

CPF: Learning a Contact Potential Field to Model the Hand-Object Interaction

Lixin Yang, Xinyu Zhan, Kailin Li, Wenqiang Xu, Jiefeng Li, Cewu Lu

Shanghai Jiao Tong University, Shanghai, China

{siriusyang, kelvin34501, kailinli, vinjohn, ljf_likit, lucewu}@sjtu.edu.cn

Abstract

Modeling the hand-object (HO) interaction not only requires estimation of the HO pose, but also pays attention to the contact due to their interaction. Significant progress has been made in estimating hand and object separately with deep learning methods, simultaneous HO pose estimation and contact modeling has not yet been fully explored. In this paper, we present an explicit contact representation namely Contact Potential Field (CPF), and a learning-fitting hybrid framework namely MIHO to Modeling the Interaction of Hand and Object. In CPF, we treat each contacting HO vertex pair as a spring-mass system. Hence the whole system forms a potential field with minimal elastic energy at the grasp position. Extensive experiments on the two commonly used benchmarks have demonstrated that our method can achieve state-of-the-art in several reconstruction metrics, and allow us to produce more physically plausible HO pose even when the ground-truth exhibits severe interpenetration or disjointedness. Our code is available at <https://github.com/lixiny/CPF>.

1. Introduction

It is essential to model hand-object interaction from a single image for understanding the human activities, in which simulating a physically plausible grasp is also crucial for VR/AR, teleoperation, and grasping applications. Given an image as input, the problem aims to not only estimate proper hand-object pose but also to recover a natural grasp configuration. While estimating hand [39, 33, 61, 4, 20, 59] or object [21, 24, 15, 57, 58] alone has made a considerable success over the past decades, simultaneously estimating hand-object pose [25, 54, 24, 30, 13] with interaction has only emerged in the past few years.

Previous works on joint hand-object estimation usually treat the contact as a result of the correct pose estimation [24, 31, 47]. Apparently, if the hand and object can be perfectly recovered, the contact between them will also be satisfied. Yet, such perfection cannot be achieved in practice. Since contact can provide rich cues to guide accurate pose and natural grasp, more attention has been recently

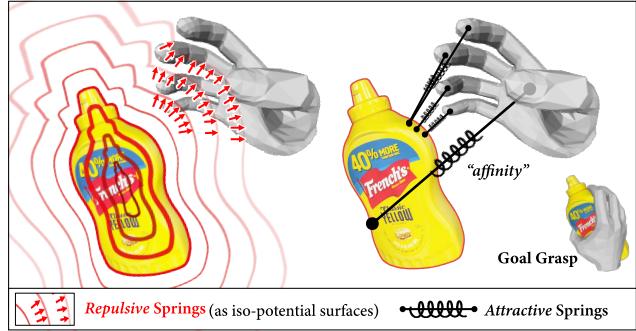


Figure 1. **Illustration of the proposed Contact Potential Field.** The contacts between hand and object vertices are modeled as the attractive (right) and repulsive (left) springs that connect paired vertex on them.

drawn to the contact modeling [5, 7] and contact representation [28, 6]. And several contact datasets [5, 7, 53] have been released to the community. However, a solution of properly integrating contact modeling into the current hand-object pose estimation pipeline has remained an open research question. The existing methods either exploit distance-based attraction and repulsion [25, 28] to mitigate disjointedness and interpenetration, or refine the predicted pose in virtue of physics simulators [30, 31, 18]. While the both solutions are considered to be irrelevant to contact *semantics*, which we will explain later, the latter solutions also lack flexibility on hand pose and shape.

To model the contact, we propose an explicit representation named **Contact Potential Field (CPF, §4)**. It is built upon the idea that the contact between a hand and an object mesh under grasp configuration is multi-point contact, which involves multiple hand-object vertex pair affinities. These affinities are regarded as the contact *semantics*, which depict the pairing of the hand-object vertices that come into contact with each other during the interaction. When noisy predicted hand and object are disjointed from each other, we shall apply an attraction to pull these vertex pairs close; While the hand and object are intersected, we shall have a repulsion to push them away. Contacts of those affinitive vertex pairs are the result of equilibrium between the attraction and repulsion. In this paper, we treat each contact

ing HO vertex pair as a spring-mass system. First, the two end-points of spring is a counterpart of the two HO vertices in affinity. Second, the spring’s elastic property is another counterpart of the intensity of the vertex pair affinity. In this way, we can model the HO interaction with a potential field, as we call it CPF, which is determined by minimal elastic energy at the grasp position. Therefore, estimating the HO pose under contact is equivalent to minimizing the elastic energy inside CPF. Representing contact as CPF has two main advantages. First, compared with contact heuristic with proximity metrics [1, 55] or distance field [28, 6], CPF is able to assign per-vertex contact *semantics* (contact points on different hand part) to object mesh. Second, by minimizing the elastic energy, CPF can uniformly avoid interpenetration and control the disjointedness. Based on CPF, we also propose a novel learning-fitting hybrid framework namely for **Modeling the Interaction of Hand and Object**, as we call it **MIHO** (§5).

Another problem with the existing methods is the representation of the hand model. Most researches adopted a skinning model, MANO [50], to represent hand. MANO is considered to be flexible and deformable with its pose and shape parameters. However, fitting on these high DoFs parameters is prone to anatomical abnormality. Researches in the robotics community adopted a dexterous hand [31, 18] in the off-the-shelf grasping software [38], which can almost guarantee a valid pose. But the rigidity of those rod-like hand is less suitable for applications in CV/CG. To make the best of both worlds, we propose a novel anatomically constrained hand model namely A-MANO (§3). It inherits the formulation of the skinning model and constrains the hand joints’ rotation within a proposed *twist-splay-bend* frame (Fig. 2).

For evaluation, we report our scores on FHB [19] and HO3D [23, 22] dataset in terms of reconstruction and physical quality metrics. Note that, the ground truth of FHB is noisy and suffers from severe interpenetration [28]. Since our method can avoid the penetration in the first place, our results are more visually and physically plausible. Therefore, we argue that, in this dataset, a higher reconstruction score does not necessarily benchmark the performance of the method. While on HO3D, we achieve state-of-the-art performance on both reconstruction and physical metrics. The contributions of this paper are as follows.

- We highlight contact in the hand-object interaction modeling task by proposing an explicit representation named CPF.
- We introduce A-MANO, a novel anatomical-constrained hand model that helps to mitigate pose’s abnormality during optimization.
- We present a novel framework, MIHO, for modeling hand-object interaction. It can achieve state-of-the-art performance on several benchmarks.

2. Related Work

3D Hand Reconstruction. Most of the existing 3D hand reconstruction methods [4, 61, 2] adopted a parametric skinning hand, *e.g.* MANO [50] as a template. To drive MANO, it is crucial to obtain joint rotation along hand kinematic tree. Boukhayma *et al.* [4] firstly proposed to regress the PCA components of the rotations. Later, directly regressing the full rotations from 3D positions [61, 59] has shown better performance. However, those high DoF regression is prone to pose abnormality. Thus, Spurr *et al.* [52] exploited biomechanical constraints over hand joints in training scheme. Different from [52], we apply rotation constraints over the axes and angles in the proposed *twist-splay-bend* coordinate frame.

Hand-object Pose Estimation. In a wide range of topics in modeling hand-object interaction, the most commonly referred one is HO pose estimation [25, 24, 13, 17, 54]. In this regard, the earlier methods focused on either hand [46, 48, 55] or object [56] pose alone, or estimated hand in grasping pose with knowing object shape prior [16, 8, 9, 10]. Jointly estimating hand and object pose was firstly presented by Romero *et al.* [49] via searching for nearest neighbors in a large database. Recently, learning-based frameworks have emerged in this area. Hasson *et al.* [25, 24] proposed two learning frameworks to recover hand-object meshes, one by synthesizing HO data under manipulation [25] and the other by exploiting photometric consistency over video sequence [24]. Doosti *et al.* [13] employed the graph neural networks [17] to lift the 2D HO keypoints into 3D space. Tekin *et al.* [54] adopted 3D YOLO [44] to predict HO pose in one stage. Korrawe *et al.* [28] recovered HO model in a form of Signed Distance Function [42].

