Collaborative learning for hand and object reconstruction with attention-guided graph convolution

Tse, Tze Ho Elden; Kim, Kwang In; Leonardis, Ales; Chang, Hyung Jin

DOI:
10.1109/CVPR52688.2022.00171

Collaborative Learning for Hand and Object Reconstruction with Attention-guided Graph Convolution

Tze Ho Elden Tse[1] Kwang In Kim[2] Aleš Leonardis[1] Hyung Jin Chang[1]
[1]University of Birmingham [2]UNIST
txt994@student.bham.ac.uk, kimki@unist.ac.kr, {a.leonardis, h.j.chang}@bham.ac.uk

Abstract

Estimating the pose and shape of hands and objects under interaction finds numerous applications including augmented and virtual reality. Existing approaches for hand and object reconstruction require explicitly defined physical constraints and known objects, which limits its application domains. Our algorithm is agnostic to object models, and it learns the physical rules governing hand-object interaction. This requires automatically inferring the shapes and physical interaction of hands and (potentially unknown) objects. We seek to approach this challenging problem by proposing a collaborative learning strategy where two-branches of deep networks are learning from each other. Specifically, we transfer hand mesh information to the object branch and vice versa for the hand branch. The resulting optimisation (training) problem can be unstable, and we address this via two strategies: (i) attention-guided graph convolution which helps identify and focus on mutual occlusion and (ii) unsupervised associative loss which facilitates the transfer of information between the branches. Experiments using four widely-used benchmarks show that our framework achieves beyond state-of-the-art accuracy in 3D pose estimation, as well as recovers dense 3D hand and object shapes. Each technical component above contributes meaningfully in the ablation study.

1. Introduction

Understanding human hand and object interaction is fundamental for meaningful interpretation of human action and behaviour [65, 72]. With the advent of deep learning and RGB-D sensors, pose estimation of isolated hands has made significant progress, *e.g.*, depth-based [12,69,74,81,82] and RGB-based [51,60,63,77,85] methods. However, despite a strong link to real applications such as augmented and virtual reality [32,52,71], joint reconstruction of hand and object [33, 35] has received relatively less attention. In this paper, we focus on the problem of hand and object reconstruction from a single RGB image (see Fig. 1).

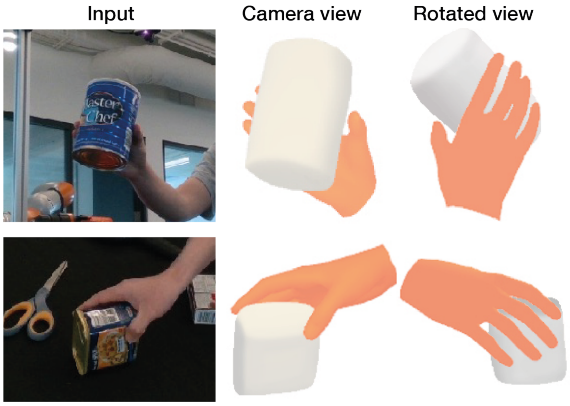

Figure 1. We propose a collaborative learning framework which allows sharing of mesh information across hand and object branches iteratively. Our model jointly reconstructs hand and object meshes from a monocular RGB image.

Joint hand and object pose estimation is a challenging problem. First, while self-occlusion in hand is a well-known problem [56, 80], when interacting with objects, hands (and objects) exhibit even greater occlusion from almost any point of view mutually [53]. Secondly, first-person-view (*e.g.*, *FHB* [24] dataset) often exhibits large degree of erratic camera motion. Recent works [23, 42, 65] have been able to tackle some major challenges in joint hand-object pose estimations in colour input. However, in the absence of physical constraints, and with sparse keypoint detection, they often lead to erroneous pose estimation or mesh reconstructions (*e.g.* hands penetrating objects).

