Interacting Hand-Object Pose Estimation via Dense Mutual Attention

Rong Wang Wei Mao Hongdong Li
The Australian National University
{rong.wang, wei.mao, hongdong.li}@anu.edu.au

Abstract

3D hand-object pose estimation is the key to the success of many computer vision applications. The main focus of this task is to effectively model the interaction between the hand and an object. To this end, existing works either rely on interaction constraints in a computationally-expensive iterative optimization, or consider only a sparse correlation between sampled hand and object keypoints. In contrast, we propose a novel dense mutual attention mechanism that is able to model fine-grained dependencies between the hand and the object. Specifically, we first construct the hand and object graphs according to their mesh structures. For each hand node, we aggregate features from every object node by the learned attention and vice versa for each object node. Thanks to such dense mutual attention, our method is able to produce physically plausible poses with high quality and real-time inference speed. Extensive quantitative and qualitative experiments on large benchmark datasets show that our method outperforms state-of-the-art methods. The code is available at `https://github.com/rongakowang/DenseMutualAttention.git`.

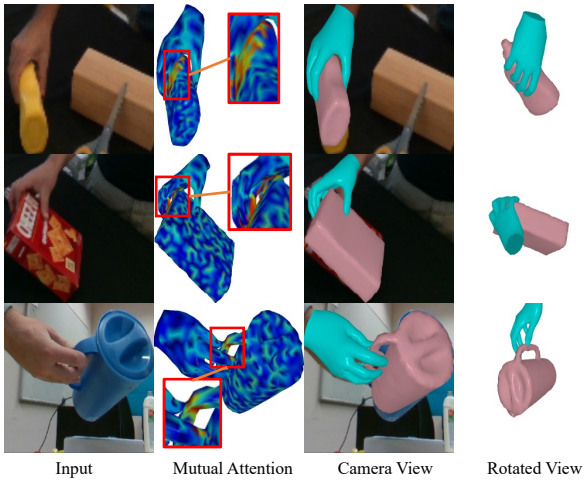

Figure 1. **Effects of the mutual attention.** Our method recovers accurate hand-object pose via dense mutual attention between all hand and object vertices. We visualize in the second column the learned average mutual attention for contacting vertices, where red regions have higher attention values and blue regions have lower values. We observe the proposed mutual attention can effectively model interaction around contacting areas. In addition, it helps to select secondary keypoints (yellow regions with medium attention values) that facilitate hand-object pose refinement.

1. Introduction

Accurate and efficient pose estimation for the scene of a hand interacting with an object from a single monocular view is desired in many applications, *e.g.* extended reality (XR) [38] and human-computer iteration (HCI) [24]. Despite that great efforts have been contributed to developing effective 3D hand pose estimation algorithms [17, 25, 40, 50, 47], joint hand-object pose estimation remains especially challenging due to the severe mutual occlusion and diverse ways of hand-object manipulation. Methods failing to tackle the aforementioned challenges tend to produce physically implausible configurations, such as interpenetration and out-of-contact. To avoid generating undesired poses, an in-depth understanding of the correlation between the hand and the interacting object is therefore required.

Research works on 3D hand-object pose estimation can be categorized as optimization-based and learning-based. While the former methods [48, 13, 10] generalize to diverse object classes, the optimization process requires multiple iterations to converge, which is not applicable for real-time applications like XR. In contrast, learning-based methods [26, 14, 12, 8, 11] can achieve real-time inference. Motivated by the optimization-based methods, soft contact losses are introduced [14, 12] to implicitly guide the network to pursuit plausible hand-object interaction. For a more effective modeling, other works focus on explicitly learning the hand-object correlation [8, 6] in the design of the network. Recently, several attention-based works [41, 11] are proposed considering its efficacy in modelling complex correlation. In [41] a self-attention mechanism is used to capture feature dependencies for either the hand or the object and the interaction between them is modeled by the exchange of global features. Most close to our

work is [11] where a cross-attention is used to model the correlation between the hand and the object. However, all above methods only model a *sparse* interaction between a pre-defined set of keypoints or features from the hand and the object, regardless of the fact that hand-object interaction actually occurs on physical regions of the surfaces.

In this work, we instead propose to model fine-grained hand-object interaction via a *dense mutual attention* mechanism. Specifically, we first estimate rough hand and object meshes separately from a single monocular image. Next, we construct the hand and object graphs based on their mesh structures, then spatially sample node features according to the rough mesh positions. Unlike [41] which transfers inter-graph dependencies via global features only, we allow direct node-to-node feature aggregation via mutual attention. Taking a node from the hand graph as an example, we calculate the object-to-hand attention for all object nodes, and then fuse the hand node feature with attention-weighted object node features to explicitly model the fine-grained interaction correlation. A similar calculation is performed to refine object node features given hand-to-object attention. Finally, we refine the hand and object poses through graph convolutional blocks equipped with the proposed mutual attention layer. We show that our method does not require iterative optimization as in [48, 13], and the dense vertex-level mutual attention can model the hand-object interaction more effectively than sparse keypoints based methods [11, 8]. In summary, our contributions are as follows.

- We propose a novel dense mutual attention mechanism that effectively models hand-object interaction by aggregating and transferring node features between the hand and object graphs.
- We design a novel hand-object pose estimation pipeline facilitating the proposed mutual attention. Extensive experiments show superior results compared to state-of-the-art methods on large benchmark datasets.

2. Related Works

In this section, we review related works on hand-object pose estimation. Since our work relies on graph convolutional networks and the attention mechanism, we also review their utilization in related tasks.

2.1. Hand-Object Pose Estimation

Most previous works tackle 3D hand pose estimation [17, 25, 40, 50, 47] and object pose estimation [27, 31, 44, 49] separately. Recently joint hand-object pose estimation has received more focus [14, 26, 28, 12, 8, 13, 11] due to the strong correlation when hands interact with objects. For learning-based methods, Hasson *et al.* [14] propose attraction and repulsion losses to penalize physically implausible reconstructions. Shaowei *et al.* [28] adopt a semi-supervised learning framework with contextual reasoning of hand and object representations. Hasson *et al.* [12] extend to video inputs by leveraging photometric and temporal consistency on sparsely annotated data. To tackle the lack of 3D ground truth, Kailin *et al.* [26] introduce an online synthesis and exploration module to generate synthetic hand-object poses from a predefined set of plausible grasps during training. In contrast to the above works, optimization-based methods [13, 48, 10] formulate the task by firstly estimating initial hand and object poses in isolation, then jointly refining them with contact constraints. However, these methods are time-consuming as the optimization process generally requires multiple iterations to converge, thus limiting their applications in real-time XR systems. In consequence, we adopt the learning-based framework and continue to introduce related works in this category in the following section.