Contact Heuristic. Exploiting contact heuristic in hand-object interaction can be traced back to several decades before [45, 14, 36]. Early works utilized some shape-specified contact physics (*e.g.* cones and blocks [45]) or predefined grasp [36] as prior. Studies on capturing [32] or imitating [3] HO interaction also leveraged contact to satisfy the reality. Later, the studies on grasping synthesis [60, 18, 31] and tracking [41, 34] turned to physical simulators to circumvent model’s intersection. Multi-point contact formulation was proposed in [29, 27, 1], which we found useful when applying physical constraints, *e.g.* [29, 27] used contact points to resolve penetration. For unified attraction and repulsion, most works employed heuristic such as proximity metric [25, 1, 55], signed distance function [28, 6], predefined contact pattern [47, 6], or turned to simulator [30, 31] for simplicity. Recently, Antotsiou *et al.* [1] refined the grasp by attracting fingers to its nearest point on object surface w.r.t distance-based energy. Hasson *et al.* [25] applied well-designed interaction losses which are also based on proximity metric. Although our method differs from all

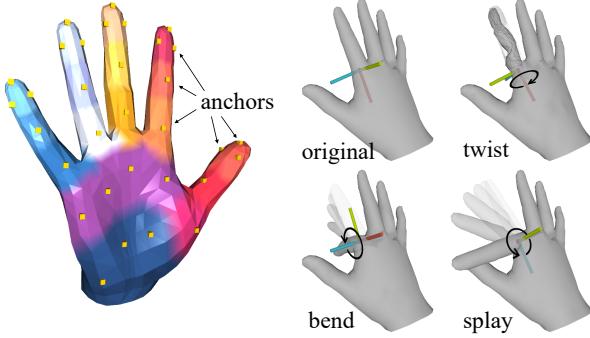


Figure 2. **Illustration of the proposed A-MANO.** Left: the subdivision of hand regions and anchors attached to it. Right: the proposed *twist-splay-bend* frame.

of the previous methods in terms of contact heuristic, we consider that both [1] and [25] are still strong baselines. Thus we will compare our contact heuristic with theirs.

3. Anatomically Constrained A-MANO

The proposed A-MANO inherits from a parametric skinning hand model, MANO [50], which drives an articulated hand mesh with pose parameters θ and shape parameters β . $\theta \in \mathbb{R}^{15 \times 3}$ is 15 joint rotations along the hand kinematic tree. And $\beta \in \mathbb{R}^{10}$ represents the PCA components of hand shape. The main differences of A-MANO from MANO are: 1) the restriction on the joints’ rotation axes and angles within the *twist-splay-bend* frame; 2) the *anchor* representation in the subdivision of hand region.

The Twist-splay-bend Frame. Fitting on 15 joint rotations of MANO requires high DoFs regression which may cause abnormal hand posture as shown in Fig. 7. Since the human hand can be modeled in a kinematic tree, and the majority of the joints only have one DoF about the *bend* axis, we can impose constraints over the rotation about the unwanted axes. Therefore the proposed *twist-splay-bend* Cartesian coordinate frame can be assigned to each joint along the kinematic tree. The frame’s *x*, *y*, *z* axes are coaxial to the 3 revolute directions: *twist*, *splay*, and *bend* direction on the basis of hand anatomy (Fig. 2 right). Then we can impose axial constraints in the *twist* and *splay* axes, and impose angular constraints w.r.t the *bend* angle. Details of the *twist-splay-bend* frame are elaborated in Supp A.1.

Anchors. Since the hand mesh of different subjects are almost identical in the subdivision of hand region (*e.g.* phalanges), we can interpolate several representative points (later we call it *anchors*) on hand mesh to largely reduce the number of HO vertex pairs. Instead of attaching springs from object mesh to all the affinitive vertices on hand mesh, we only attach them on the several hand subregion centers, as we call it *anchors* (Fig. 2 left). According to the statis-

tics [25, 7] on the contact frequency of different hand parts, we first divide the full hand palm into 17 subregions: 3 for each phalange of 5 fingers, 1 for metacarpals, and another for carpal. Then, we interpolate up to 4 anchors for each subregion. We ignore all the vertices on the back side of the hand. Details of subregion division and anchors interpolation are described in Supp A.2, A.3.

4. Contact Potential Field

Contact as Spring-Mass System. A single contact is modeled as a spring-mass system which consists of a spring and two mass points on each side (hand and object). When the spring is at its rest position, it does not store energy, whilst it is stretched or compressed, according to Hooke’s Law¹, it will store the elastic potential energy with the form: $\frac{1}{2}k|\Delta l|^2$, where k is the spring elasticity, and $|\Delta l|$ is a certain “distance” metric w.r.t. the spring’s rest position.

In CPF, we define two types of spring: *attractive* spring and *repulsive* spring. The goal of *attractive* spring is to pull the hand vertex v^h toward the object vertex v^o based on a given HO vertex pair affinity. And the goal of *repulsive* spring is to push the v^h away from v^o along the v^o ’s normal if the v^h is in the vicinity of v^o . Apart from these definitions, we should also point out that the *attractive* spring is bound with a certain pair of HO vertex affinity, while the *repulsive* spring only takes effect in the neighborhood of HO vertex pairs at some point.

- Attractive Spring. We define the rest length of *attractive* spring as 0 in which the hand vertex and object vertex are in perfect contact, and the *distance* metric $|\Delta l|$ as Euclidean distance. Given a HO affinity that includes a vertex pair: v_i^h and v_j^o , the $|\Delta l_{ij}^{\text{atr}}|$ is equal to $\|v_i^h - v_j^o\|_2$. The potential energy of the current *attractive* spring is given by:

$$E_{ij}^{\text{atr}} = \frac{1}{2}k_{ij}^{\text{atr}} * \|\Delta l_{ij}^{\text{atr}}\|_2^2 \quad (1)$$

- Repulsive Spring. We hope that the repulsion energy is high when v_i^h is penetrating or in the vicinity of v_j^o , but gradually decays as the v_i^h moves away from the object, and finally becomes negligible at certain distance. Given a proximate HO vertex pair: v_i^h and v_j^o , We define a repulsive spring to model this behavior. Supposing that the *repulsive* spring has the rest position at $+\infty$ away along the object normal n_j^o . We adopt a heuristic *distance* metric $|\Delta l| = e^{-|\Delta l_{ij}^{\text{rpl}}|} - e^{-\infty} = e^{-|\Delta l_{ij}^{\text{rpl}}|}$, where $|\Delta l_{ij}^{\text{rpl}}| = (v_i^h - v_j^o) \cdot n_j^o$ is the projection of the $(v_i^h - v_j^o)$ on the object normal n_j^o . Thus, the potential energy of the current *repulsive* spring is

$$E_{ij}^{\text{rpl}} = \frac{1}{2}k_{ij}^{\text{rpl}} * (e^{-|\Delta l_{ij}^{\text{rpl}}|})^2 \quad (2)$$

In literature, adopting repulsive effect along surface normal

¹https://en.wikipedia.org/wiki/Hooke's_law

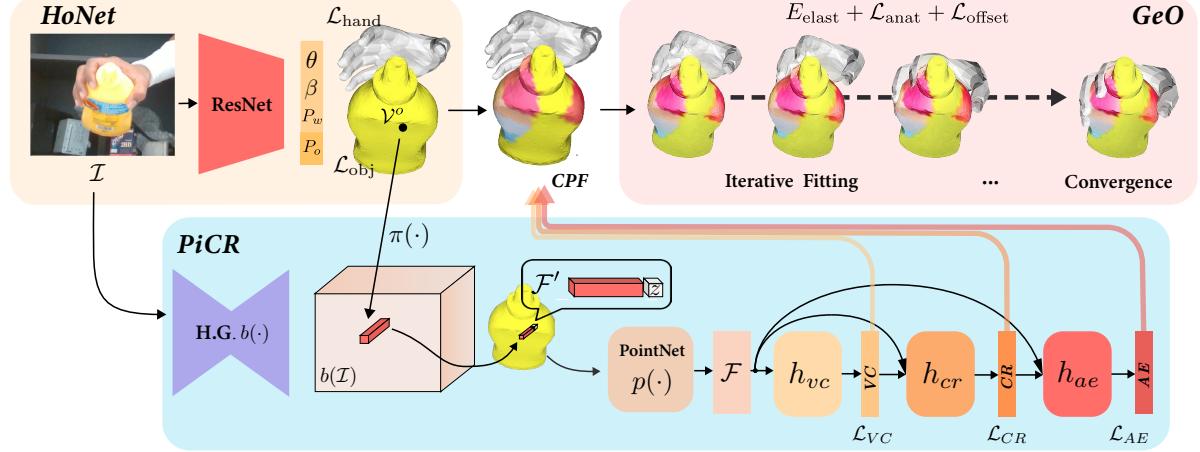


Figure 3. **The architecture of the hybrid model MIHO.** The MIHO consists of three submodules: the first HoNet estimates coarse poses of HO meshes, the second PiCR learns to recover the CPF and the last GeO retrieves the refined poses based on the CPF.

can be found in [6, 23]. [23] (Eq. 10) also discussed that $e^{-(\cdot)}$ is an efficient heuristic concerning sub-sampled set of vertices.