To fundamentally understand hand-object interactions, it is essential to fully recover 3D information, and accordingly, there has been significant improvements towards hand mesh estimations from single RGB image [3, 4, 10, 19, 25, 41, 50, 83, 84, 86]. Hasson *et al.* [35] further proposed attraction and repulsion loss terms to generate physically plausible reconstructions. Recent optimisation-based approaches [14, 34] that rely on these contact terms are limited to scenarios where hand and object are already in contact. However, the ability to reason pre-grasp stages are equally important as it allows robots to infer human in-

tents [48] and learn manipulation skills from humans [45]. Therefore, we propose a strategy that is not restricted by these contact terms and is able to learn the context of actual as well as near physical contact.

Our novel collaborative learning framework allows hand and object branches to boost each other in a progressive and iterative fashion. There are two motivations for this strategy: 1) estimating the pose of interacting hands and objects is a highly-correlated task and 2) mutual occlusions can be tackled by simultaneously sharing mesh information. This is supported by the fact that the image encoder struggles to extract useful features under mutual occlusion, and therefore capturing object mesh information would compensate this limitation for hand reconstruction (the same in object branch). Previous attempts in this context share information across branches via simple branch stacking [79] where communication bottleneck exists: We empirically observed that performance gain across network inference iterations are limited in this approach. We explicitly address this by a new unsupervised associative loss facilitating the information transfer. Further, to address frequently occurring occlusions in hand-object interaction scenarios, we propose an attention-guided graph convolution that can be trained in an unsupervised manner. Our graph convolution demonstrates the ability to improve mesh quality as well as correct hand and object poses.

Our contributions are the following:

1. We propose an end-to-end trainable collaborative learning strategy for hand-object reconstruction from a single RGB image.
2. We design an attention-guided graph convolution to capture mesh information dynamically.
3. We introduce an unsupervised training strategy for effective feature transfer between hand-object branches.
4. We demonstrate that our model achieves highly physically plausible results without contact terms.

We evaluate our method on four hand-object datasets *i.e*. *FHB* [24], *ObMan* [35], *HO-3D* [31] and *DexYCB* [17] and demonstrate that our method significantly outperform state-of-the-art approaches.

2. Related works

Our work tackles the problem of hand and object reconstruction from a single RGB image. We first review the literature on *Hand-Object Reconstruction*. Then, we focus on the line of work that leverages *Graph Convolutional Neural Networks* on hand-related tasks. Finally, we provide a brief review on *Collaborative Learning* despite its weak link in the literature.

Hand-object reconstruction. Joint reconstruction of hands and objects has been receiving increasing attention [14, 33–35]. Hasson *et al*. [35] leverages a differentiable MANO network layer enabling end-to-end learning of hand shape estimation and incorporates contact losses which encourages contact surfaces and penalises penetrations between hand and object. Hasson *et al*. [33] assumes known object models and leverages photometric consistency as self-supervision on the unannotated intermediate frames to improve hand and object reconstructions. Karunratanakul *et al*. [38] proposes an implicit representation for hand in the form of sign distance fields. Recent works mostly adopt optimisation-based procedures to jointly fit hand-object meshes [14, 34, 78]. In this paper, we propose a learning-based strategy where immediate features are shared across hand-object branches and are able to produce physically plausible interactions without any contact terms.

Graph convolution-based methods. As skeleton can be represented in a form of graph, graph convolution naturally attracts much attention in hand pose estimation. Graph convolutional neural networks (GCN) can be split into spectral-[11, 21, 40] and spatial-based methods [27, 49, 76]. For spectral-based application, [19, 25] adopt the Chebyshev spectral graph convolution [21] to compute hand mesh. Cai *et al*. [13] leverages GCN [40] and apply on the sequence of skeletons as a spatial-temporal graph to exploit the spatial and temporal consistencies for pose estimation. Doosti *et al*. [23] proposes a lightweight graph convolutional network which jointly estimates hand and object poses. Kulon *et al*. [41] proposes spiral filters to recover hand mesh directly from autoencoder. They demonstrate that spatial mesh convolutions outperform spectral methods and SMPL-based models [44,57] for hand reconstruction. In contrast, our proposed attention-guided graph convolution is able to take dynamic graph input and does not assume a fixed neighbourhood for feature aggregation.

