CPF: Learning a Contact Potential Field to Model the Hand-Object Interaction
— *Supplementary Document* —

In the supplemental document, we provide:

A. Anatomically Constrained A-MANO

A.1. Derivation of *Twist-splay-bend* Frame.

In this section, we introduce the proposed *twist-splay-bend* frame of A-MANO. Both the original MANO [11] and our A-MANO hand model are driven by the relative rotation at each articulation. To mitigate the pose abnormality, we apply constraints on the rotation *axis-angle*[1]. We intend to decompose the rotation *axis* into three components to the three axes of a Euclidean coordinate frame, in which each component depicts the proportion of rotation along that axis. Obviously, there have infinity choices of the three orthogonal axes. MANO adopts 16 identical coordinate frames whose 3 orthogonal axes are not coaxial to the direction of the hand kinematic tree (Fig. 1 left). Different from MANO, we follow the Universal Robot Description Format (URDF) [8] that describe each articulation along the hand kinematic tree as a revolute joint[2]. Nevertheless, a revolute joint only has one degree of freedom, which is not enough to drive the motion of a real hand. Thus, we assign each articulation with three revolute joints, named as *twist*, *splay* and *bend* (Fig. 1 right),

Here, we elaborate the conversion from the MANO's all identical coordinate system of to our *twist-splay-bend* frame in three steps. For each articulation, we first compute the *twist* axis as the vector from the child of the current joint to itself. Then we employ MANO's y (up) axis and derive the *bend* axis that is calculated from cross product on the *twist* and y axes. Finally, we obtain the *splay* axis by applying cross product on the *bend* and *twist* axes. We illustrate the above procedures in Fig. 2.

[1] Rotation cay be represented as rotating along an *axis* by an *angle*.

[2] https://en.wikipedia.org/wiki/Revolute_joint

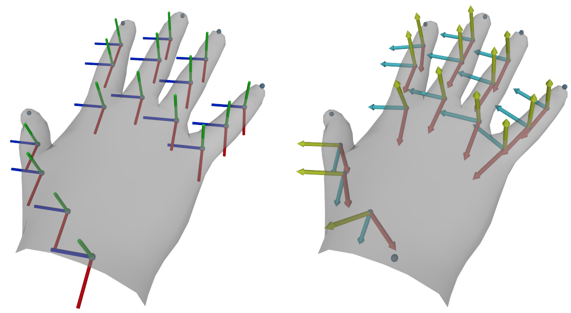

Figure 1. Visual comparison of MANO' s coordinate system to the proposed *twist-splay-bend* system.

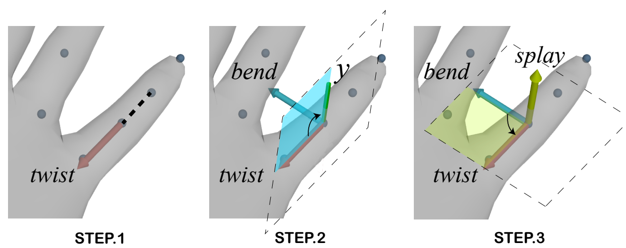

Figure 2. Illustration of converting MANO' s coordinates system to the proposed *twist-splay-bend* system.

A.2. Hand Subregion Assignment

As introduced in main text §3 (Anchors.), we divide the hand palm into 17 subregions, and interpolate the vertices in each subregion into representative *anchor* / *anchors*. In this part, we will firstly discuss how we assign hand vertices to several subregion.

According to hand anatomy, the linkage bones consists of carpal bones, metacarpal bones, and phalanges, where phalanges can be further divided into three kinds: proximal phalanges, intermediate phalanges, and distal phalanges. Here we assume the link between MANO joints are a counterpart of linkage bones on hand. We now assign the vertices of MANO into 17 subregions based on the linkage bones. The subregions' names and abbreviations are defined in Fig. 3. For clarity, we number the MANO links from 1 to 20 as illustrated in Fig. 4 (left).