2.2. GCNs-based Methods

Graph Convolutional Networks (GCNs) have been wildly applied in 3D hand pose estimation [9, 40, 20, 5] since hand meshes and kinematic trees naturally form a graph. Several works have extended GCNs to hand-object pose estimation and achieved promising results. Bardia *et al.* [8] build an adaptive Graph-UNet (HOPE-Net) combining hand joints and object bounding box corners with learnable adjacent matrices. Lin *et al.* [16] encode initial 2D poses with GCNs similar to HOPE-Net as priors for the following 3D reconstruction in a non-autoregressive Transformer. However, the aforementioned methods only construct sparse graphs from hand-object interaction scenes and do not estimate hand shapes, thus lacking expressiveness. Tze *et al.* [41] propose a collaborative method to iteratively refine results from dense hand and object graphs. However, the iterative refinement is computationally expensive, and the model-free approach in object representation often does not recover accurate object shapes.

2.3. Attention-based Methods

Attention mechanism [43] has shown remarkable success in human body [7, 23] and hand pose [30] estimation as it can effectively model long-range correlation and aggregate component features. Hampali *et al.* [11] propose to learn attention between a sparse set of sampled hand and object keypoints. In [41], an attention-guided GCN is proposed to effectively aggregate vertex features within either hand or object graphs. The interaction between the hand and the object is explored via the exchange of global features during the iterative process. In contrast, we propose to exploit mutual attention between every hand and object vertex that better learns the interaction dependencies.

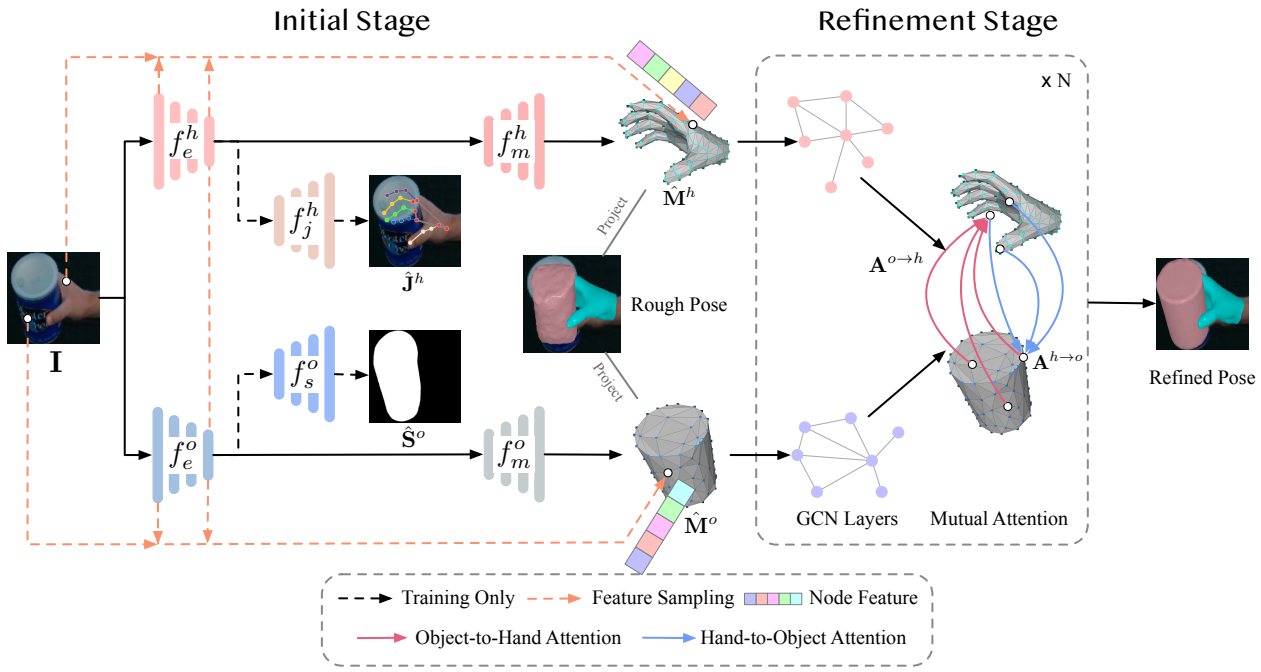

Figure 2. **Overview of our method.** Our model consists of two stages. At the initial stage (left), we use two separate branches to estimate rough meshes of hand and object ($\hat{\mathbf{M}}^h$ and $\hat{\mathbf{M}}^o$), respectively in the mesh estimator $f_m^h(\cdot)$ and $f_m^o(\cdot)$. Each estimator takes image features from the encoder $f_e^h(\cdot)$ and $f_e^o(\cdot)$ as the input. To supervise the feature extraction, we include two additional estimators ($f_j^h(\cdot)$ and $f_s^o(\cdot)$) to estimate 3D hand joints ($\hat{\mathbf{J}}^h$) and object silhouette ($\hat{\mathbf{S}}^o$) during training. At the refinement stage (right), we first construct hand and object graphs according to the mesh structures. The initial feature of each node in the graphs is sampled from the input image $\mathbf{I}$ as well as feature maps of the image encoders according to the pixel location projected from the meshes. Finally, we leverage a stack of GCN layers followed by the proposed mutual attention layer to generate the refined hand and object poses.

3. Methods

In this section, we introduce the training pipeline as shown in Figure 2. Our model consists of two stages. At the initial stage, we first separately estimate the rough hand (Section 3.1) and object pose (Section 3.2) given an input RGB image $\mathbf{I} \in \mathbb{R}^{H \times W \times 3}$. Combining rough poses from both branches, we then jointly refine them at the refinement stage using a graph convolution network equipped with the proposed mutual attention layer (Section 3.3) to explicitly model the hand-object interaction. The final outputs of the refinement stage are the 3D vertex coordinates of the hand mesh defined in the MANO [35] model and the 6D object pose in $SE(3)$ that transforms the object CAD model into the camera frame. We train the proposed model end-to-end with a multi-task training objective (Section 3.4). For the consistency of notations, we use the superscript h and o to indicate the hand and object branch respectively.