Collaborative learning. There has been a lot of literatures concerning learning multiple tasks simultaneously. They span across the spectrum of multi-task learning [7, 8, 15], domain adaptation [46, 47], distributed learning [6, 22, 70] and collaborative learning [9, 37, 54, 61]. Collaborative learning refers to making learning more efficient through sharing of information. Blum *et al*. [9] proposes a collaborative PAC (*probably approximately correct*) learning model which was built upon Valiant *et al*. [66] and [18, 54] are the follow-up works. Song *et al*. [61] introduces one form of collaborative learning framework in which multiple classifier heads of the same network are simultaneously trained on the same training data to improve generalisation and robustness without extra inference cost. There are two major mechanisms under his framework: 1) Same training datasets for multiple views from different classifiers improves generalisation and 2) Intermediate-level representation sharing. Yang *et al*. [79] exploits joint-aware features for gesture recognition and 3D hand pose estimation. Their mechanism focuses on intermediate-level representation sharing itera-

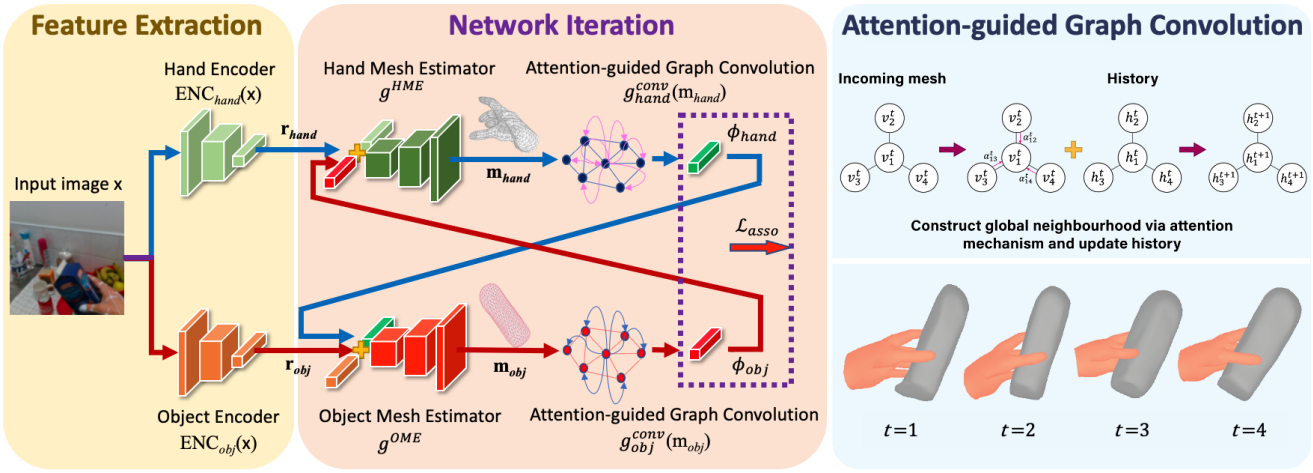

Figure 2. A schematic illustration of our framework. It takes an input image $\mathbf{x}$, which goes through two separate ResNet-18 [36] encoders, $\text{ENC}_{hand}(\mathbf{x})$ and $\text{ENC}_{obj}(\mathbf{x})$ to produce hand and object features, $\mathbf{r}_{hand}$ and $\mathbf{r}_{obj}$, respectively. Hand mesh estimator g^{HME} takes $\mathbf{r}_{hand}$ and output hand mesh $\mathbf{m}_{hand}$ which is then pass to graph convolution module g^{conv}_{hand} and output ϕ_{hand}. Object mesh estimator takes both $\mathbf{r}_{obj}$ and ϕ_{hand} to output object mesh $\mathbf{m}_{obj}$. Similarly, graph convolution module g^{conv}_{obj} takes object mesh $\mathbf{m}_{obj}$ and output ϕ_{obj} which is then combine with hand features $\mathbf{r}_{hand}$ and goes into the hand mesh estimator g^{HME}. An unsupervised associative loss is used to supervise the feature transfer process under network iterations, *i.e.* ϕ_{hand} and ϕ_{obj}. We have included an example on the bottom right corner which demonstrates the effect of our attention-guided graph convolution for iteration t.