Grasping inside Contact Potential Field. By collecting all the *attractive* and *repulsive* springs, to form a natural grasp is equivalent to minimize the elastic energy:

$$E_{\text{elast}} = \sum_i \sum_j (E_{ij}^{\text{atr}} + E_{ij}^{\text{rpl}}) \quad (3)$$

As discussed in §3, the hand vertices can be simplified to subregion *anchors*, which will largely relax the difficulty of learning and fitting inside the CPF. Thus, for *attractive* spring, we replace the Δl_{ij} in Eq.1 to $\Delta l'_{ij} = a_i - v_j^h$, where a_i is the closest anchor to v_j^h . Besides, we would like to have the repulsion force be only applied to those HO affinity pairs that are of vertices in vicinity. Thus we set zero the repulsion energy when the vertex distance $\|v_j^h - v_i^h\|_2$ is greater than a threshold $t_{\text{rpl}} = 20 \text{ mm}$.

Annotation of the Attractive Springs (k_{ij}^{atr}). While the attraction energy is bound with certain HO affinities, the repulsion energy is rather ambient and affinity-agnostic. To integrate the CPF into learning framework, we only consider the k_{ij}^{atr} as the prediction of neural network. To enable this, network shall have the abilities of 1) pairing the hand anchors and object vertices into HO affinity pair, e.g. (a_i, v_j^h) ; and 2) regressing the intensity of those affinity pairs, e.g. k_{ij}^{atr} . These require annotation of the *attractive* springs k_{ij}^{atr} .

Given the ground-truth (gt.) HO pose and their mesh model, we automatically annotate each k_{ij}^{atr} based on a heuristic of the (a_i, v_j^h) pair distance. Since each a_i may be included in several affinity pairs, we hope the attraction energy stored in each spring at gt. HO pose is well balanced. Thus we assign the gt. $\hat{k}_{ij}^{\text{atr}}$ a value that is inverse-

proportional to the gt. $|\Delta l_{ij}^{\text{atr}}|$. In order to train the network, we also bound the magnitude of $\hat{k}_{ij}^{\text{atr}}$ by 0 and 1. Here we only provide a glimpse of the annotation heuristic of $\hat{k}_{ij}^{\text{atr}}$:

$$\hat{k}_{ij}^{\text{atr}} = 0.5 * \cos\left(\frac{\pi}{s} * |\Delta l_{ij}^{\text{atr}}|\right) + 0.5 \quad (4)$$

Empirically, we set the scale factor $s = 20 \text{ mm}$ and reject those HO affinities with gt. $|\Delta l_{ij}^{\text{atr}}| \geq 20 \text{ mm}$. As for the elasticity of *repulsive* spring, we empirically set all k_{ij}^{rpl} to 1×10^{-3} . Detailed analysis of the gt. $\hat{k}_{ij}^{\text{atr}}$ and the attraction-repulsion equilibrium are provided in Supp B.1, B.2,

5. Hybrid Framework – MIHO

With respect to the proposed CPF (§4), our approach MIHO models the hand-object interaction in three stages, namely HoNet (§5.1), PiCR (§5.2), and GeO (§5.3).

As shown in Fig.3, firstly, given an RGB image \mathcal{I} , HoNet predicts a coarse pose of hand mesh $V^h = \{v_i^h \in \mathbb{R}^3 | i \leq N_h\}$ and object mesh $V^o = \{v_j^o \in \mathbb{R}^3 | j \leq N_o\}$, where N_h and N_o are the number of the vertex of hand and object respectively. Then, PiCR learns to construct the CPF and collect the elastic energy E_{elast} in it. Finally, GeO minimizes E_{elast} in CPF to yield the refined HO meshes $*V^h, *V^o$.

5.1. Hand-object Pose Estimation Network, HoNet

The HoNet first predicts coarse poses of HO meshes by the baseline model *MeshRegNet* as in [24]. The outcomes from the baseline comprise in total 37 coefficients: object 6D pose $P_o \in \mathfrak{se}(3)$ (\mathbb{R}^6), hand wrist 6D pose $P_w \in \mathfrak{se}(3)$, PCA components of MANO pose $\theta_{\text{pca}} \in \mathbb{R}^{15}$ and shape $\beta \in \mathbb{R}^{10}$. With these coefficients, HoNet could place the HO meshes into camera frame. Details of the baseline can be referred to [24].

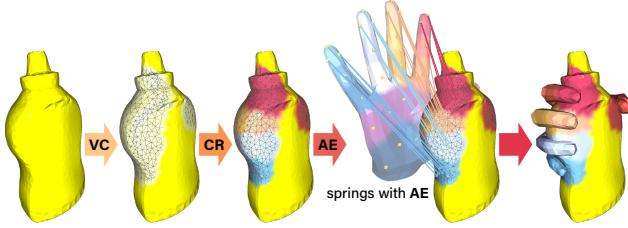


Figure 4. Illustration of assigning *Vertex Contact*, *Contact Region* and *Anchor Elasticity* onto object surface.

5.2. Pixel-wise Contact Recovery Module, PiCR

With the coarse meshes of hand and object in HoNet, PiCR learns to recover the CPF by firstly paring the hand anchors and object vertices into HO affinity pairs and then regressing the spring elasticities that describe the affinities. To achieve this, PiCR yields three cascaded outcomes: 1) *Vertex Contact* (*VC*) decides which vertices on object are in contact with hand; 2) *Contact Region* (*CR*) decides the subregion that is most likely to contact with those vertices in *VC*; 3) *Anchor Elasticity* (*AE*) represents the elasticities of the *attractive* springs. With *VC*, *CR*, and *AE*, we can then recover the CPF as illustrated in Fig. 4.

Vertex Contact. PiCR’s first outcome $VC \in \mathbb{R}^{N_o}$ stands for the contact probability of object vertices. More specifically, $VC[j]$ is a probability that implies the j -th object vertex v_j^o is in contact with hand. The loss function of *VC* is defined as a binary focal loss [35]:

$$\mathcal{L}_{VC} = - \sum_j^{N_o} \mathbb{1}_j^{img} * \alpha_j (1 - f_j)^\gamma \log(f_j) \quad (5)$$

where $f_j = p_j$ if the *gt.* \hat{v}_j^o belongs to any HO affinity, otherwise $f_j = (1 - p_j)$, and the p_j is the predicted probability at $VC[j]$. $\mathbb{1}_j^{img}$ denotes whether the vertex v_j^o is projected inside the image. α_j is inverse class frequency and γ is empirically set to 2.

Contact Region. PiCR’s second outcome $CR \in \mathbb{R}^{N_o \times 17}$ stands for the subregion probabilities of object vertices. More specifically, for the j -th query, $CR[j]$ contains 17 probabilities that indicates v_j^o ’s affinity toward 17 hand subregions. The loss function \mathcal{L}_{CR} is defined as a multi-class focal loss.

$$\mathcal{L}_{CR} = - \sum_j^{N_o} \mathbb{1}_j^{VC} * \mathbb{1}_j^{img} * (1 - m_j)^\gamma \log(m_j) \quad (6)$$

where the $m_j = \sum(p_j * t_j)$ in which $p_j = CR[j] \in \mathbb{R}^{17}$ is the predicted per-subregion probabilities through *softmax*, and $t_j \in \mathbb{R}^{17}$ is the *gt.* subregion affinity of \hat{v}_j^o as a one-hot vector. $\mathbb{1}_j^{VC}$ denotes that the *gt.* *VC* of \hat{v}_j^o is positive.

