondences between patches[1]. The axis parameterisation enables the automatic correspondence of discretised contact areas, comprising hundreds or even thousands of points, between the body and the object through an intuitive and substantially lower-dimensional representation. Figure S.1 and Fig. 3 in main illustrates several body contacts and their respective axes. For further details, we refer the reader to Lakshmipathy et al. [5].

[1]Formally, given a source and target patch, there exists a point mapping that is theoretically-guaranteed surjective, but empirically it is bijective [5].

S.1.2. Projection using the proxy mesh

Because transferring patches parameterized on non-convex shape regions can yield non-intuitive results, we construct a proxy SMPL mesh to "convexify" the hands and face features. Specifically, we take a convex hull of the hands and improve the triangulation via tangential smoothing [2] and Delaunay refinement [14] using the Geometry Central library [12]. The result (Fig. S.2) is a SMPL mesh with "webbed" hands and "smoothed-out" face features.

We first project DECO contacts from the original SMPL body to the proxy "convexified" body via closest point queries [11]. We then parameterize contacts on the proxy body, and last transfer these to objects. However, for visualization purposes, we present annotators with contacts on the original body with overlaid axes from the proxy body.

Note that these "convexified" meshes are used only as a "proxy tool" to ease defining "straight" contact axes. We do not use these later for 3D reconstruction, so the accuracy of reconstructions is not compromised.

S.1.3. Object mesh processing

We rely on the Objaverse-LVIS [3] dataset for retrieving object meshes. However, our contact parameterization and projection requires input meshes to be *manifold*, which is not true for several meshes in Objaverse-LVIS. Therefore, we perform a series of pre-processing operations to curate a database of manifold objects. Specifically, we use the PyMeshLab library [8] and apply Poisson-Disk sampling to generate 50k uniformly sampled surface points. Next, we perform Screened Poisson surface reconstruction and uni-

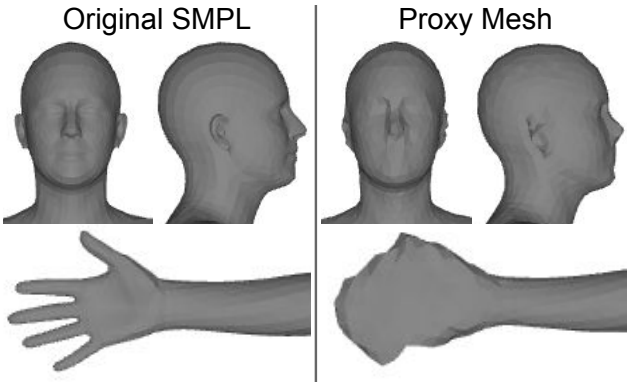

Figure S.2. Comparison of the face and hand details between the original SMPL human body mesh and the proxy mesh with "webbed" hands and "smoothed-out" face features, used for simplifying contact patch projections.

form mesh resampling on the resulting point cloud, with parameters: depth $= 8$ and samples-per-node $= 8$. We also remove floating isolated connected-component pieces which do not belong to the original mesh. This produces a smooth mesh, which is further corrected using the "3D Print Toolbox" [10] in Blender to ensure manifoldness. Last, we verify manifoldness and discard any non-manifold objects.

Although the processed meshes are manifold, they are often too high-resolution and arbitrarily scaled, making them unsuitable for online operations in our data annotation app. To address this, we decimate the meshes to a maximum of 4k faces using Blender. Additionally, we recenter the meshes to the origin and rescale them by querying GPT-4V to determine the correct scale based on the corresponding RGB image input.

S.1.4. Contact annotation tool

Following [15], we build the tool in DASH and deploy it inside a Docker container under an uWSGI application server.

Annotation interface. As shown in the tool interface in Fig. S.3, the layout is divided into 3 parts. On the left, we show annotators the original image with the "transfer" candidates denoted as tuples: {`body part name, object label`}. On the top-right, we show the human T-pose mesh with DAMON's contact regions (shown in green color) and contact axis (shown in red color). On the bottom-right, we show the 3D object.

The annotators are required to click two points on the object mesh – the first click specifies the start of the contact axis and the second click specifies its geodesic orientation. Upon registering the second click, the tool instantaneously displays the transferred contact on the object, providing visual feedback in real time. Annotators can correct errors by repeating the two clicks, which resets the prior annotations, until satisfied. For a detailed overview of the tool and its functions, please watch the video on our website.

S.1.5. PICO-db additional statistics

PICO-db contains 4123 images with paired human and object 3D contact. The images span 44 object categories. This is fewer than the 69 object categories in DAMON as we identify and reject object categories that are never (or rarely) in contact with humans in images, such as a wall clock, fire hydrant, plant, TV, etc. Additionally, we exclude objects that are too large, so they are severely truncated in images such as (sitting in an) airplane, boat, car, bus, train, etc. We also filter out images of children, since their smaller size would "compromise" contacts annotated on the bigger default-shape SMPL body. For the complete list of included object categories and their distribution refer to Fig. S.5. Note that we use the same train, validation and test splits as the DAMON dataset.

In Fig. Fig. S.4, we present the aggregate vertex-level contact distributions for six object categories: bed, bicycle, cell phone, handbag, pizza, and surfboard. These distributions illustrate that our dataset captures a wide range of interaction patterns, reflecting both frequent (canonical) and infrequent (rare or edge-case) usage scenarios. This diversity highlights the richness of human-object interactions in our dataset

S.1.6. Quality control

We adopt several strategies to ensure high-quality annotations in PICO-db. First, we select high-performing AMT annotators through a rigorous two-part qualification process. Specifically, annotators are required to (i) watch a detailed tutorial video (see video on our website), and (ii) complete test annotations on a standardized set of 10 sample images. We evaluate the annotator responses by computing the point-to-point Euclidean distance between their annotations and author-annotated *pseudo-ground-truth* (pseudo GT) labels, as performed in DAMON [15]. With this, we qualify 17 out of 150 participants. Second, we release annotation tasks in small batches and visually inspect the quality of contact annotations per batch. Annotations flagged as incorrect or low-quality are repeated in the next batch.