To assign the MANO vertices to its corresponding re-

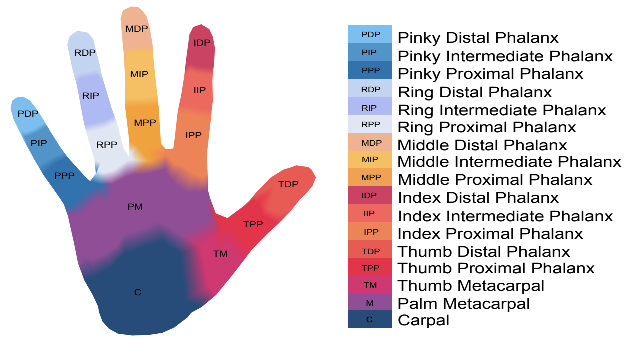

Figure 3. Hand subregions with names and abbreviations.

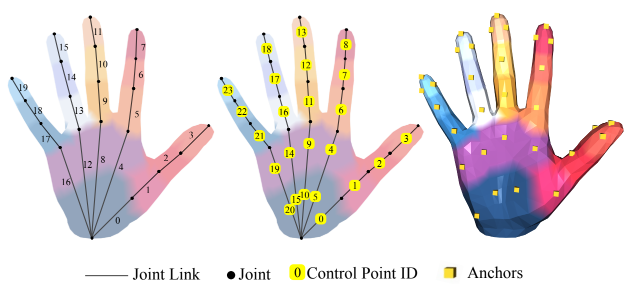

Figure 4. Left: joint links with ID; Middle: control points with ID; Right: anchors

gion, we need firstly assign the vertices to the link that lies inside the region. This is achieved by *control points*. For link 0-3, 5-7, 9-11, 13-15, 17-20, we set one control point at the midpoint of the link's ends, while for link 4, 8, 12, and 16, we set two control points at the upper and lower third of the link's ends. For clarity, we also number the control points from 0 to 23 as shown in Fig. 4 (middle). After a list of control points are obtained, we label each hand vertex to one of these control points by querying which control point it has the least distance from. Finally, we merge the vertices that belong to control points 0, 5, 10, 15, and 20 to derive subregion of **Palm Metacarpal**, and merge those vertices that belong to control points 4, 9, 14, 19 to derive subregion of **Carpal** .

A.3. Hand Anchor Selection

Here we elaborate on how we select the *anchors* based on the subregions and their control points. To ensure these anchors can be used in a common optimization framework and keep their representative power during the process of optimization, we propose the following three protocols: **a)** Anchors should be located on the surface of the hand mesh. **b)** Anchors should distribute uniformly on the surface of the region it represents. **c)** Anchors can be derived from hand vertices in a differentiable way.

Anchors are located on the surface of hand mesh (protocol **a**), so they must be located on some certain faces of the hand mesh. We can use the vertices of the face on which hand anchors reside to interpolate the anchors' position. Suppose the hand mesh has the form of $\mathbf{M} = (\mathbf{V}, \mathbf{F})$, where $\mathbf{V}$ is a set of all vertices and $\mathbf{F}$ is a set of all faces. Considering one face $\boldsymbol{f} \in \mathbf{F}$ of mesh whose vertices are stored in order: $\boldsymbol{f} = \{i_k\}, \boldsymbol{v}_k = \mathbf{V}[i_k], k \in \{1, 2, 3\}$. We can get two edges of that face: $\boldsymbol{e}_1 = \boldsymbol{v}_2 - \boldsymbol{v}_1, \boldsymbol{e}_2 = \boldsymbol{v}_3 - \boldsymbol{v}_1$. Then the local position of the anchor $\tilde{\boldsymbol{a}}$ inside the face can be represented by linear interpolation of $\boldsymbol{e}_1$ and $\boldsymbol{e}_2$: $\tilde{\boldsymbol{a}} = x_1\boldsymbol{e}_1 + x_2\boldsymbol{e}_2$, where the x_1, x_2 are some weights. Finally, the global position of the anchor $\boldsymbol{a}$ will be $\boldsymbol{a} = \boldsymbol{v}_1 + \tilde{\boldsymbol{a}} = \boldsymbol{v}_1 + x_1\boldsymbol{e}_1 + x_2\boldsymbol{e}_2 = (1 - x_1 - x_2)\boldsymbol{v}_1 + x_1\boldsymbol{v}_2 + x_2\boldsymbol{v}_3$. During the optimization process, we can use the precomputed face $\boldsymbol{f}$ and weights x_1, x_2, along with the predicted hand vertices $\mathbf{V}$ to calculate the position of all the anchors. As the anchor is a linear combination of hand vertices, any loss that is applied to the anchors' position can be back-propagated to the vertices on the MANO surface, making the anchor-bases hand mesh differentiable.