Anchor Elasticity. PiCR’s third outcome $AE \in \mathbb{R}^{N_o}$ stands

Algorithm 1: Procedure of recovering CPF

```

Input:  $\mathcal{V}^o, \mathcal{V}^h, VC, CR, AE$ 
Output:  $E_{elast}$ : elastic energy
1 recovery anchors:  $\mathcal{A} \leftarrow linear\_interpolation(\mathcal{V}^h);$ 
2 foreach  $j \in \{j \mid j \leq N_o, VC[j] > t_{vc}\}$  do
3   recover subregion id:  $r \leftarrow argmax(CR[j]);$ 
4   foreach  $a_i \in \mathcal{A}_r$  (anchors in subregion  $r$ ) do
5     recover elasticity:  $k_{ij}^{atr} \leftarrow AE[j];$ 
6      $E_{elast} + \leftarrow \frac{1}{2} * k_{ij}^{atr} \|a_i - v_j^o\|_2^2;$ 
7   foreach  $i \in \{i \mid i \leq N_h, \|v_i^h - v_j^o\|_2^2 \leq t_{rpl}\}$  do
8      $E_{elast} + \leftarrow \frac{1}{2} * k_{ij}^{rpl} |\exp(-(v_i^h - v_j^o) \cdot n_j)|^2;$ 

```

for the predicted elasticity of *attractive* springs k^{atr} . More specifically, $AE[j]$ is the elasticity k_{ij}^{atr} of an *attractive* spring that connects v_j^o to its affinitive anchor a_i in the predicted subregion: $argmax(CR[j])$. The loss function \mathcal{L}_{AE} is defined as a binary cross-entropy (BCE):

$$\mathcal{L}_{AE} = \sum_j^{N_o} \mathbb{1}_j^{VC} * \mathbb{1}_j^{img} * BCE(k_{ij}^{atr}, \hat{k}_{ij}^{atr}) \quad (7)$$

where the \hat{k}_{ij}^{atr} is the *gt.* elasticity described in §4.

With the predicted *VC*, *CR* and *AE*, as well as the coarse meshes $\mathcal{V}^o, \mathcal{V}^h$ in HoNet, PiCR finally recovers the CPF and collects the elastic energy E_{elast} as described in Algm.1. We empirically set the probability threshold of *VC*: $t_{vc} = 0.8$ and the distance threshold: $t_{rpl} = 20 mm$.

PiCR’s Framework. The proposed PiCR consists of a backbone b that extracts features from image, an encoder p that converts image features to object vertex features, and 3 heads h_{vc} , h_{cr} and h_{ae} which sequentially convert those features into *VC*, *CR*, and *AE*. As illustrated in Fig. 3, the process of feature extraction in PiCR can be expressed as:

$$\mathcal{F}' = [f(\pi(\mathcal{V}^o), b(\mathcal{I})), z(\mathcal{V}^o)]; \quad \mathcal{F} = p(\mathcal{F}') \quad (8)$$

where $b(\cdot)$ is the hourglass networks [40], $\pi(\cdot)$ is the perspective camera projection, and $f(\cdot)$ stands for aligning \mathcal{V}^o ’s 2D projection $\pi(\mathcal{V}^o)$ with the image features $b(\mathcal{I})$ through bilinear sampling. Inspired from Eq.(1) in [51], we also append the object’s root-relative z value $z(\mathcal{V}^o)$ at the end of $f(\cdot)$ to form the pixel-wise features \mathcal{F}' . Next, a PointNet [43] encoder $p(\cdot)$ is adopted to convert \mathcal{F}' to its point-wise features \mathcal{F} .

The process of three PiCR’s heads can be expressed as:

$$VC = h_{vc}(\mathcal{F}); \quad CR = h_{cr}(VC, \mathcal{F}); \quad AE = h_{ae}(CR, \mathcal{F}) \quad (9)$$

where all the heads are presented as multi-layer perceptrons. We provide implementation details in *Supp D.1*.

5.3. Grasping Energy Optimizer, GeO

The fitting part: Grasping Energy Optimizer (GeO) aims to refine the HO pose w.r.t. the recovered CPF. For the object part, we adjust its 6D pose $\mathbf{P}_o \in \mathfrak{se}(3)$. For the hand part, we jointly adjust the A-MANO’s 15 joint rotations $\{\mathbf{R}_j \in \mathfrak{so}(3) \mid j \leq 15\}$ and a wrist pose $\mathbf{P}_w \in \mathfrak{se}(3)$.

In order to mitigate the abnormal hand posture during optimization, we also define an anatomical cost function $\mathcal{L}_{\text{anat}}$ that penalizes the unwanted axial components and angular values of the 15 rotations in the proposed *twist-splay-bend* coordinate frame. First, for the joints along hand kinematic tree, we penalize the component of rotation axis \mathbf{a}^{rot} on *twist* direction: $\mathbf{n}^{\text{twist}}$, since any component that causes the finger twisting along its pointing direction is prohibited. Second, for the joints that do not belong to 5 knuckles, we also penalize the component of \mathbf{a}^{rot} on *splay* direction: $\mathbf{n}^{\text{splay}}$. Last, we penalize the rotation angle ϕ^{bend} that revolves about the *bend* axis if it is greater than $\pi/2$. The total anatomical cost can be written as:

$$\begin{aligned} \mathcal{L}_{\text{anat}} = & \sum_{j \in \text{all}} \mathbf{a}_j^{\text{rot}} \cdot \mathbf{n}_j^{\text{twist}} + \sum_{j \notin \text{knuck}} \mathbf{a}_j^{\text{rot}} \cdot \mathbf{n}_j^{\text{splay}} \\ & + \sum_{j \in \text{all}} \max \left((\phi_j^{\text{bend}} - \frac{\pi}{2}), 0 \right) \end{aligned} \quad (10)$$

We also penalize the offset of the refined hand-object vertices ${}^*\mathcal{V}^o, {}^*\mathcal{V}^h$ from their initial estimation $\mathcal{V}^h, \mathcal{V}^o$ in form of *l2* distance: $\mathcal{L}_{\text{offset}}$. We implement GeO in PyTorch with Adam solver. The whole optimization process can be expressed as:

$${}^*\mathcal{V}^o, {}^*\mathcal{V}^h \leftarrow \underset{\mathbf{P}_o, \mathbf{P}_w, \mathbf{R}_j}{\operatorname{argmin}} (E_{\text{elast}} + \mathcal{L}_{\text{anat}} + \mathcal{L}_{\text{offset}}) \quad (11)$$

6. Experiments and Results

6.1. Datasets

We would like to train and evaluate MIHO w.r.t. the real-world dataset that involves human hand interacting with textured object. In the community, there exist mainly four datasets that contain images and ground-truth 3D HO annotation, namely ObMan [25], FHB [19] and HO3D [22, 23] and ContactPose [7]. However, only FHB and HO3D satisfy our requirements in this study.

First-person Hand Action Benchmark, FHB. FHB is a first-person RGBD video dataset of hand in manipulation with objects. The ground-truth of hand poses was captured via magnetic sensors. In our experiments, we use a subset of FHB that contains 4 objects with a scanned model and pose annotation. We adopt the *action* split following the protocol given by [24, 54], and filter out the samples with a minimum HO distance greater than 5 mm, which yields us 7223 samples for training and 7373 for testing.

HO3D. HO3D is another dataset that contains precise hand-object pose during the interaction. Due to historical reasons, there are two versions of HO3D, namely v1 [22] and v2 [23]. In our experiments, we mainly compare our methods with the baseline [24] on HO3Dv1, but also conduct several comparisons with the recently released pre-trained model of [24] on HO3Dv2. Similar to FHB, we filter out samples with distance threshold 5mm. It’s also worth mentioning that, since our method requires a known object model, as well as a stable grasping configuration, nearly 5448 samples in HO3Dv2 test set are not suitable for our methods to report. Therefore, we manually select 6076 samples in HO3Dv2 test set to compare MIHO with [24]. We call this split by HO3Dv2⁻. Besides, training HO3Dv1 in previous methods [22, 24] requires an extra synthetic dataset that is not publicly available. Thus we manually augment the HO3Dv1 training set (referred as HO3Dv1⁺) and reproduce the results (referred as [24]⁺) comparable with those in [24]. Details of HO3Dv2⁻ selection and the augmentation procedures are provided in *Supp C.1, C.2*.

6.2. Metrics

Modeling the HO interaction requires not only a proper pose of both hand and object but also a natural grasp configuration. Here, we report 5 metrics in total that cover both reconstruction and grasp quality. Note that, since considering either of those metrics alone may yield misleading comparison, we consider them **together** for evaluation.