S.2. Leveraging VLLMs in PICO-fit

In this work, we exploit the general world knowledge of VLLMs in two ways (i) to initialize object scale in PICO-fit and (ii) to refine human-contact predictions from DECO.

For initializing object scale, we input the test image to GPT-4V and query the object's scale by using the following prompt:

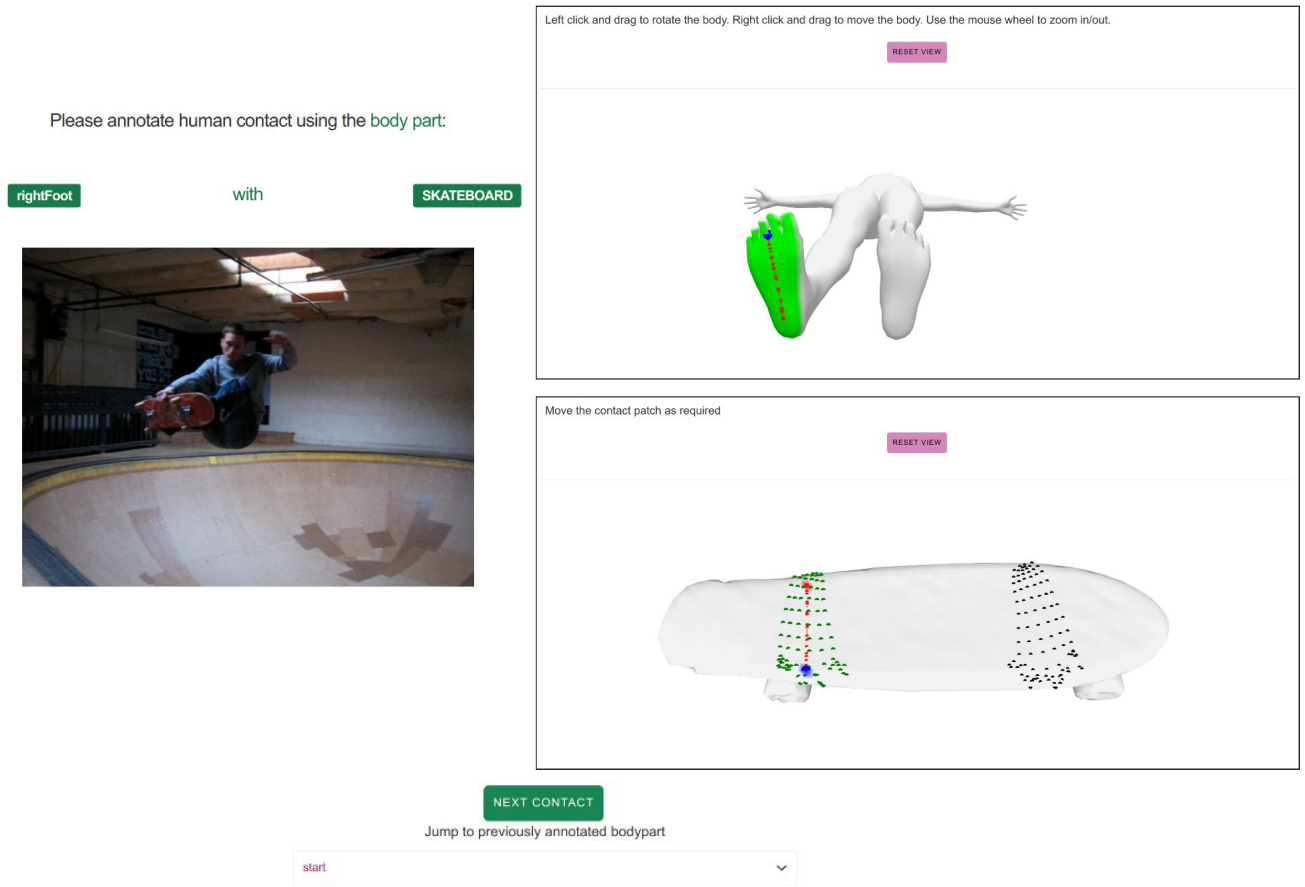

Figure S.3. Layout of the contact annotation tool. **Left side:** Original image with labels for the object category and the current body part above it. **Right side:** Human mesh in T-pose with contact regions (shown in green color) and contact axis (shown in red color). Below that, we show the 3D object, with the left foot contact already transferred and the right foot contact being annotated in the current step. Users can navigate back to previous steps (body parts) with the help of the drop-down menu situated below the "Next contact" button.

```
How big is the <OBJECT> in the <IMAGE>
that the human is interacting with?
Use the other objects and the scale of
the human to estimate the size.  Answer
should be single number, in meters, that
corresponds to the length of the longest
side of the <OBJECT>.
```

We also use GPT-4V to refine DECO's body contact predictions. Since PICO-fit relies on DECO predictions to retrieve contacts from PICO-db for both the body and the object, as well as the object shape, any errors in the estimated body contact may propagate to subsequent steps in the PICO-fit pipeline.