We utilize control points introduced in §A.2 to derive anchors. Since the anchor selection is independent of hand's configuration, we adopt a flat hand in the canonical coordinate system. As illustrated in Fig. 4 (middle, right), the control points are roughly uniformly distributed in each subregion. Each control point will correspond to an anchor of that subregion. The **Carpal** is an only exception: we select only 3 over 5 (ID: 5, 10, 20) of the control points in the subregion of **Carpal** for anchor derivation.

To derive an anchor from a control point, we need to get one face (consist of 3 integers) and two weights. **1)** ***Non-tip regions.*** For non-tip regions, we cast a ray that is originated from each control point in a certain subregion, and pointing to the palm surface. We retrieve the first intersection of the ray with hand mesh. This intersection will be the anchor that correspond to the control point, also the subregion. **2)** ***Tip regions.*** For tip regions, we would select three anchors of each control point to increase the density of anchors in that subregion, as tip involves more contact information during manipulation. For the control point in tip subregions, we first cast a ray originated from the control point and get the intersection point on the hand mesh. Then a cone is created with the control point as apex, the intersection point as the base center, and a base radius. The base radius is estimated by the maximum distance of vertices in the subregion to their control point. Three generatrices equally distributed on cone surface are selected as new ray casting directions. We cast three rays from the control point in the direction of the three generatrices and retrieve the intersection points with hand mesh. These intersection points will be selected as anchors to that control point in the fingertip regions.

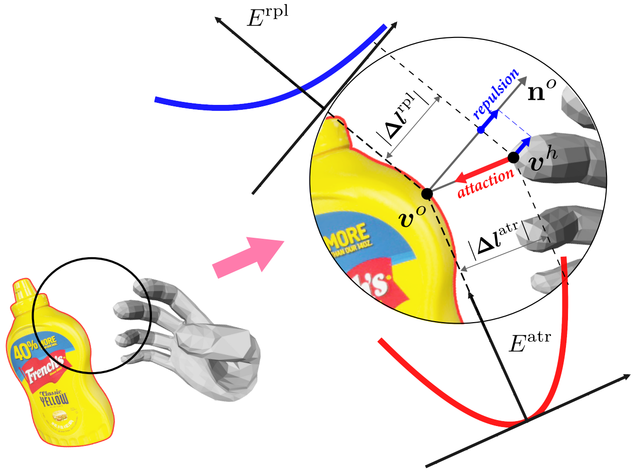

Figure 5. Illustration of the elastic energy w.r.t. a pair of hand-object vertices.