MPVPE. We compute the mean per vertex position error for both hand and object in camera space to assess the quality of pose estimation.

Penetration Depth (PD). To measure how deep that the hand is penetrating the object’s surface, we calculate the penetration depth that is the maximum distance of all the penetrated hand vertices to their closest object surface.

Solid Intersection Volume (SIV). To measure how much space intersection that occurs during estimation, we voxelize the object mesh into 80^3 voxels, and calculate the sum of the voxel volume inside the hand surface.

Disjointedness Distance (DD). We also encourage stable HO contact, which can be depicted as attracting fingertips onto the object surface. Therefore, we define the disjointedness metrics as the average distance of hand vertices in 5 fingertips region to their closest object surface.

Simulation Displacement (SD). We further evaluate the grasp stability in a modern physics simulator [11]. We measure the average displacement of object’s center over a fixed time period by holding the hand steadily and applying gravity to the object [25].

6.3. Comparison with State-of-the-Arts

For the FHB dataset, we compare our methods with the previous SOTA [24, 25] of hand-object reconstruction. For

Datasets		FHB					HO3Dv1 ⁺					HO3Dv2 ⁻	
Method		Ours [†]	Ours [‡]	gt.	[24]	ObMan*	Ours [†]	Ours [‡]	gt.	[24] ⁺	Ours [‡]	[24]	
Hand MPVPE (mm) ↓		21.16	19.54	0	17.51	18.42	24.56	23.99	0	24.80	-	-	
Object MPVPE (mm) ↓		21.06	21.57	0	21.06	21.17	18.10	19.15	0	18.10	73.28 ◇	75.77 ◇	
Penetra. depth (mm) ↓		16.13	16.92	19.55	20.63	19.76	11.87	11.42	7.55	18.57	16.47	20.02	
Solid intersec. vol. (cm^3) ↓		12.56	11.76	20.41	21.10	16.16	3.63	3.46	3.57	9.62	7.44	9.25	
Disjoint. distance (mm) ↓		24.54	22.41	37.28	37.40	27.95	11.71	11.83	14.53	18.62	37.04	41.41	
Displacement (mm) ↓		58.79	58.02	63.40	65.48	59.41	28.16	27.66	12.37	25.68	39.33	41.03	

Table 1. Quantitative results and detailed comparison with the previous state-of-the-art [24, 25] on the FHB and HO3D datasets. “gt.” denotes the ground-truth. “[†]” denotes ours *hand-alone* optimization setting and “[‡]” denotes the jointly *hand-object* setting. “*” denotes the reproduced ObMan [25]. “◇” denotes the *wrist-relative* object vertex error. “-” indicates the results that are not available.

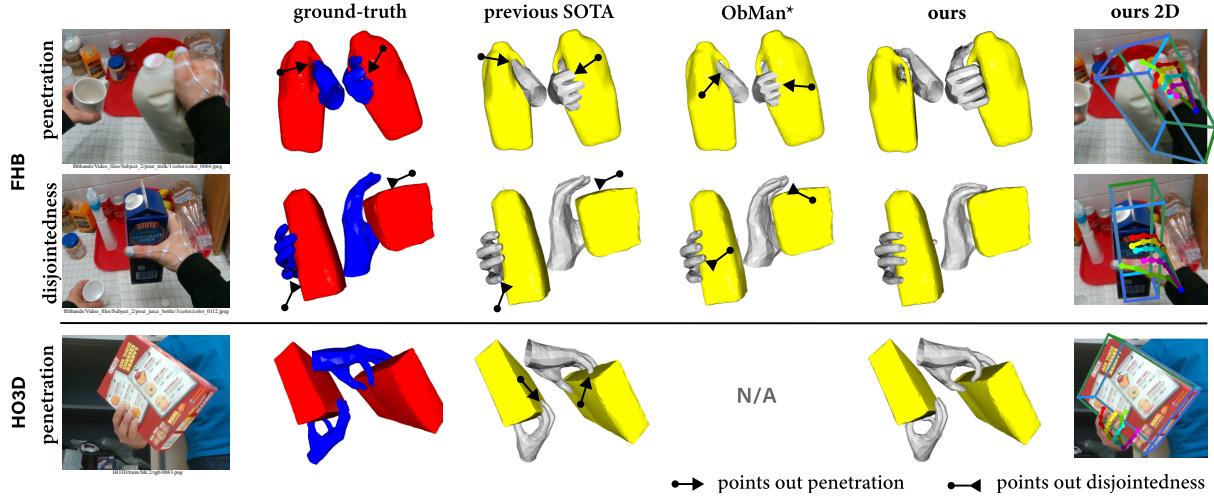


Figure 5. Qualitative Comparison with ground-truth and previous arts on the FHB and HO3D datasets.

[24], we select the results under the setting of full data supervision. Since [24] didn’t exploit any repulsion and attraction loss during training, direct comparisons on intersection and disjointedness may not be convincing enough. While the contact losses were considered in another work named ObMan [25], it only represented the genus 0 object mesh as a deformable icosphere, which is also not directly comparable with ours (known object model). To ensure rational comparisons, we migrate the *repulsion loss* and *attraction loss* from ObMan to the *MeshRegNet* in [24], and reproduce the results on par with it. We call this adaptation: ObMan*. For the HO3Dv1 dataset, we compare our results with the reproduced [24]⁺.

We report our results under two experimental settings: 1) **hand-alone** that fixes the object at the initial prediction in HoNet, and only optimizes the hand pose in GeO; 2) **hand-object** that jointly optimizes the hand and object poses in GeO. In Tab.1 we show our comparisons with the previous SOTA in all 5 metrics. For FHB dataset, as analyzed in [7], its ground-truth suffers from frequent interpenetration. We find that lower vertex error does not necessarily benchmark a higher reconstruction quality. Indeed, as shown in Tab.1 (col. 4, 5), either ground-truth or [24] reveals substantial

solid intersection volume, penetration depth and disjointedness. We find that MIHO outperforms [24] by a margin of 3.71 mm in penetration depth, 9.34 cm³ in solid intersection volume, and 14.99 mm in disjointedness distance, while only suffers from minor performance cost in hand MPVPE of 2.03 mm and object MPVPE of 0.51 mm. In the mean time, our simulation displacement also demonstrates the stability of our predicted grasp. These are consistent with our expectation that the CPF can by nature repulse intersection away and attract disjointedness to touch. As for HO3Dv1 testing set, our method also outperformed the previous SOTA over the most metrics. In terms fo simulation displacement, we found [24]⁺ slightly outperforms us by 1.98 mm. Based on our inspection in the Bullet [11] simulator, their stability are mainly attributed to the forces resulting from the intersection that balance each other. Visual comparisons are shown in Fig. 5. As for HO3Dv2, since we only test MIHO on the subset: HO3Dv2⁻, our results are not directly suitable for submitting to its online evaluation server. Thus, we only report the object 3D vertex errors on HO3Dv2⁻ based on the given annotation. We firstly align the predicted object vertex to the predicted hand wrist joint, then compute the *wrist-relative* object vertex er-

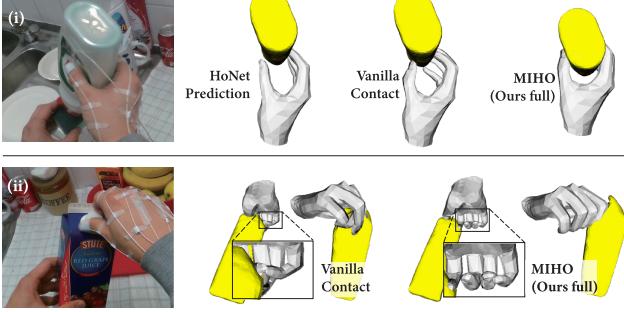


Figure 6. Comparisons of MIHO with simple contact heuristic.

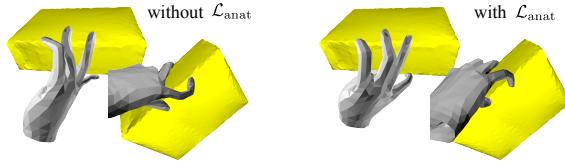


Figure 7. Example to illustrates the efficacy of our proposed A-MANO with anatomical constraints ($\mathcal{L}_{\text{anat}}$).

ror with those in ground-truth. Detailed comparisons are in Tab. 1 (col. 11, 12).