While DECO is robust and generalizes well to in-the-wild scenarios, it has a strong bias for predicting false positives on the feet, and often misses body parts in contact. To tackle this, we refine DECO's predictions by removing feet contact if it is not predicted by GPT-4V, and adding contact on any body parts that are additionally predicted by GPT-4V. To this end, we use the following prompt:

```
List the body parts of the human that are
in contact with the <OBJECT> (touching or
supporting the object) in this <IMAGE>.
These are all the body parts to
consider:  head, neck, torso,
hips, leftUpperArm, rightUpperArm,
leftForeArm, rightForeArm, leftHand,
rightHand, leftUpperLeg, rightUpperLeg,
leftLowerLeg, rightLowerLeg, leftFootSole,
rightFootSole, topOfLeftFoot,
topOfRightFoot.  Answer should be only
a comma-separated list of the body parts,
nothing else.
```

Impact of GPT-4V. DECO+GPT-4V contact is higher quality relative to DECO; F1 improves from 0.29 to 0.35 on InterCap. PA-CD$_{h+o}$ also improves, from 11.76 to 10.33. On object scale, GPT-4V yields 17.0 cm RMSE on InterCap. Fig. S.6 shows visual examples indicating the estimated size and contacting body parts from GPT-4V on

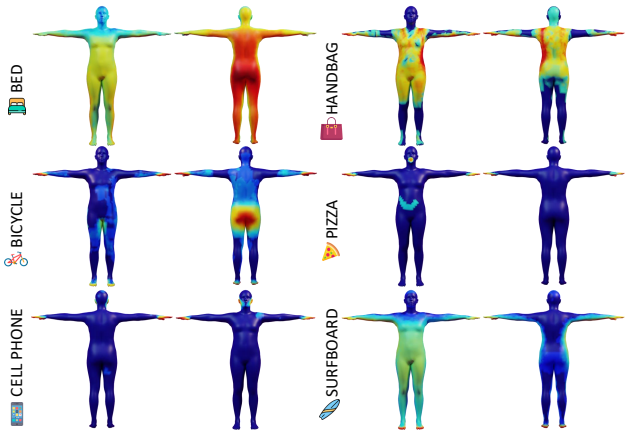

Figure S.4. Aggregate statistics showing object-wise contact probabilities across all body vertices in the PICO-db dataset. The body part closeups show the contact probabilities normalized for that body part. Red implies higher probability of contact while blue implies lower probability. **Zoom in**.

some challenging images with diverse objects. The object size refers to the length of the largest dimension.

S.3. PICO-fit additional details

S.3.1. Implementation details

We use the Adam optimizer, with parameter-specific learning rates ranging between 0.01 and 0.04, assigning higher rates to object rotation parameters to accelerate exploration of their search space.

Empirically, we find that reconstructions are most consistent with the following set of loss weights:

- for the second stage of the optimization we use weights: $\lambda_c = 4, \quad \lambda_p = 100, \quad \lambda_o^m = 0.4, \quad \lambda_o^s = 4$
- for the third stage of the optimization we use weights: $\lambda_c = 4, \quad \lambda_p = 50, \quad \lambda_h^m = 0.1, \quad \lambda_{\theta_C} = 0.05$

S.3.2. Quantitative evaluation details on InterCap

We evaluate on InterCap by reporting the Procrustes-Aligned (PA) Chamfer Distance (CD). Since state-of-the-art methods use different output formats, we standardize to ensure a fair evaluation.

While using the joint human and object mesh for alignment is standard practice [9], the Procrustes-alignment algorithm assumes higher weight on the human, since the human mesh has considerably more vertices than the object mesh in 3D HOI datasets [1, 4]. As a side-effect, this leads the PA-CD$_h$ to be often lower than PA-CD$_o$, which is evident in Tab. 1 in main.

For evaluating CONTHO [9] we use the authors' published code and annotation file. We adapt their evaluation code for all methods and use the same InterCap test split they release.

Stage IDs	$\mathcal{L}_c$	$\mathcal{L}_{o,m}$	$\mathcal{L}_p$	$\mathcal{L}_{h,m}$	CD$_h$ ↓	CD$_o$ ↓	CD$_{h+o}$ ↓
1 + 2 + 3	✗	✓	✓	✓	22.71	39.39	26.63
2 + 3 (no 1)	✓	✓	✓	✓	9.24	34.39	12.9
1 + (2&3 comb.)	✓	✓	✓	✓	8.13	19.1	10.66
1 + 2 + 3 (PICO-fit)	✓	✓	✓	✓	6.66	13.34	8.36

Table S.1. Additional ablations, extending Tab. 2 of the paper.

PHOSA [17] outputs SMPL meshes to represent the human pose and shape, which are inconsistent with the ground-truth SMPL-X meshes in InterCap. While we use the joint human and object mesh to Procrustes align predictions with ground-truth, in case of the human body, we exclude the head vertices. We do this as the body vertices share the same topology between SMPL and SMPL-X, whereas the head does not. After alignment, we sample the same number of points in both PHOSA predicted meshes as well as the ground-truth meshes in InterCap to compute chamfer distance.

The HDM [16] model trains on ProciGen [16], a synthetic dataset building on BEHAVE and InterCap. It outputs point clouds for both the human and the object in InterCap's "Cam1" coordinate frame. To compute the chamfer distance, we bring all ground-truth meshes from InterCap in the same "Cam1" coordinate frame and sample equal number of points (= 8192) as in the predictions, before aligning the predicted and ground-truth point clouds using Iterative Closest Point (ICP) and computing CD. Note that for alignment, we use the *combined* human and object point cloud. Unlike for Procrustes-alignment, since ICP uses the same number of points from the human and the object, HDM's chamfer distance scores are more balanced between the human and the object in Tab. 1 in main.

S.3.3. Perceptual study details

To ensure reliable participants, we repeat the first three images at the end of the study and exclude them during evaluation to serve as a warm-up for participants. We also include four catch trials—pairs of images with decisions that are intentionally straightforward—to identify and filter out participants who may provide random inputs. We exclude all submissions with even a single failed catch trial, which results in discarding 27 out of 100 total completions. For the layout we use in the perceptual study, see Fig. S.7.

S.3.4. Additional Ablations

We ablate $\mathcal{L}_c$ in Tab. S.1, top and bottom rows. Note that PICO-fit needs both human and object contact maps for $\mathcal{L}_c$, and hence, we cannot ablate them separately. Results show that $\mathcal{L}_c$ is essential for performance.