B. Spring's Elasticity

B.1. Elastic Energy Analysis

Here we illustrate elastic energy between a pair of points $\boldsymbol{v}_i^h$ and $\boldsymbol{v}_j^o$, denoting one vertex on hand surface and another vertex on object surface respectively. The vertex on object surface binds with a vector $\mathbf{n}_j^o$ representing the normal direction at this vertex (also the direction of repulsion). Then we compute the offset vector $\boldsymbol{\Delta l}_{ij}^{\text{atr}} = \boldsymbol{v}_i^h - \boldsymbol{v}_j^o$, and the projection of the offset vector on object normal $\mathbf{n}_j^o$: $|\boldsymbol{\Delta l}_{ij}^{\text{rpl}}| = (\boldsymbol{v}_i^h - \boldsymbol{v}_j^o) \cdot \mathbf{n}_j^o$. $|\boldsymbol{\Delta l}_{ij}^{\text{rpl}}|$ is positive if $\boldsymbol{v}_i^h$ falls outside the object, and negative if $\boldsymbol{v}_i^h$ falls inside the object. We use an exponential function here to provide magnitude and gradient heuristic for optimizer: **a)** the less $|\boldsymbol{\Delta l}_{ij}^{\text{rpl}}|$ is, the more $\boldsymbol{v}_i^h$ penetrates into the object. The gradient of repulsive energy will be an exponential increasing function of $\boldsymbol{\Delta l}_{ij}^{\text{rpl}}$. **b)** when $\boldsymbol{v}_i^h$ intersects into the object, both the repulsion and the attraction will push $\boldsymbol{v}_i^h$ towards the surface; when $\boldsymbol{v}_i^h$ is outside the object, the attraction and repulsion will point to opposite directions, leading to a balance point outside but in the vicinity to the object's surface. We provide an intuitive illustration in Fig. 5.

B.2. Anchor Elasticity Assignment

As discussed in main text §4 (Annotation of the Attractive Springs), we treat the elasticity of the *attractive* spring as the network prediction. Here, we shall provide the annotation heuristics of the *attractive* spring $\hat{k}^{\text{atr}}$ First, we set the anchor $\boldsymbol{a}_i$ - vertex $\boldsymbol{v}_j^o$ pair with ground-truth distance $|\boldsymbol{\Delta}\hat{\boldsymbol{l}}_{ij}^{\text{atr}}| > 20mm$ as invalid contact and has $\hat{k}_{ij}^{\text{atr}} = 0$. Second, for those anchor-vertex pairs within the distance threshold 20 mm, an inverse-proportional $\hat{k}_{ij}^{\text{atr}}$ is assigned according to the $|\boldsymbol{\Delta}\hat{\boldsymbol{l}}_{ij}^{\text{atr}}|$:

$$\hat{k}_{ij}^{\text{atr}} = 0.5 * \cos\left(\frac{\pi}{s} * |\boldsymbol{\Delta}\hat{\boldsymbol{l}}_{ij}^{\text{atr}}|\right) + 0.5 \quad (1)$$

where the scale factor $s = 20\, mm$.

To note, we do not have a strict requirement on the function of $\hat{k}_{ij}^{\text{atr}}$. Any other functions should also work when satisfying: **a)** $k = 1$ when $|\boldsymbol{\Delta l}| = 0$; **b)** k is inverse proportional to $|\boldsymbol{\Delta l}|$ in the range of 0 to 20 mm; **c)** k is bounded by 0 and 1. The choice of cosine function is simply due to its smoothness.

Figure 6. Unsuitable samples in HO3Dv2 testing set.

C. HO3D Dataset

C.1. Analysis and Selection

As we mentioned in the main text §6.1, several samples in the HO3D testing set do not suit for evaluating MIHO. Firstly, since GeO requires the predicted 6D pose of the known objects, all the grasps of the *pitcher* have to be removed. Secondly, many interactions of hand and objects in the testing set are not stable. For example, sliding the palm over the surface of a *bleach cleanser bottle*, may cause a strange contact and mislead the optimization in GeO. Therefore, we only select the grasps that can pick up the objects firmly. We show several unsuitable samples in Fig. 6. Table.1 shows our final selection on HO3Dv2 test set, as we called HO3Dv2$^-$.