6.4. Ablation Study

In this experiment, we further evaluate the effectiveness of the proposed CPF and A-MANO. In the main text, we include three of the most representative studies. The ablation studies are mainly conducted on the FHB test set with *action* split. For more studies on 1) impact from the magnitude of k^{rpl} ; 2) A-MANO with PCA pose; 3) unwanted twist correction; please visit *Supp D.2*.

Comparison with simple Distance-based Contact Heuristics. To show the superiority of the CPF over the distance-based contact heuristics, we compare the fitting stage of MIHO with two simple yet strong baselines: (a) *Vanilla Contact* that removes the E_{elast} term in Eq. 11 and purely attracts the anchors on fingertips to its nearest object vertex (similar to [1]) in a given threshold which we set as 20 mm; (b) *ObMan Contact* that replaces E_{elast} in Eq. 11 by the well-designed interaction losses in ObMan [25]. All the three experiments start from the same HO pose predicted by HoNet (§5.1). We show in Tab. 2 that by exploiting CPF, MIHO can surpass the simple baselines on most of the metrics. Note that, since both (a) and (b) directly optimize the disjointedness term, their results show better resistance on it. The last column in Tab. 2 shows that our methods can save average time per iteration by 46% compared with *ObMan Contact*. We also conduct two qualitative comparisons in Fig. 6. The first one shows that CPF can learn the contact *semantics* to guide the optimization that better matches visual cues, whereas the *Vanilla Contact* fails to form a valid grasp. The second

Settings	Scores					$t_{\text{iter}}(ms)$
	HE ↓	OE ↓	PD ↓	SIV ↓	DD ↓	
MIHO (ours full)	19.54	21.57	16.92	11.76	22.41	55.77
(a) Vanilla Contact	24.01	24.29	18.36	15.64	16.32	45.40
(b) ObMan Contact	22.15	22.54	15.13	16.20	11.97	103.41

Table 2. Ablation study on the different contact heuristics. HE, OE stands for 3D hand and object vertex error. PD, SIV and DD are the abbreviation of metrics in §6.2

Settings	PD↓	SIV↓	DD↓
with E^{rpl} (ours full)	16.92	11.76	22.41
without E^{rpl}	17.79	13.76	20.27
gt. FHB	19.55	20.41	37.28

Table 3. Ablation study on the *repulsive* springs.

shows that CPF can maintain subtle interaction, as no attraction will be applied on those non-affinitive vertex pairs (see ring and pinky fingers when unscrewing the juice cap).

Effectiveness of Repulsive Springs. To measure the efficacy of *repulsive* springs in CPF, we remove all the repulsion energy E^{rpl} induced by them, leaving the attraction as the unique type of energy applied on hand and object. As we expected, the result in Tab. 3 witnesses the accumulation of PD and SIV. To note, even without the *repulsive* springs, we still witness a remarkable improvement of PD and SIV over the FHB ground-truth. This is attributed to the repulsive behavior of the *attractive* springs: when hand is inside the object surface, the energy stored in the *attractive* springs will act as repulsion that pushes out the hand.

Effectiveness of the Anatomical Constraints. We further highlight the efficacy of adopting the anatomical constraints. We conduct a contrastive experiment whose only difference is the absence of $\mathcal{L}_{\text{anat}}$. Both experiments start from a zero (flat) hand and minimize the E_{elast} based on the same predicted CPF. We show in Fig. 7 that the anatomical constraints are able to effectively prevent abnormality during the optimization.

7. Conclusion

In this work, we propose a novel contact representation named CPF and a learning-fitting hybrid framework MIHO to help modeling hand and object interaction. Comprehensive evaluations show that our methods, while being able to recover precise hand-object pose, can also effectively 1) avoid interpenetration and control disjointedness, and 2) prevent abnormality in hand pose. We hope CPF can serve as an effective contact representation for future works on hand-object interaction. Later, we also plan to develop for an object-agnostic representation of CPF, for the interaction in general cases.

References

- [1] Dafni Antotsiou, Guillermo Garcia-Hernando, and Tae-Kyun Kim. Task-oriented hand motion retargeting for dexterous manipulation imitation. In *ECCV Workshops*, 2018. [2](#), [3](#), [8](#)
- [2] Seungryul Baek, Kwang In Kim, and Tae-Kyun Kim. Pushing the envelope for rgb-based dense 3d hand pose estimation via neural rendering. In *CVPR*, 2019. [2](#)
- [3] Christoph W Borst and Arun P Indugula. Realistic virtual grasping. In *VR*, 2005. [2](#)
- [4] Adnane Boukhayma, Rodrigo de Bem, and Philip HS Torr. 3d hand shape and pose from images in the wild. In *CVPR*, 2019. [1](#), [2](#)
- [5] Samarth Brahmbhatt, Cusuh Ham, Charles C Kemp, and James Hays. Contactdb: Analyzing and predicting grasp contact via thermal imaging. In *CVPR*, 2019. [1](#)
- [6] Samarth Brahmbhatt, Ankur Handa, James Hays, and Dieter Fox. Contactgrasp: Functional multi-finger grasp synthesis from contact. In *IROS*, 2019. [1](#), [2](#), [4](#)
- [7] Samarth Brahmbhatt, Chengcheng Tang, Christopher D. Twigg, Charles C. Kemp, and James Hays. ContactPose: A dataset of grasps with object contact and hand pose. In *ECCV*, 2020. [1](#), [3](#), [6](#), [7](#)
- [8] Minjie Cai, Kris M Kitani, and Yoichi Sato. Understanding hand-object manipulation with grasp types and object attributes. In *RSS*, 2016. [2](#)
- [9] Minjie Cai, Kris M Kitani, and Yoichi Sato. An ego-vision system for hand grasp analysis. *IEEE Transactions on Human-Machine Systems*, 2017. [2](#)
- [10] Chiho Choi, Sang Ho Yoon, Chin-Ning Chen, and Karthik Ramani. Robust hand pose estimation during the interaction with an unknown object. In *ICCV*, 2017. [2](#)
- [11] Erwin Coumans and Yunfei Bai. Pybullet, a python module for physics simulation in robotics, games and machine learning, 2017. [6](#), [7](#)
- [12] 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. [13](#)
- [13] Bardia Doosti, Shujon Naha, Majid Mirbagheri, and David J Crandall. Hope-net: A graph-based model for hand-object pose estimation. In *CVPR*, 2020. [1](#), [2](#)
- [14] George ElKoura and Karan Singh. Handrix: animating the human hand. In *SIGGRAPH*, 2003. [2](#)
- [15] Haoqiang Fan, Hao Su, and Leonidas J Guibas. A point set generation network for 3d object reconstruction from a single image. In *CVPR*, 2017. [1](#)
- [16] Thomas Feix, Ian M Bullock, and Aaron M Dollar. Analysis of human grasping behavior: Object characteristics and grasp type. *IEEE transactions on haptics*, 2014. [2](#)
- [17] Hongyang Gao and Shuiwang Ji. Graph u-nets. *ICLR*, 2019. [2](#)
- [18] Guillermo Garcia-Hernando, Edward Johns, and Tae-Kyun Kim. Physics-based dexterous manipulations with estimated hand poses and residual reinforcement learning. *arXiv preprint arXiv:2008.03285*, 2020. [1](#), [2](#)
- [19] 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. [2](#), [6](#), [15](#), [16](#)
- [20] Lihao Ge, Zhou Ren, Yuncheng Li, Zehao Xue, Yingying Wang, Jianfei Cai, and Junsong Yuan. 3d hand shape and pose estimation from a single rgb image. In *CVPR*, 2019. [1](#)
- [21] Thibault Groueix, Matthew Fisher, Vladimir G Kim, Bryan C Russell, and Mathieu Aubry. A papier-mâché approach to learning 3d surface generation. In *CVPR*, 2018. [1](#)
- [22] 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. [2](#), [6](#), [14](#), [16](#)
- [23] Shreyas Hampali, Mahdi Rad, Markus Oberweger, and Vincent Lepetit. Honnote: A method for 3d annotation of hand and object poses. In *CVPR*, 2020. [2](#), [4](#), [6](#), [15](#), [16](#)
- [24] 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. [1](#), [2](#), [4](#), [6](#), [7](#), [13](#)
- [25] 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. [1](#), [2](#), [3](#), [6](#), [7](#), [8](#), [13](#)
- [26] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. [13](#)
- [27] Markus Höll, Markus Oberweger, Clemens Arth, and Vincent Lepetit. Efficient physics-based implementation for realistic hand-object interaction in virtual reality. In *VR*, 2018. [2](#)
- [28] Korrawe Karunratanakul, Jinlong Yang, Yan Zhang, Michael Black, Krikamol Muandet, and Siyu Tang. Grasping field: Learning implicit representations for human grasps. In *3DV*, 2020. [1](#), [2](#)
- [29] Jun-Sik Kim and Jung-Min Park. Physics-based hand interaction with virtual objects. In *ICRA*, 2015. [2](#)
- [30] Mia Kokic, Danica Kragic, and Jeannette Bohg. Learning to estimate pose and shape of hand-held objects from rgb images. *arXiv preprint arXiv:1903.03340*, 2019. [1](#), [2](#)
- [31] Mia Kokic, Danica Kragic, and Jeannette Bohg. Learning task-oriented grasping from human activity datasets. *RAL*, 2020. [1](#), [2](#)
- [32] Paul G Kry and Dinesh K Pai. Interaction capture and synthesis. *TOG*, 2006. [2](#)
- [33] Dominik Kulon, Riza Alp Guler, Iasonas Kokkinos, Michael M Bronstein, and Stefanos Zafeiriou. Weakly-supervised mesh-convolutional hand reconstruction in the wild. In *CVPR*, 2020. [1](#)
- [34] Nikolaos Kyriazis and Antonis Argyros. Physically plausible 3d scene tracking: The single actor hypothesis. In *CVPR*, 2013. [2](#)
- [35] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollar. Focal loss for dense object detection. In *ICCV*, 2017. [5](#)