Next, we analyze alternative optimization strategies for PICO-fit. To evaluate the effect of Stage 1 which uses dense contact correspondences to initialize object pose w.r.t. the body, we run only Stage 2 and 3 and report results in Tab. S.1, second row. In Tab. S.1, third row, we first run Stage 1, followed by a joint optimization of Stage 2 and 3

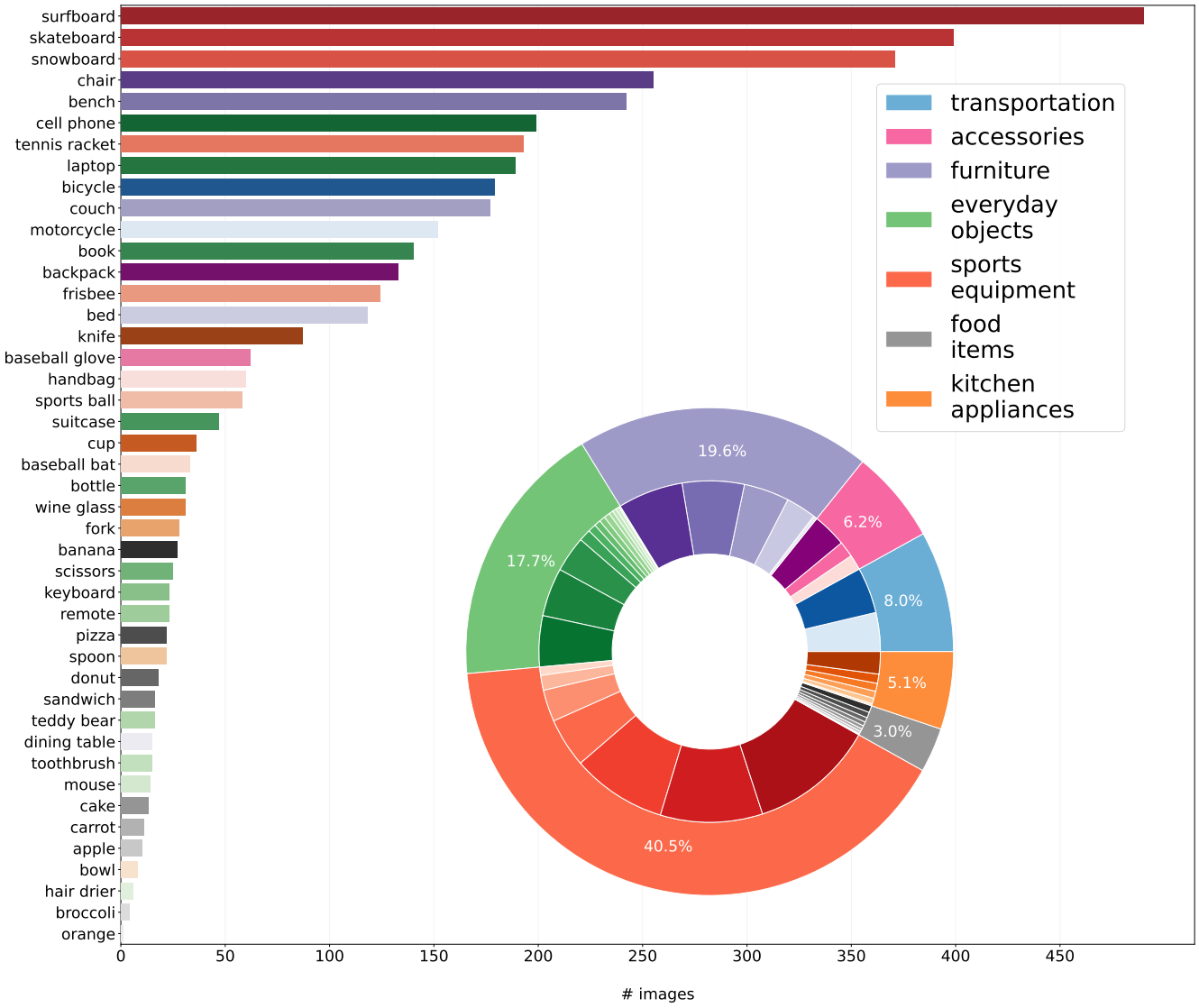

Figure S.5. Statistics on the object categories in PICO-db. **Histogram:** Object labels (y-axis) and the number of images in which they are present (x-axis). **Pie chart:** Object labels are grouped into 7 main categories; inner colors correspond to the colors in the histogram.

together. The results show that the proposed optimization scheme in PICO-fit significantly outperforms these alternatives, particularly for recovering accurate object poses.

The qualitative ablation in Fig. S.8 demonstrates the effect of each stage in PICO-fit. In Stage 1, PICO-fit establishes contact between the human and the object, though the object may not yet be aligned with the image. Stage 2 refines the object's alignment with the image, albeit at the cost of slight contact misalignment. Finally, Stage 3 optimally balances contact, interpenetration, and image alignment by refining the human's contacting limbs. This results in a 3D HOI that is both image-aligned in 2D and plausible in 3D.