Sequences	Frame ID
SM1	All
MPM10-14	30-450, 585-685
SB11	340-1355, 1415-1686
SB13	340-1355, 1415-1686

Table 1. **HO3Dv2$^-$ selection.** We select 6076 samples in the HO3Dv2 test set to evaluate MIHO.

C.2. Data Augmentation

We augment the training sample in HO3Dv1 in terms of poses and grasps. **a)** To generate more poses, we firstly randomize a disturbance transformation to the hand and object poses in the object canonical coordinate system. Then, we apply the disturbance on the hand and object meshes and render these meshes to image by a given camera intrinsic. **b)** To generate more grasps, we fit more stable grasps

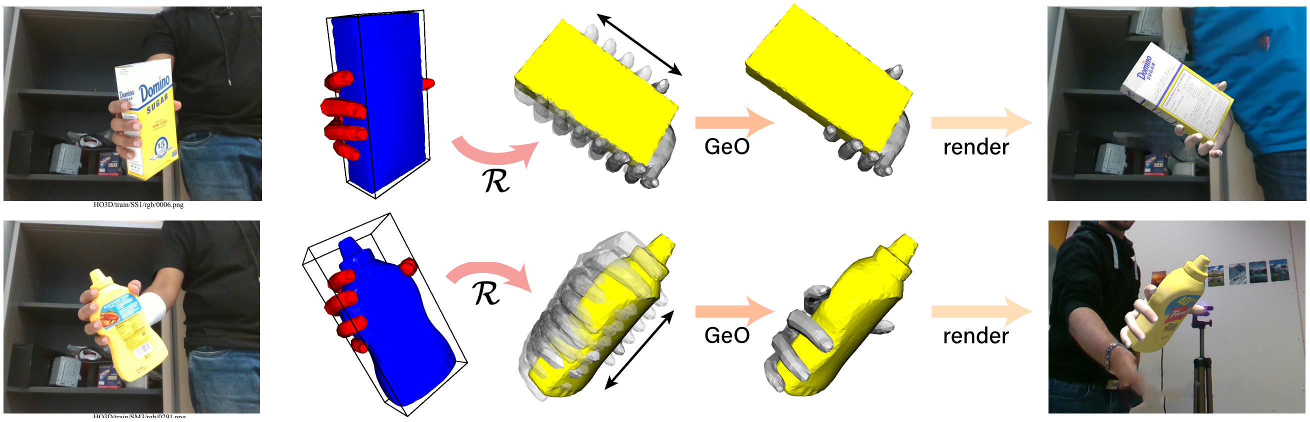

Figure 7. **HO3D [3] Dataset augmentation.** We demonstrate the process of generating synthetic training images. $\mathcal{R}$ stands for the random transformation.

around the object. Specifically, as we show in Fig. 7, the generation procedure is achieved by 2 steps: 1) Manually move the hand around the tightest bounding cuboid of the object. 2) Refine the hand pose in the proposed GeO. Since the *attractive* springs in CPF are unavailable here, we replace the attraction energy in main text Eq. 3 with the $\mathcal{L}_A$ in [6], Eq. 4, and retain the repulsion energy and the anatomical cost. The optimization process of grasping generation can be expressed as:

$$\hat{\mathcal{V}}^h \longleftarrow \underset{(\boldsymbol{P}_w, \boldsymbol{R}_j)}{argmin}(\mathcal{L}_A + E^{\text{rpl.}} + \mathcal{L}_{\text{anat}}) \tag{2}$$

D. Experiments and Results

D.1. Implementation Details

In this section we provide more implementation details about the HoNet, PiCR, and GeO module.

HoNet. The HoNet module employs ResNet-18 [7] backbone initialized with ImageNet [1] pretrained weights. For FHB and HO3Dv2 dataset, we use the pretrained weights released from [5]. For the HO3Dv1 dataset, we train the HoNet with Adam solver and a constant learning rate of 5×10^{-4} in total 200 epochs.