tively across multiple tasks. In this paper, we improve on [79] with an attention-guided graph convolution and an unsupervised associative loss to guide the intermediate-level representation sharing process. Also, our proposed graph convolution is based on a multi-head attention mechanism which possesses the spirit of [61] to improve generalisation with multiple views on the same dataset.

3. Collaborative estimation of hand and object meshes

Our training pipeline, as shown in Fig. 2, takes an input RGB image $\mathbf{x} \in \mathbb{R}^{256\times256}$ and involves 4 steps for one iteration: 1) Reconstruct hand mesh using the parametric MANO model [57]; 2) Extract hand features from hand mesh guided by our associative loss; 3) Reconstruct object mesh by fusing object encoder features and extracted hand features from the previous step; and 4) Extract object features from object mesh. Our architecture is split into hand and object branches. Each branch has a ResNet-18 [36] encoder pre-trained on ImageNet [58]: $\text{ENC}_{hand}(\mathbf{x})$ and $\text{ENC}_{obj}(\mathbf{x})$.

The key motivation for our approach is to leverage the implicit hand-object relationship: We target the problem of mutual occlusion in hand-object interactions by simultaneously sharing 3D reconstructions under our collaborative learning framework. However, naïvely connecting network branches tended to accumulated errors, leading to highly unstable training. Therefore, we propose an attention-guided graph convolution to capture 3D reconstructions dynamically. In addition, by following the notion that hand shape deforms according to object shape, we propose an unsupervised associative loss to improve the feature transfer process from hand to object, and vice versa. Our networks are trained in an end-to-end manner. Alg. 1 summarises the training process.

3.1. Hand mesh estimator g^{HME}

We adopted the differential MANO [57] model from [35]. It maps pose ($\boldsymbol{\theta} \in \mathbb{R}^{51}$) and shape ($\boldsymbol{\beta} \in \mathbb{R}^{10}$) parameters to a mesh with $N = 778$ vertices. Pose parameters ($\boldsymbol{\theta}$) consists of 45 DoF (*i.e.* 3 DoF for each of the 15 finger joints) plus 6 DoF for rotation and translation of the wrist joint. Shape parameters ($\boldsymbol{\beta}$) are fixed for a given person. A kinematic tree is formed with the 15 joints and the wrist joint as the first parent node. Joint locations can be obtained using the kinematic tree with global rotation based on $\boldsymbol{\theta}$.

Given the 512-dimensional hand feature vector $\mathbf{r}_{hand}$, we use a fully connected layer to regress $\boldsymbol{\theta}$ and $\boldsymbol{\beta}$. The original MANO model uses 6-dimensional PCA (principal component analysis) subspace of $\boldsymbol{\theta}$ for computational efficiency. However, we empirically observed that full 45-dimensional pose space better captures a variety of hand poses especially over sequential datasets. A hand mesh can be defined as $\mathbf{m}_{hand} = (\mathbf{v}_{hand}, \mathbf{f}_{hand})$, where $\mathbf{v}_{hand} \in \mathbb{R}^{778\times3}$ refers to a set of vertices in the mesh and $\mathbf{f}_{hand} \in \mathbb{R}^{1538\times3}$ refers to a close set of edges (*i.e.* a triangle face has 3 edges). The mesh faces $\mathbf{f}_{hand}$ is provided by MANO [57].