S.3.5. Failure cases

Like all current methods, PICO-fit might fail under truly novel interactions if these differ significantly from those included in PICO-db. We show examples of PICO-fit failures under unusual contact scenarios in Fig. S.9, both due to inaccurate human contact (row 1-3) and object contact (row 4-5). Figure S.10 demonstrates additional PICO-fit failures caused due to incorrect human pose initialization (row 1) and incorrect object retrieval (row 2). Further, to develop understanding of PICO-fit failures, we randomly sampled 500 PICO-fit reconstructions and hired Master's students to categorize them into failure modes. Most PICO-fit failures result from: (1) Incorrect human pose initialization by OSX [6] (5/500), (2) incorrect object retrieval which does

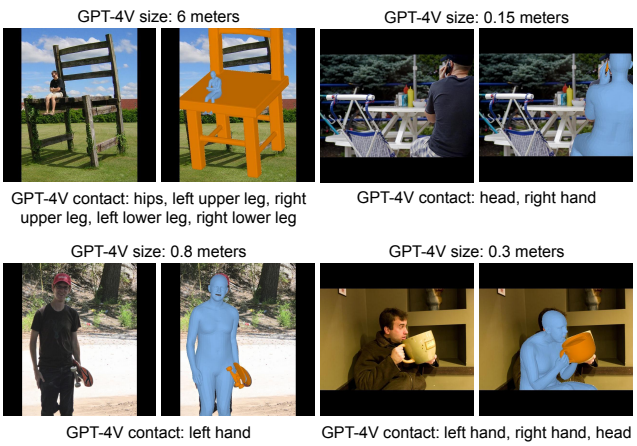

Figure S.6. Visual examples showing GPT-4V predicted object size and contacting body parts. "Size" implies length of the largest dimension.

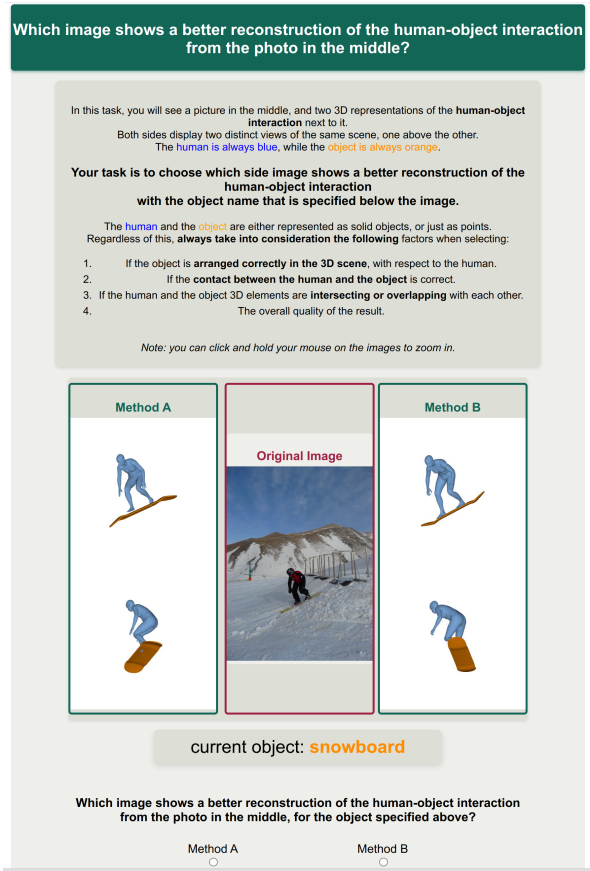

Figure S.7. Layout of the perceptual study. Below the extensive but simple instructions, participants are presented with two different views of the reconstructions from two methods (randomly swapped). Our interface correctly adapts to any screen size, but users are also able to click on the images to zoom in

not match the image $(12/500)$, (3) incorrect human-contact prediction by DECO $(85/500)$, and (4) invalid object contact retrieval even when the inferred human contact is correct $(20/500)$.

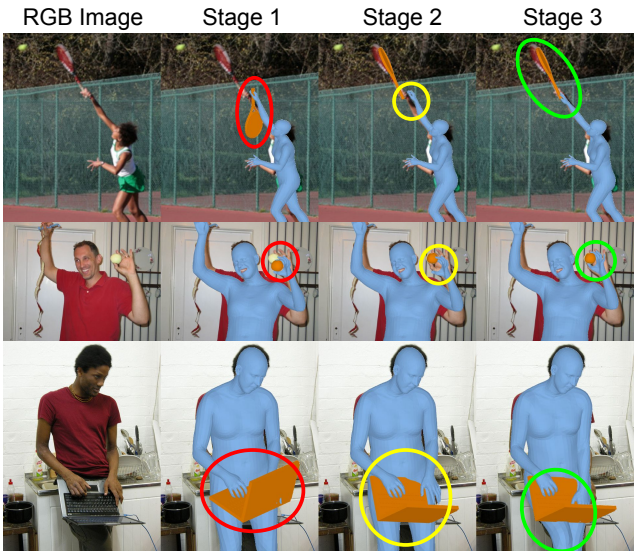

Figure S.8. Ablation study for PICO-fit's stages.

Despite the failures, we note that PICO-fit handles significantly more object instances than existing work; PICO-fit handles 627 objects, namely 1-2 orders of magnitude more than the 8 and 30 objects of PHOSA and CONTHO/HDM, respectively. Further, as shown in Tab. 1 and Figs. 7 and 8 in main, PICO-fit achieves SOTA performance on OOD and in-the-wild datasets, indicating superior generalization.

S.4. Additional qualitative results

Fig. S.11 shows qualitative comparisons of CONTHO, HDM and PHOSA* alongside PICO-fit and PICO-fit* for object categories handled by all baselines. Fig. S.12 shows HOI reconstructions from PICO-fit on various object categories, and Fig. S.13 does the same for PICO-fit*.