3.1. Hand Pose Estimation

Following [29], we propose to represent a hand mesh via *lixels*. Specifically, we define the position of a 3D vertex $\mathbf{x} = [u, v, z]^T \in \mathbb{R}^3$ as its projected pixel coordinates (u, v) and depth (z). We then quantize the pixel coordinates and depth and into 3 independent 1D heatmap vectors $(\mathbf{l}_u, \mathbf{l}_v, \mathbf{l}_z)$, where $\mathbf{l}_u, \mathbf{l}_v, \mathbf{l}_z \in \mathbb{R}^L$. After scaling and normalization via the softmax operation, each entry (known as a lixel) of the heatmap vectors represents the probability of the pixel location or the depth for the vertex. Given such lixels, the vertex position can be computed with the soft-argmax [4] operation as:

$$u = \frac{W}{L} \cdot \text{soft-argmax}(\mathbf{l}_u) \,, \tag{1}$$

$$v = \frac{H}{L} \cdot \text{soft-argmax}(\mathbf{l}_v) \,, \tag{2}$$

$$z = \frac{2D}{L} \cdot \text{soft-argmax}(\mathbf{l}_z) + r_z - D \,, \tag{3}$$

where W and H are the width and height of the image. L is the quantization level. D is the depth radius[1] relative to the wrist joint estimated from the training data, and r_z is the wrist joint depth[2], which is assumed to be known [26, 40] to resolve the scale ambiguity in the single view input.

Given the camera intrinsic $\mathbf{K}$, pixel coordinates, and depth, we can easily recover the 3D vertex's Euclidean coordinates in the camera space. As shown in [29], such repre-

[1]We therefore quantize the depth ranging in $[r_z - D, r_z + D]$.
[2]Relative depth in the object mesh also refers to the wrist joint.

sentation is more robust and effective than directly regressing 3D coordinates, and is more memory-efficient than 3D voxel representation as it decouples the three components. Unless otherwise specified, throughout the rest sections, our model will produce the 3 vectors $(\mathbf{l}_u, \mathbf{l}_v, \mathbf{l}_z)$ when estimating mesh vertices and hand joints. Those vectors will then be converted to the vertex position (u, v, z) using the equations 1,2 and 3.

Recall that, at the initial stage, we use two separate branches to estimate rough hand and object meshes. In particular, given the input image $\mathbf{I}$, the hand pose estimation branch first extracts image features using an image feature encoder $f_e^h(\cdot)$:

$$\{\mathbf{F}_{(i)}^h\} = f_e^h(\mathbf{I}) \ , \tag{4}$$

where $f_e^h(\cdot)$ is implemented as a ResNet-50 [15] encoder pre-trained on the ImageNet [37] and $\{\mathbf{F}_{(i)}^h\}$ denotes the collection of feature maps extracted from the i-th layer of the encoder. In particular, we denote the image feature map from the final layer as $\mathbf{F}^h$ for succinct notions.

To guide the feature extraction, we additionally feed the estimated image feature from the final layer in a hand joint estimator $f_j^h(\cdot)$:

$$\hat{\mathbf{J}}^h = f_j^h(\mathbf{F}^h) \ , \tag{5}$$

where $\hat{\mathbf{J}}^h \in \mathbb{R}^{21\times 3}$ are the estimated positions of 21 hand joints. Note that, the joint estimator is only used for the purpose of feature extraction supervision in training. During testing, the entire joint estimator is removed.

Finally, given the final image feature $\mathbf{F}^h$, we obtain a rough hand mesh $\hat{\mathbf{M}}^h \in \mathbb{R}^{778\times 3}$ from the hand mesh estimator $f_m^h(\cdot)$:

$$\hat{\mathbf{M}}^h = f_m^h(\mathbf{F}^h) \ . \tag{6}$$

3.2. Object Pose Estimation

Similar to the hand pose estimation branch, we first extract image features through an image encoder $f_e^o(\cdot)$ which has the same architecture as $f_e^h(\cdot)$ but does not share weights:

$$\{\mathbf{F}_{(i)}^o\} = f_e^o(\mathbf{I}) \ . \tag{7}$$

We also use $\mathbf{F}^o$ to denote the feature map extracted from the final layer of $f_e^o(\cdot)$.

Since there are no unanimous keypoints defined for all classes of objects, we alternatively use the object silhouette to supervise the feature extraction. Specifically, we design the object mask estimator $f_s^o(\cdot)$ taking of the input as $\{\mathbf{F}_{(i)}^o\}$. Following the image segmentation literature [36], we include skip-connections from the image encoder to the mask estimator. Hence all image features are forwarded into the estimator to obtain the object silhouette $\hat{\mathbf{S}}^o \in \mathbb{R}^{H\times W}$ as:

$$\hat{\mathbf{S}}^o = f_s^o(\{\mathbf{F}_{(i)}^o\}) \ . \tag{8}$$

Similarly, we construct the object mesh estimator $f_m^o(\cdot)$ symmetric to $f_m^h(\cdot)$. When estimating the object mesh, we follow the previous work [13] to assume that the object CAD models are given and the meshes are resampled to have 1000 vertices using ACVD [42] for the convenience of batch training. The object mesh $\hat{\mathbf{M}}^o \in \mathbb{R}^{1000\times 3}$ can be computed as:

$$\hat{\mathbf{M}}^o = f_m^o(\mathbf{F}^o) \ . \tag{9}$$

Note that, at the initial stage, instead of directly regressing the target rough 6D object pose, we adopt a model-free approach as used in [14] to estimate rough object meshes. Empirically, we find that such a strategy is more robust and it better facilitates feature sampling introduced in the following section.

3.3. Hand-Object Pose Refinement

Given the rough meshes of the hand $\hat{\mathbf{M}}^h$ and the object $\hat{\mathbf{M}}^o$, we then jointly refine them by exploiting their correlations. To this end, we regard those meshes as two graphs and propose to use the graph convolutional network (GCN) [22] to capture the intra-graph dependencies. To further model the inter-graph interaction, we propose a novel mutual attention layer that allows fine-grained feature aggregation between two graphs.

Graph Construction. As shown in Figure 2, the hand and the object are modeled by separate graphs with vertices as nodes and their connections defined in the mesh structures as edges. Vertices belonging to different branches are disconnected and communicate via mutual attention. Motivated by [40], we initialize the features of each graph node from the feature extraction module at the initial stage. Taking the hand graph for example, given the pixel coordinates of n-th node $\mathbf{v}_n = [u_n, v_n]^T$ in the rough mesh $\hat{\mathbf{M}}^h$, we spatially sample local features from image features $\{\mathbf{F}_{(i)}^h\}$ using a bilinear interpolation operation $f_b(\cdot)$. In the meantime, we fuse final image features from both the hand and object branches to obtain a global feature containing the global information of the hand and object mesh structures. The initial node features $\mathbf{h}_n^h$ is computed as a concatenation of the local and global features:

$$\mathbf{h}_n^h = f_b(\mathbf{I}(\mathbf{v}_n)) \oplus f_b(\{\mathbf{F}_{(i)}^h(\mathbf{v}_n)\}_{i\in\mathcal{X}}) \oplus f_g(\mathbf{F}^h + \mathbf{F}^o) \ , \tag{10}$$

where $\mathbf{h}_n^h \in \mathbb{R}^K$, $\mathcal{X}$ is a set of layer indices from which we sample feature maps, $f_g(\cdot)$ is a global feature fusion unit, and $\oplus$ denotes the concatenation operation.