Hand reconstruction loss $\mathcal{L}_{hand}$. We directly optimise root-relative 3D positions by minimising their L2 distance to the corresponding ground-truth vertex positions $\mathbf{v}^*_{hand}$:

$$\mathcal{L}_V(\mathbf{v}_{hand}) = \|\mathbf{v}_{hand} - \mathbf{v}^*_{hand}\|_2^2 . \quad (1)$$

When ground truth vertex positions are not available, we supervise on 3D joint locations $\mathbf{J} \in \mathbb{R}^{n\times 3}$ where n refers to the number of joints. The 3D joint loss is defined as:

$$\mathcal{L}_J(\mathbf{J}) = \|\mathbf{J} - \mathbf{J}^*\|_2^2 , \quad (2)$$

where $\mathbf{J}^*$ refers to ground truth joint positions. The resulting loss is defined as: $\mathcal{L}_{hand} = \mathcal{L}_V + \mathcal{L}_J$. We do not adopt hand shape regularisation as in [35] as we empirically observed that our iterative process already prevents extreme mesh deformation.

3.2. Object mesh estimator g^{OME}

Given the 512-dimensional object feature vector $\mathbf{r}_{obj}$, we adopt AtlasNet [29] from [35] to estimate object mesh $\mathbf{m}_{obj} = (\mathbf{v}_{obj}, \mathbf{f}_{obj})$, *i.e.* $\mathbf{v}_{obj} \in \mathbb{R}^{642\times 3}$ refers to object vertices and $\mathbf{f}_{obj} \in \mathbb{R}^{1280\times 3}$ refers to object mesh faces.

Object reconstruction loss $\mathcal{L}_{obj}$. As object mesh is reconstructed in the camera coordinate frame, it can be directly optimised by minimising the Chamfer distance as in [29]. The resulting loss is defined as:

$$\mathcal{L}_{obj}(\mathbf{v}_{obj}) = \frac{1}{2}\Big(\sum_{x\in\mathbf{v}_{obj}} d_{\mathbf{v}^*_{obj}}(x) + \sum_{y\in\mathbf{v}^*_{obj}} d_{\mathbf{v}_{obj}}(y)\Big), \quad (3)$$

where $\mathbf{v}^*_{obj}$ refers to the points uniformly sampled on the surface of the ground truth object, $d_{\mathbf{v}^*_{obj}}(x) = \min_{y\in\mathbf{v}^*_{obj}} \|x-y\|_2^2$, and $d_{\mathbf{v}_{obj}}(y) = \min_{x\in\mathbf{v}_{obj}} \|x-y\|_2^2$.

3.3. Attention-guided graph convolution g^{conv}

Preliminary. We propose to use the message passing scheme [27] in graph convolution to capture mesh information and transfer to the opposite branch. By denoting vertex feature $\mathbf{v}_i^{(k)} \in \mathbb{R}^F$ of vertex i in layer k, the first step of such message passing scheme can be described as:

$$\mathbf{msg}_i^k = \text{AGGREGATE}^{(k)}\big(\{\mathbf{v}_u^{(k-1)}, u \in \mathcal{N}(i)\}\big), \quad (4)$$

where message $\mathbf{msg}_i^k$ is formed by aggregating neighbourhood $\mathcal{N}(i)$ around vertex i from previous layer $(k-1)$. The second step updates vertex feature with this new message:

$$\mathbf{v}_i^k = \text{UPDATE}^{(k)}\big(\mathbf{v}_i^{(k-1)}, \mathbf{msg}_i^k\big). \quad (5)$$

The choice for neighbourhood $\mathcal{N}(i)$, aggregating function $\text{AGGREGATE}^{(k)}$ and update function $\text{UPDATE}^{(k)}$ are crucial. There has been a variety of functions proposed in the literature [21, 27, 40, 76]. In this work, we propose to leverage attention mechanism to construct aggregating neighbourhood and a history term for updating node features.