References

[1] 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. In *Computer Vision and Pattern Recognition (CVPR)*, pages 15935–15946, 2022. S.4

[2] Long Chen and Michael Holst. Efficient mesh optimization schemes based on optimal delaunay triangulations. *Computer Methods in Applied Mechanics and Engineering*, 200 (9):967–984, 2011. S.1

[3] Matt Deitke, Dustin Schwenk, Jordi Salvador, Luca Weihs, Oscar Michel, Eli VanderBilt, Ludwig Schmidt, Kiana Ehsani, Aniruddha Kembhavi, and Ali Farhadi. Objaverse: A universe of annotated 3D objects. In *Computer Vision and Pattern Recognition (CVPR)*, pages 13142–13153, 2023. S.1

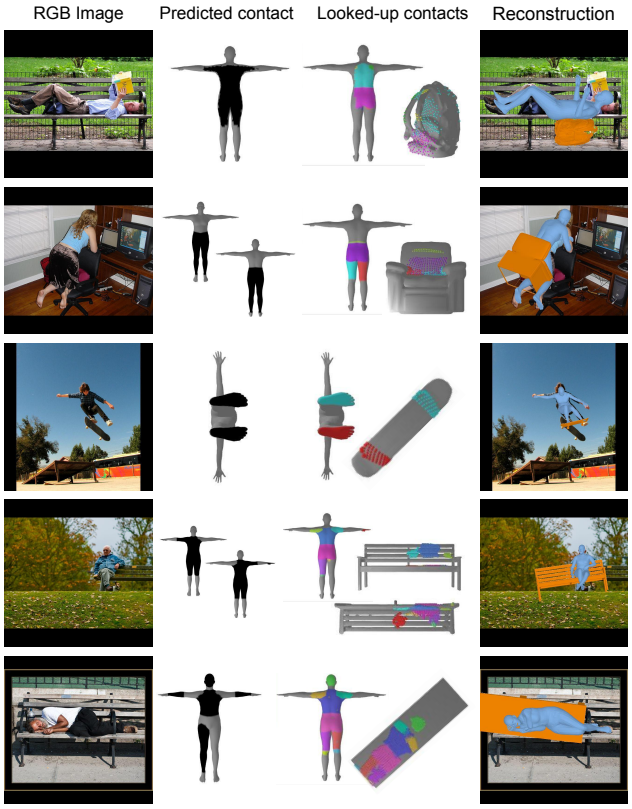

Figure S.9. Example interactions where PICO-fit lookups on PICO-db fail. Each row from left to right: input image, predicted body contact from DECO + GPT-4V, looked-up contact from PICO-db and 3D reconstructions overlaid on the images. Rows 1-3: incorrect human contact prediction. Rows 4-5: incorrect object contact retrieval.

Figure S.10. Failure cases of PICO-fit. Each row (from left to right) shows two input images and corresponding PICO-fit reconstructions overlaid on the image. Top row: incorrect human pose initialization. Bottom row: incorrect object retrieval.

[4] Yinghao Huang, Omid Taheri, Michael J. Black, and Dimitrios Tzionas. InterCap: Joint markerless 3D tracking of humans and objects in interaction from multi-view RGB-D images. *International Journal of Computer Vision (IJCV)*, 132(7):2551–2566, 2024. S.4

[5] Arjun S. Lakshmipathy, Nicole Feng, Yu Xi Lee, Moshe Mahler, and Nancy S. Pollard. Contact Edit: Artist tools for intuitive modeling of hand-object interactions. *Transactions on Graphics (TOG)*, 42(4), 2023. S.1

[6] Jing Lin, Ailing Zeng, Haoqian Wang, Lei Zhang, and Yu Li. One-stage 3D whole-body mesh recovery with component aware transformer. In *Computer Vision and Pattern Recognition (CVPR)*, pages 21159–21168, 2023. S.5

[7] Joseph S.B. Mitchell, David M. Mount, and Christos H. Papadimitriou. The discrete geodesic problem. *SIAM Journal on Computing*, 16(4):647–668, 1987. S.1

[8] Alessandro Muntoni and Paolo Cignoni. PyMeshLab, 2021. S.1

[9] Hyeongjin Nam, Daniel Sungho Jung, Gyeongsik Moon, and Kyoung Mu Lee. Joint reconstruction of 3D human and object via contact-based refinement transformer. In *Computer Vision and Pattern Recognition (CVPR)*, 2024. S.4

[10] Mikhail Rachinskiy. 3D Print Toolbox, 2024. S.2

[11] R. Sawhney. FCPW: Fastest closest points in the west. https://github.com/rohan-sawhney/fcpw, 2021. S.1

[12] Nicholas Sharp, Crane Keenan, et al. geometry-central.net, 2019. S.1

[13] Nicholas Sharp, Yousuf Soliman, and Keenan Crane. The vector heat method. *Transactions on Graphics (TOG)*, 38 (3), 2019. S.1

[14] Jonathan Richard Shewchuk. Delaunay refinement algorithms for triangular mesh generation. *Computational Geometry*, 22(1):21–74, 2002. 16th ACM Symposium on Computational Geometry. S.1

[15] Shashank Tripathi, Agniv Chatterjee, Jean-Claude Passy, Hongwei Yi, Dimitrios Tzionas, and Michael J. Black. DECO: Dense estimation of 3D human-scene contact in the wild. In *International Conference on Computer Vision (ICCV)*, pages 8001–8013, 2023. S.2

[16] Xianghui Xie, Bharat Lal Bhatnagar, Jan Eric Lenssen, and Gerard Pons-Moll. Template free reconstruction of human-object interaction with procedural interaction generation. In *Computer Vision and Pattern Recognition (CVPR)*, 2024. S.4

[17] Jason Y. Zhang, Sam Pepose, Hanbyul Joo, Deva Ramanan, Jitendra Malik, and Angjoo Kanazawa. Perceiving 3D human-object spatial arrangements from a single image in the wild. In *European Conference on Computer Vision (ECCV)*, pages 34–51, 2020. S.4