Appendix

In the supplemental document, we provide:

- §A Anatomically Constrained A-MANO.
- §B Detailed Analysis of the Spring’s Elasticity.
- §C Detailed Analysis of the HO3D Dataset.
- §D More Experiments and Results.
- §E More Qualitative Results.

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 [50] 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*². 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. 8 left). Different from MANO, we follow the Universal Robot Description Format (URDF) [37] that describe each articulation along the hand kinematic tree as a revolute joint³. 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. 8 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. 9.

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

²Rotation can be represented as rotating along an *axis* by an *angle*.

³https://en.wikipedia.org/wiki/Revolute_joint

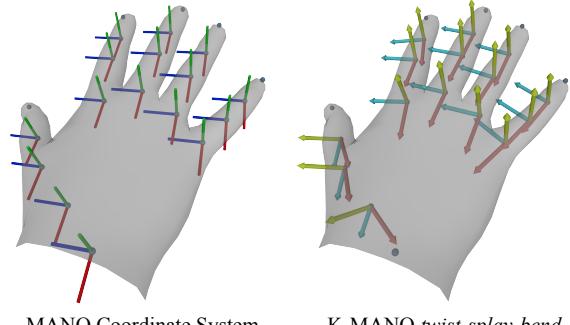


Figure 8. Visual comparison of MANO’s coordinate system to the proposed *twist-splay-bend* system.

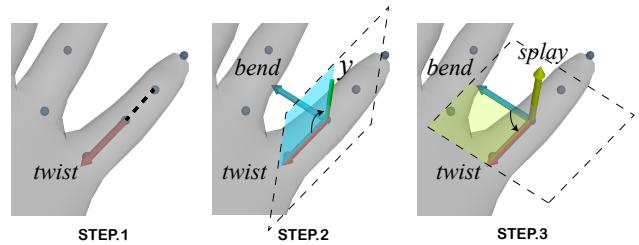


Figure 9. Illustration of converting MANO’s coordinates system to the proposed *twist-splay-bend* system.

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. 10. For clarity, we number the MANO links from 1 to 20 as illustrated in Fig. 11 (left).

To assign the MANO vertices to its corresponding region, 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.11 (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 opti-

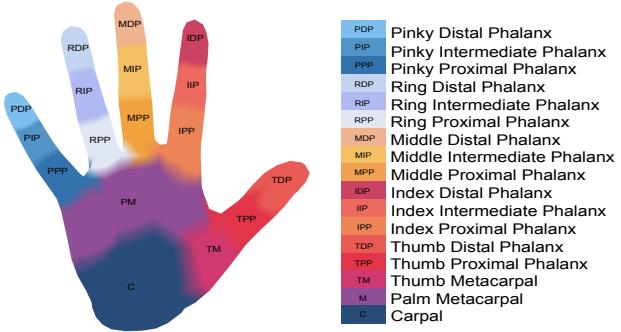


Figure 10. Hand subregions with names and abbreviations.

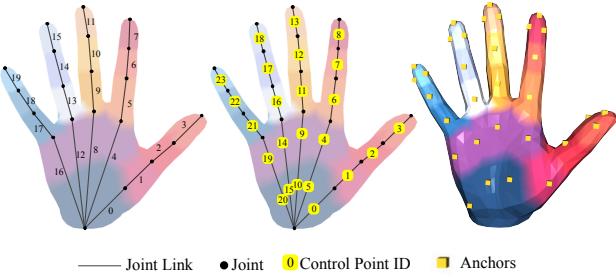


Figure 11. Left: joint links with ID; Middle: control points with ID; Right: anchors

mization, 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 $f \in \mathbf{F}$ of mesh whose vertices are stored in order: $f = \{i_k\}, v_k = \mathbf{V}[i_k], k \in \{1, 2, 3\}$. We can get two edges of that face: $e_1 = v_2 - v_1, e_2 = v_3 - v_1$. Then the local position of the anchor \tilde{a} inside the face can be represented by linear interpolation of e_1 and e_2 : $\tilde{a} = x_1 e_1 + x_2 e_2$, where the x_1, x_2 are some weights. Finally, the global position of the anchor a will be $a = v_1 + \tilde{a} = v_1 + x_1 e_1 + x_2 e_2 = (1 - x_1 - x_2) v_1 + x_1 v_2 + x_2 v_3$. During the optimization process, we can use the precomputed face 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 co-

ordinate system. As illustrated in Fig.11 (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.

B. Spring's Elasticity

B.1. Elastic Energy Analysis

Here we illustrate elastic energy between a pair of points v_i^h and 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 n_j^o representing the normal direction at this vertex (also the direction of repulsion). Then we compute the offset vector $\Delta l_{ij}^{atr} = v_i^h - v_j^o$, and the projection of the offset vector on object normal n_j^o : $|\Delta l_{ij}^{rpl}| = (v_i^h - v_j^o) \cdot n_j^o$. $|\Delta l_{ij}^{rpl}|$ is positive if v_i^h falls outside the object, and negative if v_i^h falls inside the object. We use an exponential function here to provide magnitude and gradient heuristic for optimizer: **a)** the less $|\Delta l_{ij}^{rpl}|$ is, the more v_i^h penetrates into the object. The gradient of repulsive energy will be an exponential increasing function of Δl_{ij}^{rpl} . **b)** when v_i^h intersects into the object, both the repulsion and the attraction will push v_i^h towards the surface; when 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. 12.

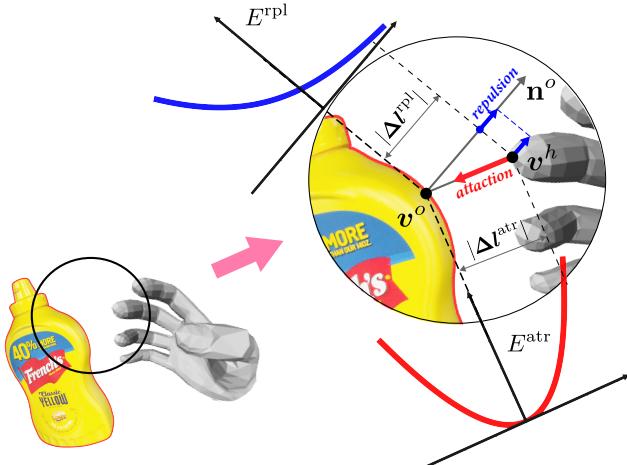


Figure 12. Illustration of the elastic energy w.r.t. a pair of hand-object vertices.