Objective. By defining P to be the number of iterations per forward pass, the input is a sequence of meshes $(\mathbf{m}^1_\theta, \mathbf{m}^2_\theta, ...)$ where $\mathbf{m}^t_\theta = (\mathbf{v}^t_\theta, \mathbf{f}^t_\theta)$ for $t \in [1, \dots, P]$ is defined by vertices $\mathbf{v}^t_\theta$ and faces $\mathbf{f}^t_\theta$ for either branch $\theta \in \{hand, obj\}$. The objective is to estimate feature offset Δ^t_{hand} from the hand branch for object reconstruction, and vice versa:

$$\mathbf{r}^{t+1}_{obj} = \mathbf{r}^t_{obj} + \Delta^t_{hand}. \quad (6)$$

Attention-guided graph convolution. As the above sequential task involves dynamically evolving graphs, static graph convolution would not be suitable because the weights are only being updated after P iterations. Therefore, a solution should maintain the history of operations. Furthermore, our experiments confirm that static graph convolutions that assumes fixed neighbourhood do not benefit from increasing iterations P (see Table 6).

By assuming input mesh vertices $\mathbf{v}_\theta$ is an un-ordered set, we propose to dynamically construct neighbourhoods $\mathcal{N}(i)$ using attention mechanism [5, 26]. Attention coefficient $\alpha_{ij} \in [0, 1]$ is defined as the importance of vertex j's features to vertex i [68]. Node j is included in the neighbourhood $\mathcal{N}(i)$ of i when α_{ij} is larger than a threshold, *i.e.* 0.5. Finally, our proposed graph convolution layer at iteration t can be defined by rewriting Eqs. (4–5) as:

$$\alpha^t_{ij} = \frac{\exp\Big(\text{LeakyReLU}(\mathbf{a}^\top[\mathbf{W}\mathbf{v}^t_i \| \mathbf{W}\mathbf{v}^t_j])\Big)}{\sum_{k\in\mathbf{v}^t}\exp\Big(\text{LeakyReLU}(\mathbf{a}^\top[\mathbf{W}\mathbf{v}^t_i \| \mathbf{W}\mathbf{v}^t_k])\Big)} \quad (7)$$

where attention coefficient α^t_{ij} is computed using incoming vertices $\mathbf{v}^t = \{\mathbf{v}^t_1, ..., \mathbf{v}^t_N\}$ with N being the maximum mesh vertices and learnable weights $\mathbf{a} \in \mathbb{R}^{2F}$ and $\mathbf{W} \in \mathbb{R}^{F\times 3}$. Note that F is a hyperparameter and $\|$ is concatenation operation. We then update history $\mathbf{h}^t_i$ of vertex i:

$$\mathbf{h}^{t+1}_i = \text{LayerNorm}\Big(\frac{1}{K}\sum_{k=1}^{K}\sum_{j\in\mathcal{N}^t(i)} \alpha^{t,k}_{ij}\mathbf{v}^t_j + \mathbf{h}^t_i\Big), \quad (8)$$

where $\mathcal{N}^t(i)$ is the aggregating neighbourhood around vertex i at t, history $\mathbf{h}^t = \{\mathbf{h}^t_1, ..., \mathbf{h}^t_N\}$ and it is initialised as $\mathbf{0}$. Similar to [67, 68], we find multi-head attention α^k_{ij} to be beneficial and apply layer normalisation [2] to stabilise and enable faster training. We use residual connection [36] to track the history sequence and prevent performance drop on increasing iterations. In the final step, we use a fully connected layer to resize to the same size as image features $\mathbf{r}_\theta(\mathbf{x})$, namely ϕ_θ.

Discussions. Our proposed graph convolution is reminiscent to GAT [68] and any k-nearest neighbours (k-NN) based dynamic graph convolutions like EdgeConv [73]. However, our approach differentiates from those because

firstly, we do not assume static graph inputs. Secondly, we differentiate from GAT [68] by how we leverage attention mechanism - they aggregate on fixed and local neighbourhood whereas we take this further by dynamically constructing global neighbourhood using attention mechanism. In addition, as the incoming mesh are 3D positions, k-NN like approaches suffer from local neighbourhood aggregation and high k-NN computational cost at each iterations. In short, our proposed method is able to capture long-range dependencies from dynamic graph in a single layer. In Table 6, we experiment with two common graph convolution operators (GCN [40] and spiral mesh convolution [28, 41]) and demonstrate superior performance of our proposed attention-guided graph convolution.