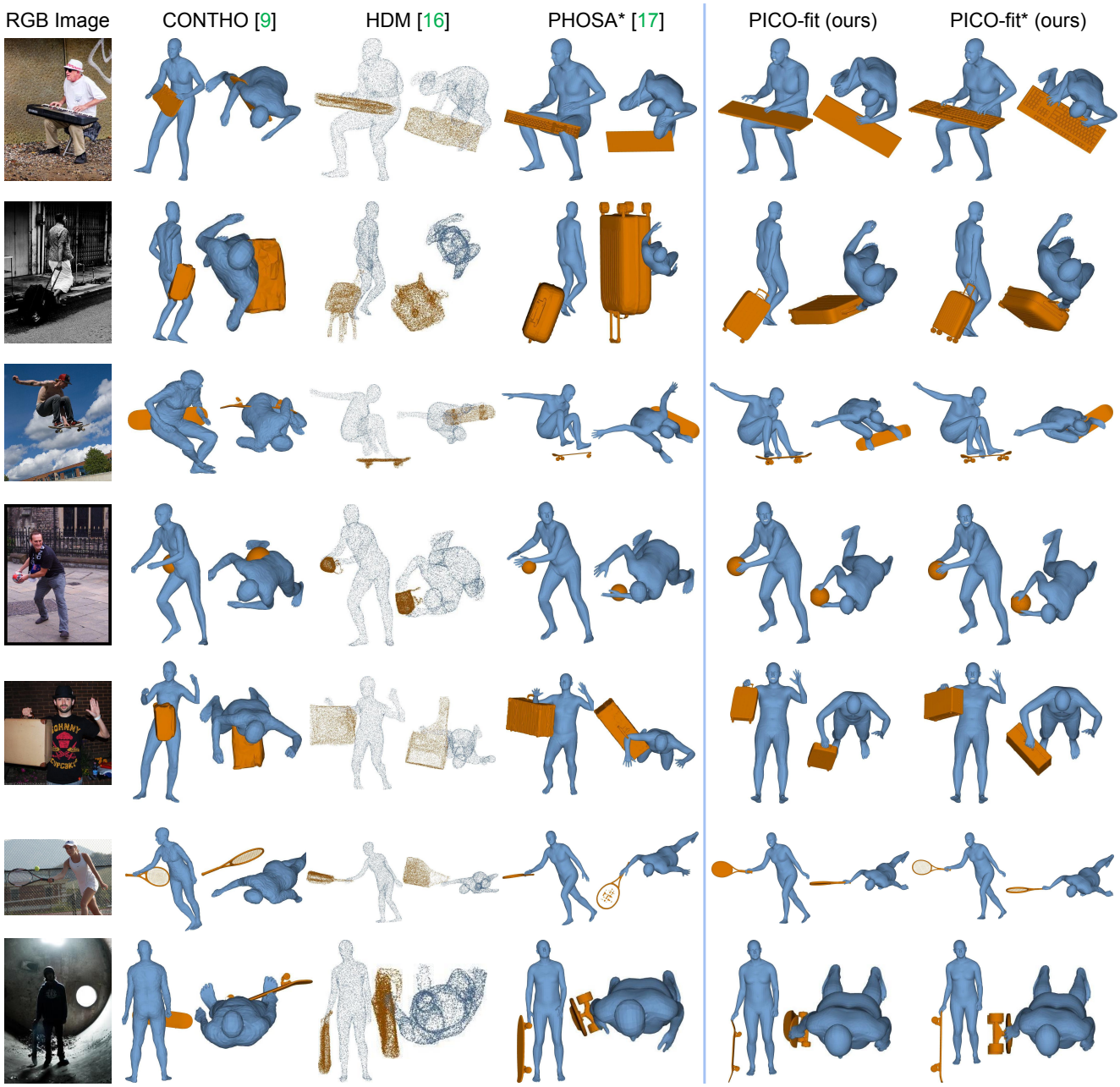

Figure S.11. Qualitative evaluation of CONTHO, HDM and PHOSA* alongside PICO-fit and PICO-fit* on object categories handled by all baselines. Since HDM cannot produce image overlays, we present only front- and top-down views for all methods