For the m-th node of the object graph, we compute the initial feature $\mathbf{h}_m^o \in \mathbb{R}^K$ in a similar way:

$$\mathbf{h}_n^h = f_b(\mathbf{I}(\mathbf{v}_m)) \oplus f_b(\{\mathbf{F}_{(i)}^o(\mathbf{v}_m)\}_{i\in\mathcal{X}}) \oplus f_g(\mathbf{F}^h + \mathbf{F}^o) \ , \tag{11}$$

Graph Convolutional Layer. After initializing the node features, we then follow [46] to update the node features

via graph convolutional layers. For hand nodes, the feature updating can be expressed as:

$$\mathbf{h}_n'^{h} = \text{MLPs}^h(\mathbf{h}_n^h + \sum_{i \in \mathcal{N}_n} \mathbf{h}_i^h) \ , \tag{12}$$

where $\mathcal{N}_n$ is the indices of neighboring nodes to the n-th node and MLPs^h denotes several sequential multi-layer perceptrons. Updating object node features follows the same in equation 12 by changing the superscript h to o. Intuitively, the graph convolutional layers exploit neighboring correlation from the topology of the mesh model and thus, can effectively model intra-graph dependencies.

Mutual Attention Layer. As shown in Figure 2, following one or several graph convolutional layers, we model hand-object interaction in the mutual attention layer. For each node from one graph, our mutual attention layer aims to aggregate features from the other graph via the attention mechanism. Specifically, for every node feature in the hand graph, we first use three 1D convolutional layers to extract the query, key, and value, and collect all queries, keys, and values as $\mathbf{Q}^h \in \mathbb{R}^{778\times H}$, $\mathbf{K}^h \in \mathbb{R}^{778\times F}$ and $\mathbf{V}^h \in \mathbb{R}^{778\times F}$ respectively, where each row of them is the query, key or value of a particular node. Similarly, we have the query, key and value for the object graph as $\mathbf{Q}^o \in \mathbb{R}^{1000\times F}$, $\mathbf{K}^o \in \mathbb{R}^{1000\times F}$ and $\mathbf{V}^o \in \mathbb{R}^{1000\times F}$ respectively. We then compute the object-to-hand attention between the queries from the hand graph and the keys from the object graph following [43] as:

$$\mathbf{A}^{o\to h} = \text{softmax}(\frac{\mathbf{Q}^h \mathbf{K}^{oT}}{\sqrt{F}}) \ , \tag{13}$$

where $\mathbf{A}^{o\to h} \in \mathbb{R}^{778\times 1000}$ is the object-to-hand attention map, with the i-th row denoting the expected contribution proportion of all object nodes to the i-th hand node. The softmax operation is performed along the second dimension. We can then aggregate object node features weighted by the object-to-hand attention as:

$$\mathbf{h}^{o\to h} = \mathbf{A}^{o\to h}\mathbf{V}^o \ , \tag{14}$$

where $\mathbf{V}^{o\to h} \in \mathbb{R}^{778\times F}$ is the aggregated features from the object graph. Similarly, we can compute the hand-to-object attention as:

$$\mathbf{A}^{h\to o} = \text{softmax}(\frac{\mathbf{Q}^o \mathbf{K}^{hT}}{\sqrt{F}}) \ , \tag{15}$$

where $\mathbf{A}^{h\to o} \in \mathbb{R}^{1000\times 778}$. And we can compute the hand-to-object feature as:

$$\mathbf{h}^{h\to o} = \mathbf{A}^{h\to o}\mathbf{V}^h \ , \tag{16}$$

where $\mathbf{V}^{h\to o} \in \mathbb{R}^{1000\times F}$. We finally fuse the aggregate feature with the original feature in each node as:

$$\tilde{\mathbf{h}}_n^h = f_v^h(\mathbf{h}_n'^{h} \oplus \mathbf{h}^{o\to h}) \ , \quad \tilde{\mathbf{h}}_n^o = f_v^o(\mathbf{h}_n'^{o} \oplus \mathbf{h}^{h\to o}) \ . \tag{17}$$

where $\tilde{\mathbf{h}}_n^h$ is the refined node feature as the output of each block, and $f_v^h(\cdot)$, $f_v^o(\cdot)$ are independent fusion units.

Intuitively, the mutual attention encodes the feature similarities between object and hand features. Since the local features are retrieved from the interpolation in the spatial domain, we expect vertices that are spatially close should be encoded with similar features due to the averaging effect in the interpolation. In this sense, the attention mechanism can effectively exploit interaction priors around contacting areas, as illustrated in Figure 1. In addition, since we evaluate the mutual attention between every pair of hand and object vertices, this process also allows for fine-grained hand-object interactions, which as will be shown in the experiment section, performs better than methods with only attention between sparse keypoints [11].

Refined Pose. The final output of the hand GCN is a mesh vertex offset $\Delta\mathbf{M}^h \in \mathbb{R}^{778\times 3}$, the refined hand mesh is then $\tilde{\mathbf{M}}^h = \hat{\mathbf{M}}^h + \Delta\mathbf{M}^h$. The object GCN outputs a 6D pose including the rotation and translation. In particular, inspired by [49] the object GCN extracts one object pose from every node of the object graph and the final pose $(\hat{\mathbf{R}}^o, \hat{\mathbf{T}}^o)$ is the average across all poses. We empirically found this gives a better pose than only estimating one pose from the entire graph.