3.4. Associative supervision

Due to mutual occlusion in hand-object scenarios [53], it is challenging for the image encoder to capture useful information for mesh reconstruction. Instead, here we rely on the fact that hand pose changes with respect to different objects. For example, we hold cups differently depending on whether it has handle or not. We hypothesise that object branch benefits from hand mesh information (and vice versa for hand branch) and assume that good feature transfer in collaborative learning occurs when these features are highly similar within the same object class and distinctive across all other object classes. However, in practice, such object class information is not available. Hence, we propose an unsupervised loss to facilitate effective feature transfer.

Given $\phi_\theta = \{\phi_\theta^1, ..., \phi_\theta^B\}$ with B being the input batch size, we update the image features by simple addition. In the following, we describe an unsupervised loss for ϕ_θ.

Associative loss $\mathcal{L}_{asso}$. Our approach is inspired by [30] which was originally designed for semi-supervised learning. We imagine a walker going along $\Phi_i = [\phi_{hand}^i; \phi_{obj}^i]$ where $i \in \{1, \dots, B\}$. As each Φ_i comes in pair with the same object class, we define a correct walk if transition is under the same object class. We define similarity between two embeddings as:

$$M_{ij} = \Phi_i^\top \Phi_j, \quad 1 \leq i, j \leq B. \tag{9}$$

A single transition based on embeddings similarity is defined as:

$$P_{ij} = P(\Phi_j | \Phi_i) = \frac{\exp(M_{ij})}{\sum_{j'} \exp(M_{ij'})}. \tag{10}$$

The round trip probability (Markov Chain) of walking from i to j can then be defined as:

$$P_{ij}^{round} = \sum_{k \in \{1,\dots,B\}} P_{ik} P_{kj}. \tag{11}$$

We further extend this into an unsupervised loss by encouraging the walker to walk back to its starting batch index i. This can be achieved by leveraging the fact that batch index implicitly refers to an object class $C_{obj} \in \{1, \dots, O\}$ and $O \ll B$. An unsupervised loss $\mathcal{L}_{asso}$ can be obtained as:

$$\mathcal{L}_{asso}(\phi_\theta) = \left\| U - P^{round} \right\|_F^2, \tag{12}$$

where $\| \cdot \|_F$ is the Frobenius norm and U is a diagonal matrix of $\frac{1}{O}$ values: The i-th diagonal entry U_{ii} represents that the walker starts at and returns to state i. U can be adjusted if dataset is class-imbalanced.

4. Experiments

Implementation details. We implement our method in PyTorch [55]. All experiments are run on an Intel i9-CPU @ 3.50GHZ, 16 GB RAM, and one NVIDIA RTX 3090 GPU. We train all parts of the network simultaneously with Adam optimiser [39] at a learning rate 10^{-4} for 400 epochs. We then freeze the ResNet [36] encoders and decrease the learning rate to 10^{-5} for another 100 epochs. We empirically fixed $K = 3$ attention heads and $P = 2$ iterations to produce the best results. Our final loss $\mathcal{L}_{final}$ is defined as:

$$\mathcal{L}_{final} = \mathcal{L}_{hand} + \mathcal{L}_{obj} + \mathcal{L}_{asso}. \tag{13}$$

Datasets. *First-person hand benchmark (FHB).* This is a widely-used dataset [24] which contains egocentric RGB-D videos on a wide range of hand-object interactions. The ground-truth of hand and object poses are captured via magnetic sensors. There are 4 available objects, *i.e.* juice bottle, liquid soap, milk and salt. For fair comparisons with [33, 65], we follow the same *action split* for evaluation where each object is present in both training and testing. We also compare with [35] which uses the *subject split* of the dataset following their experimental settings: They filtered frames when the hand is further than $1cm$ away from the manipulated object and excluded the milk object. We call this subset *FHB*$^-$ which contains a total of 3 objects.