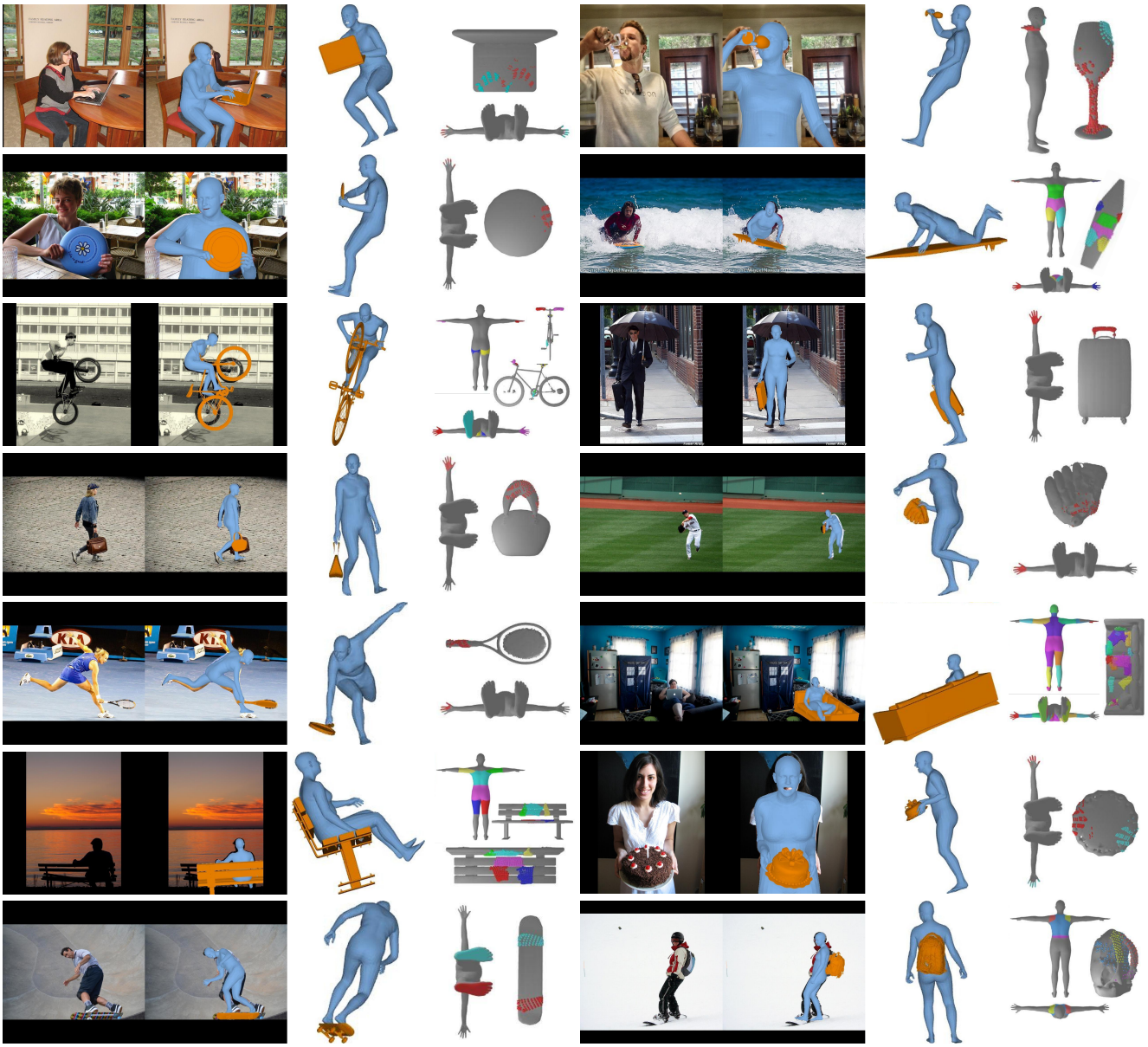

Figure S.12. 3D HOI reconstructions from our PICO-fit method on various object categories. For each triplet in each row we see (from left to right): an input RGB image, PICO-fit's estimated meshes overlaid on the image (camera view), a side view, and the contact annotations that were looked up and taken from PICO-db.

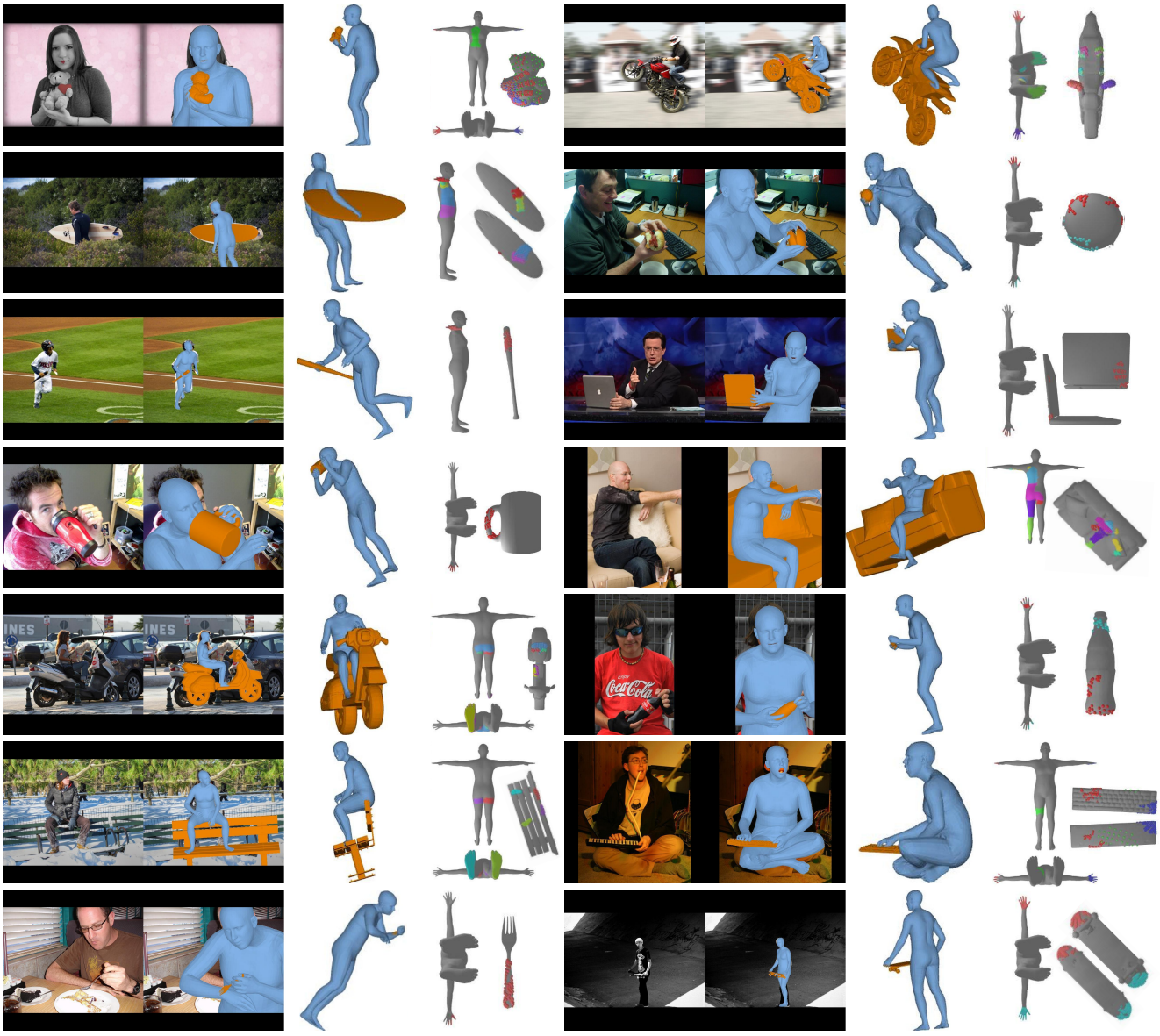

Figure S.13. 3D HOI reconstructions from our PICO-fit* method on various object categories. For each triplet in each row we see (from left to right): an input RGB image, PICO-fit*'s estimated meshes overlaid on the image (camera view), a side view, and the corresponding contact annotations from PICO-db.