3.4. Training Objectives

To effectively train the proposed model, we adopt a multi-tasking training objective. We first adopt an L1 loss to supervise rough and refined mesh predictions as:

$$\mathcal{L}_m = ||\hat{\mathbf{M}}^h - \mathbf{M}^h||_1 + ||\tilde{\mathbf{M}}^h - \mathbf{M}^h||_1 + ||\hat{\mathbf{M}}^o - \mathbf{M}^o||_1 \ , \tag{18}$$

where $\mathbf{M}^h$ and $\mathbf{M}^o$ denote the ground truth mesh for the hand and the object respectively. Following [40], we further refine the mesh quality by imposing the edge loss $\mathcal{L}_e$ and the normal loss $\mathcal{L}_n$ to penalize flying vertices and irregular surfaces as:

$$\begin{aligned}\mathcal{L}_e = &\sum_i |||\hat{\mathbf{e}}_i^h| - |\mathbf{e}_i^h|||_1 + |||\tilde{\mathbf{e}}_i^h| - |\mathbf{e}_i^h|||_1 \\ &+ \sum_j |||\hat{\mathbf{e}}_j^o| - |\mathbf{e}_j^o|||_1 \ ,\end{aligned} \tag{19}$$

$$\begin{aligned}\mathcal{L}_n = &\sum_i \|\langle\hat{\mathbf{e}}_i^h, \mathbf{n}_i^h\rangle\|_1 + \|\langle\tilde{\mathbf{e}}_i^h, \mathbf{n}_i^h\rangle\|_1 \\ &+ \sum_j \|\langle\hat{\mathbf{e}}_j^o, \mathbf{n}_i^o\rangle\|_1 \ ,\end{aligned} \tag{20}$$

where $\hat{\mathbf{e}}_i^h$ and $\tilde{\mathbf{e}}_i^h$ denote the i-th mesh edge vector of the rough hand mesh and the refined hand mesh respectively. $\hat{\mathbf{e}}_j^o$ is the j-th mesh edge of the rough object mesh. $|\cdot|$ represents the length of the edge. $\mathbf{e}_i^h$ and $\mathbf{n}_i^h$ are the ground truth edge vector and the normal of the corresponding edge.

To supervise the refined object pose, we adopt an L2 loss on the estimated rotation quaternion and translation as:

$$\mathcal{L}_o = \|\hat{\mathbf{R}}^o - \mathbf{R}^o\|_2 + \|\hat{\mathbf{T}}^o - \mathbf{T}^o\|_2 \ , \tag{21}$$

where $\mathbf{R}^o$ and $\mathbf{T}^o$ denote the ground truth object pose.

To supervise the hand joint estimation, we adopt a joint loss $\mathcal{L}_j$ between the ground truth joints $\mathbf{J}^h$ and the predicted joints from the joint estimator $\hat{\mathbf{J}}^h$, as well as the regressed joints from the predicted hand mesh, *i.e.*, we use the joint regression matrix $\mathbf{G} \in \mathbb{R}^{21\times 778}$ defined in the MANO [35] model to obtain the joint locations, then calculate the joint loss as:

$$\mathcal{L}_j = \|\mathbf{G}\hat{\mathbf{M}}^h - \mathbf{J}^h\|_1 + \|\mathbf{G}\tilde{\mathbf{M}}^h - \mathbf{J}^h||_1 + ||\hat{\mathbf{J}}^h - \mathbf{J}^h\|_1 \ , \tag{22}$$

Besides, we also guide the prediction of the object silhouette using a cross-entropy loss as:

$$\mathcal{L}_s = -\sum_{i=1}^{H\times W} y_i \log s_i \ , \tag{23}$$

where s_i is the i-th pixel in the predicted object silhouette $\hat{\mathbf{S}}^o$ and y_i is the ground truth at the same pixel.

Finally, inspired by [40], we impose a finger rendering loss $\mathcal{L}_f$ to supervise the alignment of fingers in the image space. We adopt a differentiable renderer $f_r(\cdot)$ [19] to render the refined hand mesh as well as the ground truth hand mesh using the given camera intrinsic $\mathbf{K}$. We then classify the type of finger for each vertex based on the maximum blending weights defined in MANO and provide a distinct color texture for each finger. The loss can be formally written as the L1 loss between the two rendered images:

$$\mathcal{L}_f = ||f_r(\tilde{\mathbf{M}}^h) - f_r(\mathbf{M}^h)||_1 \ . \tag{24}$$

The overall training loss is a weighted sum of all individual loss functions, defined as:

$$\mathcal{L} = \lambda_m\mathcal{L}_m + \lambda_e\mathcal{L}_e + \lambda_n\mathcal{L}_n + \lambda_o\mathcal{L}_o + \lambda_j\mathcal{L}_j + \lambda_s\mathcal{L}_s + \lambda_f\mathcal{L}_f \ , \tag{25}$$

where we empirically set $\lambda_m = \lambda_e = \lambda_n = \lambda_j = 1$, $\lambda_o = 10$, $\lambda_s = \lambda_f = 100$ so that all loss terms are roughly in the same scale.

4. Experiment Results

In this section, we first introduce the datasets for the training (Section 4.1) and define the evaluation metrics on each dataset (Section 4.2). We then provide the implementation details (Section 4.3) for our experiments, and compare the results with state-of-the-art methods both quantitatively and qualitatively (Section 4.4). Finally, we perform an ablation study to investigate the effects of the mutual attention layer and demonstrate the learned interaction from the estimated attention maps (Section 4.5).

4.1. Training Data

Datasets. We evaluate our methods on two large-scale hand-object benchmarks: HO3D v2 [10] and DexYCB [3], each containing 66K and 589K images of human interacting YCB [2] objects. We train the model separately on each dataset based on the official train-test split, in particular, we use the default S0 split for the DexYCB testing set. For a fair comparison in the DexYCB dataset, we follow [41] to select input frames where the hand and object are both visible with an in-between distance less than $1cm$ to ensure a physical contact can be established. We crop the input images in both datasets using the provided hand-object bounding box following [26] and resize all images into 256×256 pixels.

Data Augmentation. Considering that the HO3D dataset is relatively small-scale, to facilitate the training we perform two types of augmentation, *i.e.* view synthesis to resolve occlusion ambiguity and grasp synthesis to increase the diversity of hand-object interaction. For view synthesis, we randomly rotate the camera relative to the object center. We additionally generate 5K distinct hand manipulating YCB objects scenes using the GrabNet [39] to perform grasp synthesis. We manually verify that the synthesized poses are not seen in the testing set and are physically plausible by empirically filtering out samples with the contact loss and penetration loss [14] greater than the threshold $\lambda_c = 0.012$ and $\lambda_p = 0.1$ respectively.

We use Pytorch3D [34] to render the synthetic image counterparts from the augmented poses. We adopt the HTML [32] model for realistic hand skin colors and textures, and superimpose the rendered hand-object images on top of randomly sampled backgrounds from the indoor-scene dataset [33]. To reduce the domain gap, we further perform photometric augmentation on rendered images, including random contrast and brightness transforms uniformly sampled from [0.5, 1.5]. In addition, we add random Gaussian blurs on synthetic images with σ uniformly sampled from [0.1, 1]. The three types of inputs, *i.e.* real, view-synthetic, and grasp-synthetic images are distributed in 0.45: 0.45: 0.1 in a training batch for the HO3D training set. Compared with [26], we introduce the data augmentation mostly with a simple view transformation, however, our model achieves better performance with less augmented grasping data as shown in Section 4.4.