PiCR. The PiCR module employs a Stacked Hourglass Networks [9] (with 2 stacks) as backbone, a PointNet [10] as the point encoder, and three multi-layer perceptrons as heads. The image features yield from the two hourglass stacks are gathered together and sequentially fed into the PointNet encoder and three heads. While the loss is computed over the sum of two rounds prediction, both PointNet encoder and the three heads have only one instance throughout PiCR module. At the evaluation stage, we only use the image features from the last hourglass stack to get the prediction from three heads.

We train the PiCR module with two stages. **1) *Pretraining.*** We pretrain the PiCR module with the input image and the ground-truth object mesh in camera space. The ground-truth object mesh are disturbed by a minor rotation and translation shift. We employ Adam solver with an initial learning rate of 1×10^{-3}, decaying 50% every 100 epochs. The total epochs during pretraining stage is 200. **2) *Fine-tuning.*** At the fine-tuning stage, we feed PiCR module with the object vertices predicted from HoNet. The HoNet's weights is freezed during PiCR fine-tuning. We employ Adam solver and set the initial learning rate in fine–tuning stage as 5×10^{-4}, decayed to 50% every 100 epochs, and finished at 200 epochs. In both stages, we set the training mini-batch size to 8 per GPU, and a total of 4 GPUs are used.

GeO. The GeO is a fitting module based on the non-linear optimization. For each sample, we minimize the cost function in 400 iterations, with a initial learning rate of 1×10^{-2}, reduced on plateau that the cost function has stopped decaying in 20 consecutive iterations. We implement GeO in PyTorch thanks for its auto derivative, and an Adam solver is employed when updating the arguments. To note, GeO can also support any other optimization toolbox.

D.2. Ablation Study

As referred in main text §6.4 (Ablation Study), this section contains another three ablation studies. all the following experiments are under the ***hand-object*** setting.

The Impact of the k^{rpl}. While the elasticity k^{atr} of the *attractive* springs are predicted in PiCR, the elasticity k^{rpl} of those *repulsive* strings are empirically set to 1×10^{-3}. In order to measure the impact of the magnitude of k^{rpl} on repulsion, we test our GeO with seven experiment settings in which the k^{rpl} is set to $\{0.2,\ 0.6,\ 1.0,\ 1.4,\ 2.0,\ 4.0,\ 8.0\} \times$

10^{-3}, respectively. The experiment with $k^{\mathrm{rpl}} = 1 \times 10^{-3}$ is in accord with the default experiment in main text. As shown in Tab. 2, while the large k^{rpl} can reduce the solid interpenetration volume, it may also push the attraction apart thus is not preferable in the reconstruction metrics: hand MPVPE and object MPVPE.

k^{rpl}	Scores				
	HE ↓	OE ↓	PD ↓	SIV ↓	DD ↓
2.0×10^{-4}	19.49	21.57	17.77	13.22	20.85
6.0×10^{-4}	19.51	21.57	17.22	12.40	21.63
$\mathbf{1.0 \times 10^{-3}}$	**19.54**	**21.57**	**16.92**	**11.76**	**22.41**
1.4×10^{-3}	19.59	21.58	16.75	11.00	23.24
2.0×10^{-3}	19.69	21.59	16.41	10.09	24.55
4.0×10^{-3}	20.03	21.63	15.09	7.65	29.33
8.0×10^{-3}	20.95	21.92	12.86	4.27	40.79

Table 2. **Ablation results:** the impact of the magnitude of k^{atr}. HE stands for hand mean per vertex position error (mm); OE stands for object mean per vertex position error (mm); PD stands for penetration depth (mm); SIV stands for solid intersection volume (cm^3); D stands for disjointedness distance (mm).