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 $\hat{k}_{ij}^{\text{atr}}$. First, we set the anchor a_i - vertex v_j^o pair with ground-truth distance $|\Delta l_{ij}^{\text{atr}}| > 20\text{mm}$ 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 $|\Delta l_{ij}^{\text{atr}}|$:

$$\hat{k}_{ij}^{\text{atr}} = 0.5 * \cos\left(\frac{\pi}{s} * |\Delta l_{ij}^{\text{atr}}|\right) + 0.5 \quad (12)$$

where the scale factor $s = 20\text{ mm}$.

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 $|\Delta l| = 0$; **b**) k is inverse proportional to $|\Delta l|$ 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.

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. 13. Table.4 shows our final selection on HO3Dv2 test set, as we called HO3Dv2⁻.



Figure 13. Unsuitable samples in HO3Dv2 testing set.

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

Table 4. **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 around the object. Specifically, as we show in Fig. 14, 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 [25] Eq. 4, and retain the repulsion energy and the anatomical cost. The optimization process of grasping generation can be expressed as:

$$\hat{\mathcal{V}}^h \leftarrow \underset{(\mathbf{P}_w, \mathbf{R}_j)}{\operatorname{argmin}} (\mathcal{L}_A + E^{\text{rpl.}} + \mathcal{L}_{\text{anat}}) \quad (13)$$

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 [26] backbone initialized with ImageNet [12] pretrained weights. For FHB and HO3Dv2 dataset, we use the pre-trained weights released from [24]. 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 [40] (with 2 stacks) as backbone, a PointNet [43] as the point encoder, and three multi-layer perceptrons as

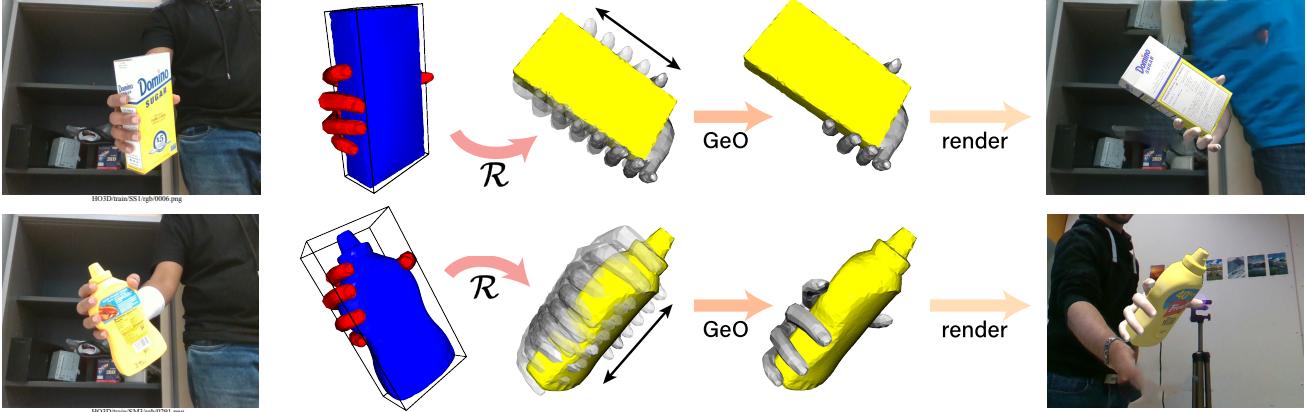


Figure 14. **HO3D [22] Dataset augmentation.** We demonstrate the process of generating synthetic training images. \mathcal{R} stands for the random transformation.

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 follow-

ing 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 $k^{\text{rpl}} = 1 \times 10^{-3}$ is in accord with the default experiment in main text. As shown in Tab. 5, 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
1.0×10^{-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 5. **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. 6 shows that our full MANO can achieve a

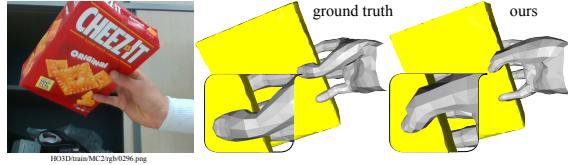


Figure 15. Example to show that our A-MANO can mitigate the unwanted twist (see thumb) exhibited in ground-truth.

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 6. **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. 15, since A-MANO imposes constraints on the *twist* component of the rotation axis, it can achieve a more visually pleasing result in such case.

E. More Qualitative Results

We demonstrate the qualitative results of MIHO in Fig. 16 on both the FHB [19] and HO3D dataset [23]. Note that the ground truth of the test set in HO3Dv2⁻ [23] is not available.

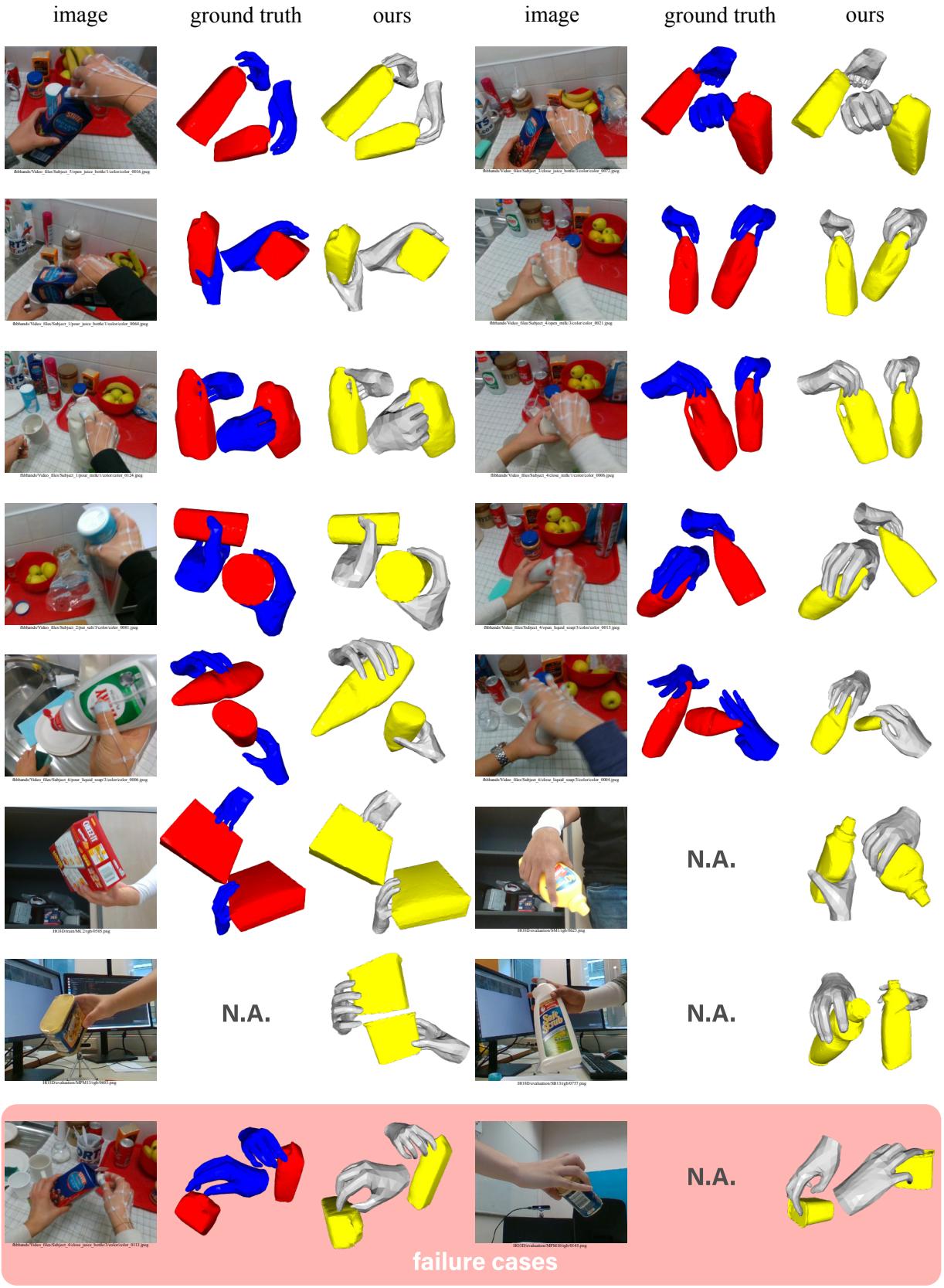


Figure 16. Qualitative results on FHB [19], HO3Dv1[22] and HO3Dv2⁻ [23] datasets. The last row shows the failure cases.