4.2. Evaluation Metrics

In order to consistently compare the results with state-of-the-art methods, we adopt the evaluation metrics on each benchmark dataset that are majorly reported by related works. We refer readers in the supplementary materials for additional metrics reported in some works [26, 11].

HO3D Metrics. For the hand pose evaluation, we follow

Table 1. **Quantitative comparison on the HO3D v2 testing set.** Best results are highlighted in **bold** and unavailable results are marked with "-". Additional object metrics for [26, 11] are compared and included in the supplementary material.

	Hand				Object		Interaction	
Methods	MJE $(cm)\downarrow$	AUC-MJE $\uparrow$	MME $(cm)\downarrow$	AUC-MME $\uparrow$	MME $(cm)\downarrow$	ADD-S $(cm)\downarrow$	PD $(mm)\downarrow$	CP $(\%)\uparrow$
Hasson *et al.* [12]	3.69	0.469	1.14	0.773	8.7	2.9	-	-
Hasson *et al.* [13]	2.68	0.510	1.20	0.761	8.0	3.8	1.5	77.5
Keypoint Trans. [11]	2.57	0.532	-	-	-	-	-	-
Artiboost [26]	2.53	0.532	1.09	0.782	-	-	-	-
Ours	**2.38**	**0.560**	**1.06**	**0.789**	**5.7**	**2.3**	**1.3**	**85.6**

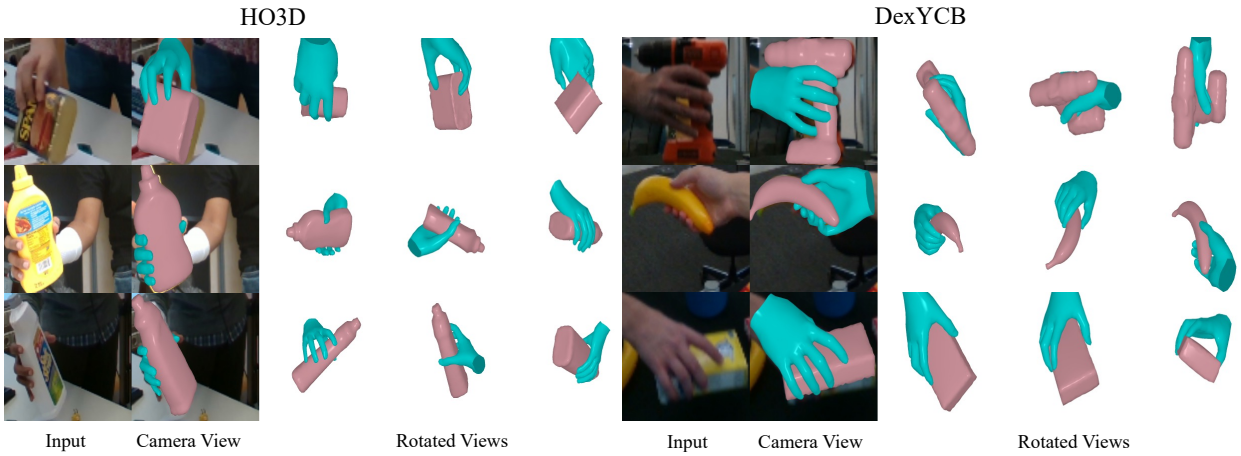

Figure 3. **Qualitative results on the HO3D and DexYCB testing sets.** The predicted hand and object pose well align with input images in the camera view. Rotated view images show the grasping configuration is physically plausible and valid contacts can be established.

the official evaluation metrics in the HO3D v2 CodaLab Challenge. Specifically, we report the mean joint error (MJE) [51] and mean mesh error (MME) [52] as the average Euclidean distance between predicted and ground truth joints/meshes after root joint and global scale alignment. In addition, we report the AUC of the percentage of correct keypoints (PCK) curve in an interval from $0cm$ to $5cm$ with 100 equally spaced thresholds. For the object pose evaluation, we follow [40] to report the MME for the object mesh and the standard pose estimation average closest point distance (ADD-S) [45]. Finally, we report the mean penetration depth (PD) [1] and the contact percentage (CP) [18] between the hand and object meshes to evaluate the hand-object interaction.

DexYCB Metrics. We adopt the evaluation metrics used in recent works [26, 41] for the DexYCB dataset. Specifically, for the hand pose, we also report the mean joint error (MJE). For the object pose, We report the mean corner error (MCE) as the distance of the bounding box corners positions between the predicted and ground truth object meshes. Finally, we report the mean penetration depth to evaluate the hand-object collision as well.

4.3. Implementation Details

We train the network using the Adam [21] optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.999$ on a single NVIDIA RTX 3090 GPU. We set the batch size as 24 and train the model in 25 epochs. The initial learning rate is set as $1e^{-4}$ and decayed by 0.1 after every 10 epochs. Our model achieves an inference speed of 34 FPS on an NVIDIA RTX 3090 GPU, which can be severed for future real-time applications. We refer the readers in the supplementary material for the detailed network architecture for each module.

4.4. Results

Comparison with State-of-the-Arts. In Table 1, we evaluate our model on the HO3D v2 testing set and compare the results with state-of-the-art methods [12, 13, 26, 11]. All results under hand metrics are collected from the official HO3D v2 CodaLab Challenge outcomes. From the table, we observe that our method achieves superior results across all hand, object, and interaction metrics. In particular, our method not only produces more accurate hand and object poses, but also generates physically realistic hand-object grasping in higher quality as we observe a lower penetration and a higher contact rate than [13]. Meanwhile, our method leverages an efficient feed-forward pipeline from a single image input and does not require computationally-expensive optical flows as temporal clues [12] or iterative optimization process [13]. Furthermore, our method does not rely on sophisticated contact losses as in [12, 13], showing the superiority of our method in modeling hand-object interaction. Compared to [26], our model is trained with

remarkably less augmented data, yet achieves improved results without introducing much complexity. Finally, thanks to the dense mutual attention, our method improves the performance by a large margin than the sparse keypoints-based method [11].