A-MANO with PCA Pose. Since the MANO can also be driven by the PCA components of joint rotation, we further conduct experiments to demonstrate the superiority of our full MANO (MANO with 15 relative joint rotations) over the PCA MANO (MANO with 15 PCA components of rotations). Tab. 3 shows that our full MANO can achieve a notable decrease in the hand MPVPE. We attribute this to the fact that the PCA MANO tends to recovery a hand that is inclined to the mean flat pose, while our full version imposes higher flexibility on the hand pose.

However, fitting on the 15 rotations in forms of $\mathfrak{so}(3)$ brings $15 \times 3 = 45$ degree of freedoms, which is less stable against pose abnormality. Hence in order to fully exploit the advantages when fitting on the rotations of 15 joints, we have to combine the anatomical constrains with it.

Settings	Scores				
	HE ↓	OE ↓	PD ↓	SIV ↓	DD ↓
Full MANO	**19.54**	**21.57**	**16.92**	**11.76**	**22.41**
PCA MANO	23.32	24.41	22.47	11.90	26.72

Table 3. **Ablation results:** the MANO with PCA pose.

Unwanted Twist Correction. In this part, we show the effectiveness when fitting the 15 rotations with anatomical constrains. We observe an unwanted twist of thumb in the ground-truth pose of HO3Dv1 testing set. As shown in Fig. 8, since A-MANO imposes constraints on the *twist* component of the rotation axis, it can achieve a more visually pleasing result in such case.

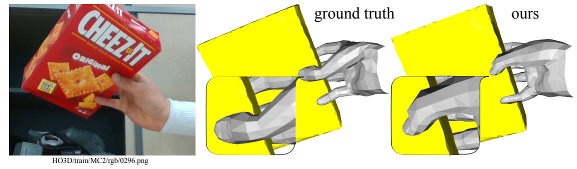

Figure 8. Example to show that our A-MANO can mitigate the unwanted twist (see thumb) exhibited in ground-truth.

E. More Qualitative Results

We demonstrate the qualitative results of MIHO in Fig. 9 on both the FHB [2] and HO3D dataset [4]. Note that the ground truth of the test set in HO3Dv2$^-$ [4] is not available.

References

[1] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 4

[2] Guillermo Garcia-Hernando, Shanxin Yuan, Seungryul Baek, and Tae-Kyun Kim. First-person hand action benchmark with rgb-d videos and 3d hand pose annotations. In *CVPR*, 2018. 5, 6

[3] Shreyas Hampali, Markus Oberweger, Mahdi Rad, and Vincent Lepetit. Ho-3d: A multi-user, multi-object dataset for joint 3d hand-object pose estimation. *arXiv preprint arXiv:1907.01481*, 2019. 4, 6

[4] Shreyas Hampali, Mahdi Rad, Markus Oberweger, and Vincent Lepetit. Honnotate: A method for 3d annotation of hand and object poses. In *CVPR*, 2020. 5, 6

[5] Yana Hasson, Bugra Tekin, Federica Bogo, Ivan Laptev, Marc Pollefeys, and Cordelia Schmid. Leveraging photometric consistency over time for sparsely supervised hand-object reconstruction. In *CVPR*, 2020. 4

[6] Yana Hasson, Gul Varol, Dimitrios Tzionas, Igor Kalevatykh, Michael J Black, Ivan Laptev, and Cordelia Schmid. Learning joint reconstruction of hands and manipulated objects. In *CVPR*, 2019. 4

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 4

[8] Kevin M Lynch and Frank C Park. *Modern Robotics*. Cambridge University Press, 2017. 1

[9] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *ECCV*, 2016. 4

[10] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *CVPR*, 2017. 4

[11] Javier Romero, Dimitrios Tzionas, and Michael J Black. Embodied hands: Modeling and capturing hands and bodies together. *TOG*, 2017. 1