To further justify the effectiveness of the model, we also evaluate our model on the recently released DexYCB dataset and compare the results with [14, 13, 41] in Table 2. Note that while [13] has the same setting with us, [14, 41] do not assume known object CAD models, therefore tackle a more challenging task and can perform worse in estimating accurate object meshes. Hence we only compare with them in the hand metric. The results show that our method consistently outperforms baseline methods in all comparable metrics.

Table 2. **Quantitative comparison on the DexYCB testing set.** Best results are highlighted in **bold** and non-comparable results are marked with "-".

Methods	Hand MJE (cm)↓	Object MCE (cm)↓	Interaction PD (mm)↓
Hasson [14]	1.76	-	-
Hasson [13]	1.88	5.25	0.79
Tze *et al.* [41]	1.53	-	-
Ours	**1.27**	**3.26**	**0.67**

Qualitative Results. We show qualitative results on the HO3D and DexYCB testing sets in Figure 3. We render the estimated hand and object meshes under the camera view along with three randomly rotated views. It can be seen that our method produces accurate hand-object poses that align well with the given image input, and the estimated poses satisfy physical constraints, *i.e.* a valid grasping can be observed. More results can be found in the supplementary material.

4.5. Ablation Study

To further justify the effectiveness of the proposed mutual attention mechanism, we further perform an ablation study. We first visualize the attention maps in Figure 1 (second column). For object-to-hand attention, we select hand vertices whose minimal distance to the object is less than $1cm$, and visualize the corresponding average attention between all object vertices. The hand-to-object attention is visualized in a similar way for contacting object vertices. The Figure shows that contacting areas contain higher attention values (in red) than non-contacting areas (in blue), which illustrates that the mutual attention mechanism can effectively model hand-object interaction correlation to facilitate pose refinement by exploiting contact priors.

We further construct variant baselines with alternative utilization of hand-object interaction priors and compare the results in Table 3. With the GCN refinement (w/o attention), the network can effectively improve hand and object pose estimation from the rough stage (w/o GCN) by a large margin thanks to the information from intra-graph dependencies.

Table 3. **Effects of the mutual attention layer**. Best results are highlighted in **bold**.

Methods	Hand MJE (cm)↓	Hand MME (cm)↓	Object MME (cm)↓	Interaction PD (mm)↓
w/out GCN	2.84	1.29	13.4	3.6
w/out attention	2.66	1.20	7.7	2.9
all edge	2.79	1.34	8.9	4.3
w/o hand-to-object	2.46	1.10	6.2	1.4
w/o object-to-hand	2.50	1.12	5.8	1.3
mutual attention	**2.38**	**1.09**	**5.7**	**1.3**

A naive baseline method for hand-object feature aggregation is to have the hand and object graphs fully connected (all edge), analog to [8]. However, despite that this approach works well in sparse graphs, *e.g.* including only hand joints and object bounding box corners as graph nodes, it is hard to extend the approach to a dense mesh graph. We hypothesize that the fully connected graph significantly increases the model complexity, thus making the network hard to train and converge. In addition, equally aggregating noisy features without adaptive weighting can also mislead the network in prediction.

Finally, we examine the variants where only one direction of attention is used. When we allow only hand features aggregating to object nodes (w/o object-to-hand), we observe an increase in performance in hand metrics, however, the object pose estimation is impaired compared to the converse variant (w/o hand-to-object). When the full mutual attention is included (mutual attention), we observe the best performing result. We therefore conclude that mutual attention benefits both hand and object pose estimation.

5. Discussion

Limitation. Our work relied on the lixel representation for hand and object meshes estimation, since the representation quantizes the image space, there is no valid correspondence to vertices that are outside the camera's field of view. Hence our method can not properly handle scenes where the hand or object is only partially included in the image. Moreover, we have only considered objects from a subset of classes where well-defined CAD models can be provided, future works should consider the interaction between hands with a more diverse set of interacting objects.

Conclusion. In this paper, we proposed a novel dense mutual attention mechanism to effectively model fine-grained hand-object interaction. To exploit both intra-class and inter-class dependencies, we integrate mutual attention in the graph convolutional networks to refine the initially-estimated hand-object pose. Our method surpasses state-of-the-art methods when evaluated on widely-used benchmark datasets, demonstrating the effectiveness of the proposed techniques.

References

[1] 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 *European Conference on Computer Vision*, pages 361–378. Springer, 2020.

[2] Berk Calli, Arjun Singh, Aaron Walsman, Siddhartha Srinivasa, Pieter Abbeel, and Aaron M Dollar. The ycb object and model set: Towards common benchmarks for manipulation research. In *2015 international conference on advanced robotics (ICAR)*, pages 510–517. IEEE, 2015.

[3] Yu-Wei Chao, Wei Yang, Yu Xiang, Pavlo Molchanov, Ankur Handa, Jonathan Tremblay, Yashraj S Narang, Karl Van Wyk, Umar Iqbal, Stan Birchfield, et al. Dexycb: A benchmark for capturing hand grasping of objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9044–9053, 2021.

[4] Olivier Chapelle and Mingrui Wu. Gradient descent optimization of smoothed information retrieval metrics. *Information retrieval*, 13(3):216–235, 2010.

[5] Liangjian Chen, Shih-Yao Lin, Yusheng Xie, Yen-Yu Lin, and Xiaohui Xie. Temporal-aware self-supervised learning for 3d hand pose and mesh estimation in videos. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 1050–1059, January 2021.

[6] Chiho Choi, Sang Ho Yoon, Chin-Ning Chen, and Karthik Ramani. Robust hand pose estimation during the interaction with an unknown object. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3123–3132, 2017.

[7] Xiao Chu, Wei Yang, Wanli Ouyang, Cheng Ma, Alan L Yuille, and Xiaogang Wang. Multi-context attention for human pose estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1831–1840, 2017.

[8] Bardia Doosti, Shujon Naha, Majid Mirbagheri, and David J Crandall. Hope-net: A graph-based model for hand-object pose estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6608–6617, 2020.

[9] Liuhao 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10833–10842, 2019.

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

[11] Shreyas Hampali, Sayan Deb Sarkar, Mahdi Rad, and Vincent Lepetit. Keypoint transformer: Solving joint identification in challenging hands and object interactions for accurate 3d pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11090–11100, 2022.

[12] 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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 571–580, 2020.

[13] Yana Hasson, Gül Varol, Cordelia Schmid, and Ivan Laptev. Towards unconstrained joint hand-object reconstruction from rgb videos. In *2021 International Conference on 3D Vision (3DV)*, pages 659–668. IEEE, 2021.

[14] 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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11807–11816, 2019.