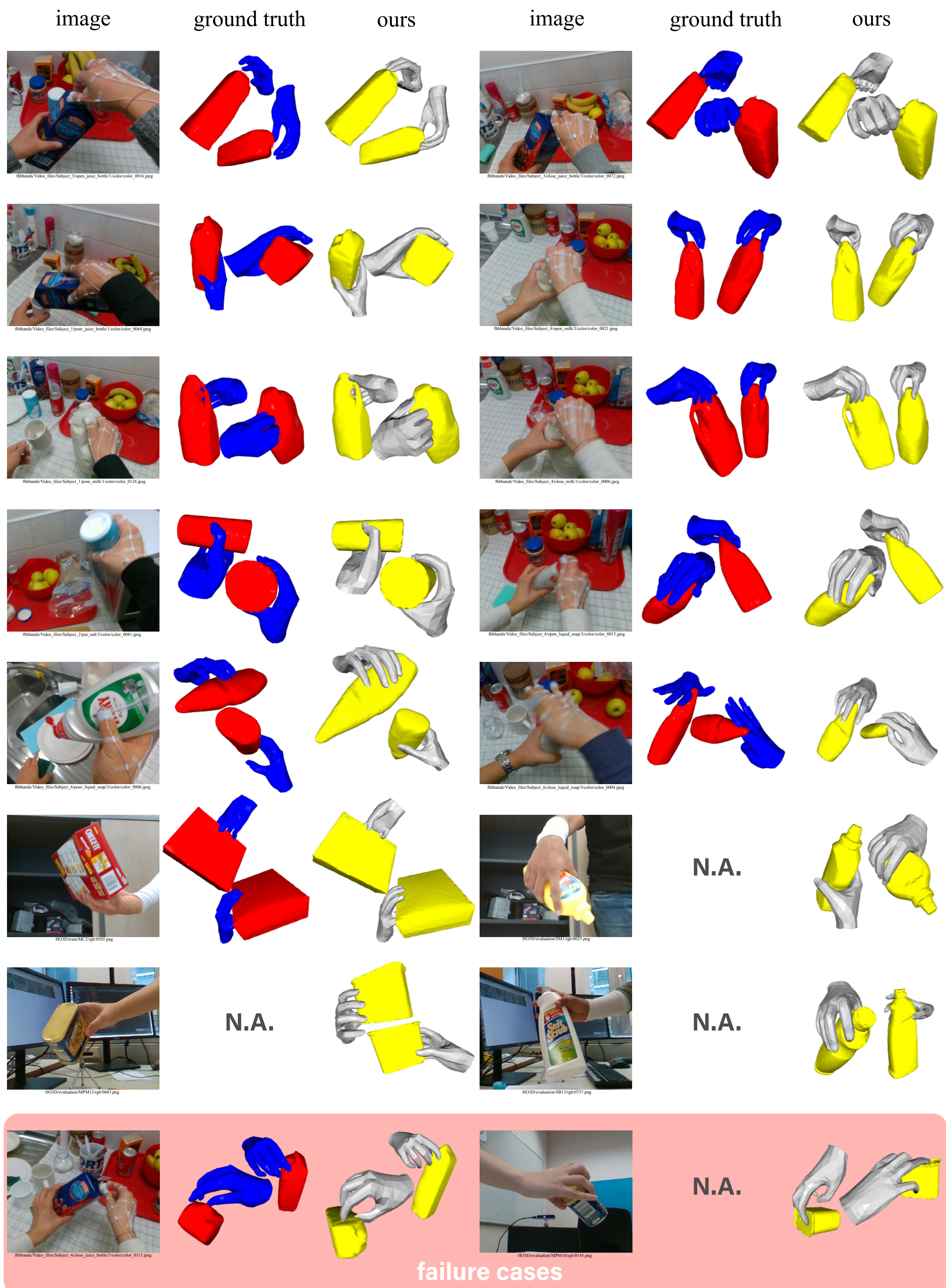

Figure 9. **Qualitative results on FHB [2], HO3Dv1[3] and HO3Dv2^{-} [4] datasets.** The last row shows the failure cases.