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[16] Lin Huang, Jianchao Tan, Jingjing Meng, Ji Liu, and Junsong Yuan. Hot-net: Non-autoregressive transformer for 3d hand-object pose estimation. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 3136–3145, 2020.

[17] Umar Iqbal, Pavlo Molchanov, Thomas Breuel Juergen Gall, and Jan Kautz. Hand pose estimation via latent 2.5 d heatmap regression. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 118–134, 2018.

[18] Korrawe Karunratanakul, Jinlong Yang, Yan Zhang, Michael J Black, Krikamol Muandet, and Siyu Tang. Grasping field: Learning implicit representations for human grasps. In *2020 International Conference on 3D Vision (3DV)*, pages 333–344. IEEE, 2020.

[19] Hiroharu Kato, Yoshitaka Ushiku, and Tatsuya Harada. Neural 3d mesh renderer. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3907–3916, 2018.

[20] Leyla Khaleghi, Alireza Sepas Moghaddam, Joshua Marshall, and Ali Etemad. Multi-view video-based 3d hand pose estimation. *arXiv preprint arXiv:2109.11747*, 2021.

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

[22] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[23] Muhammed Kocabas, Chun-Hao P Huang, Otmar Hilliges, and Michael J Black. Pare: Part attention regressor for 3d human body estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11127–11137, 2021.

[24] Hema Swetha Koppula and Ashutosh Saxena. Anticipating human activities for reactive robotic response. In *IROS*, page 2071. Tokyo, 2013.

[25] Dominik Kulon, Riza Alp Guler, Iasonas Kokkinos, Michael M Bronstein, and Stefanos Zafeiriou. Weakly-supervised mesh-convolutional hand reconstruction in the

wild. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4990–5000, 2020.

[26] Kailin Li, Lixin Yang, Xinyu Zhan, Jun Lv, Wenqiang Xu, Jiefeng Li, and Cewu Lu. Artiboost: Boosting articulated 3d hand-object pose estimation via online exploration and synthesis. *arXiv preprint arXiv:2109.05488*, 2021.

[27] Yi Li, Gu Wang, Xiangyang Ji, Yu Xiang, and Dieter Fox. Deepim: Deep iterative matching for 6d pose estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 683–698, 2018.

[28] Shaowei Liu, Hanwen Jiang, Jiarui Xu, Sifei Liu, and Xiaolong Wang. Semi-supervised 3d hand-object poses estimation with interactions in time. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14687–14697, 2021.

[29] Gyeongsik Moon and Kyoung Mu Lee. I2l-meshnet: Image-to-lixel prediction network for accurate 3d human pose and mesh estimation from a single rgb image. In *European Conference on Computer Vision*, pages 752–768. Springer, 2020.

[30] JoonKyu Park, Yeonguk Oh, Gyeongsik Moon, Hongsuk Choi, and Kyoung Mu Lee. Handoccnet: Occlusion-robust 3d hand mesh estimation network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1496–1505, 2022.

[31] Sida Peng, Yuan Liu, Qixing Huang, Xiaowei Zhou, and Hujun Bao. Pvnet: Pixel-wise voting network for 6dof pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4561–4570, 2019.

[32] Neng Qian, Jiayi Wang, Franziska Mueller, Florian Bernard, Vladislav Golyanik, and Christian Theobalt. Html: A parametric hand texture model for 3d hand reconstruction and personalization. In *European Conference on Computer Vision*, pages 54–71. Springer, 2020.

[33] Ariadna Quattoni and Antonio Torralba. Recognizing indoor scenes. In *2009 IEEE conference on computer vision and pattern recognition*, pages 413–420. IEEE, 2009.

[34] Nikhila Ravi, Jeremy Reizenstein, David Novotny, Taylor Gordon, Wan-Yen Lo, Justin Johnson, and Georgia Gkioxari. Accelerating 3d deep learning with pytorch3d. *arXiv preprint arXiv:2007.08501*, 2020.

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

[36] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[37] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

[38] Sebastian Starke, He Zhang, Taku Komura, and Jun Saito. Neural state machine for character-scene interactions. *ACM Trans. Graph.*, 38(6):209–1, 2019.

[39] Omid Taheri, Nima Ghorbani, Michael J Black, and Dimitrios Tzionas. Grab: A dataset of whole-body human grasping of objects. In *European conference on computer vision*, pages 581–600. Springer, 2020.

[40] Xiao Tang, Tianyu Wang, and Chi-Wing Fu. Towards accurate alignment in real-time 3d hand-mesh reconstruction. In *International Conference on Computer Vision (ICCV)*, pages 11698–11707, 2021.

[41] Tze Ho Elden Tse, Kwang In Kim, Ales Leonardis, and Hyung Jin Chang. Collaborative learning for hand and object reconstruction with attention-guided graph convolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1664–1674, 2022.

[42] Sébastien Valette, Jean Marc Chassery, and Rémy Prost. Generic remeshing of 3d triangular meshes with metric-dependent discrete voronoi diagrams. *IEEE Transactions on Visualization and Computer Graphics*, 14(2):369–381, 2008.

[43] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[44] Gu Wang, Fabian Manhardt, Federico Tombari, and Xiangyang Ji. Gdr-net: Geometry-guided direct regression network for monocular 6d object pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16611–16621, 2021.

[45] Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. *arXiv preprint arXiv:1711.00199*, 2017.

[46] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826*, 2018.

[47] John Yang, Yash Bhalgat, Simyung Chang, Fatih Porikli, and Nojun Kwak. Dynamic iterative refinement for efficient 3d hand pose estimation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1869–1879, 2022.

[48] Lixin Yang, Xinyu Zhan, Kailin Li, Wenqiang Xu, Jiefeng Li, and Cewu Lu. Cpf: Learning a contact potential field to model the hand-object interaction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11097–11106, 2021.

[49] Pengshuai Yin, Jiayong Ye, Guoshen Lin, and Qingyao Wu. Graph neural network for 6d object pose estimation. *Knowledge-Based Systems*, 218:106839, 2021.

[50] Christian Zimmermann, Max Argus, and Thomas Brox. Contrastive representation learning for hand shape estimation. In *DAGM German Conference on Pattern Recognition*, pages 250–264. Springer, 2021.

[51] Christian Zimmermann and Thomas Brox. Learning to estimate 3d hand pose from single rgb images. In *Proceedings of the IEEE international conference on computer vision*, pages 4903–4911, 2017.

[52] Christian Zimmermann, Duygu Ceylan, Jimei Yang, Bryan Russell, Max Argus, and Thomas Brox. Freihand: A dataset for markerless capture of hand pose and shape from single

rgb images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 813–822, 2019.