Algorithm 1 Collaborative learning algorithm

Require: $\mathbf{x}$: input image, P : network iteration

function OPTIMISE($\mathcal{L}_{Total}$)
  $\mathbf{r}_{hand} \leftarrow \text{ENC}_{hand}(\mathbf{x})$ ▷ Extract hand features
  $\mathbf{m}_{hand} \leftarrow \text{g}^{HME}(\mathbf{r}_{hand})$ ▷ Get hand mesh
  for $t = 1$ to P **do**
    $\phi_{hand} \leftarrow g_{hand}^{conv}(\mathbf{m}_{hand})$ ▷ Hand Graph Conv.
    $\mathbf{r}_{obj} \leftarrow \text{ENC}_{obj}(\mathbf{x}) + \phi_{hand}$ ▷ Feature update
    $\mathbf{m}_{obj} \leftarrow \text{g}^{OME}(\mathbf{r}_{obj})$ ▷ Get object mesh
    $\phi_{obj} \leftarrow g_{obj}^{conv}(\mathbf{m}_{obj})$ ▷ Object Graph Conv.
    $\mathbf{r}_{hand}{}' \leftarrow \mathbf{r}_{hand} + \phi_{obj}$ ▷ Feature update
    $\mathbf{m}_{hand} \leftarrow \text{g}^{HME}(\mathbf{r}_{hand}{}')$
  end for
end function

ObMan. This is a large synthetic dataset [35] which was produced by rendering hand meshes with selected objects from ShapeNet [16]. It captures 8 object categories and results in a total of 2,772 meshes which are split among 154,000 image frames. We pretrained the network on *ObMan* before training on other real datasets: We observed in our preliminary experiments that their setting led to consistent improvements over training directly on real data.

DexYCB. This is a recent real dataset for capturing hand grasping of objects [17]. It consists a total of 582,000 image frames on 20 objects from YCB-Video dataset [75]. We present results on all 4 official dataset split settings.

HO-3D. [31] is most similar to *DexYCB* where it consists of 78,000 images frames on 10 objects. We present results on the official dataset split (version 2). The hand mesh error is reported after procrustes alignment and in mm.

Evaluation metrics. *Hand error.* We report the mean endpoint error (mm) over 21 joints and use the percentage of correct keypoints (PCK) score to evaluate at different error thresholds.

Object error. We measure the accuracy of object reconstruction by computing the Chamfer distance (mm) between points sampled on ground truth and predicted mesh.

Hand-object interaction. To understand hand-object interaction, we followed [35] to include penetration depth (mm) and intersection volume (cm^3). Penetration depth refers to the maximum distances from hand mesh vertices to the object's surface when in a collision. Intersection volume is obtained by voxelising the hand and object using a voxel size of $0.5cm$.

Results. *Joint hand-object reconstruction.* As recent efforts on joint hand-object reconstructions [14, 33, 34, 38, 78] assume known object models, we compare with [35] (adopted differential MANO model, AtlasNet and does not assume known object models) in Table 1. Similar to *FHB*, we used the default *DexYCB* split and filtered frames when hand and manipulated object are $1cm$ apart. We name this subset to be *DexYCB*$^-$ and retrain [35] using their released code. As shown, there is still a presence of interpenetration at test time and even increases the hand error by $0.7mm$ on *FHB*$^-$ with contact loss in [35]. This is mainly due to the fact that their model is not implicitly learning the physical rules imposed by the contact loss. In contrast, our method consistently outperforms [35] with a higher hand-object reconstruction accuracy. In addition, we provide qualitative comparisons on *FHB* and *CORe50* [43] datasets in Fig. 3.

Hand pose estimation. We first compare with state-of-the-art methods on *HO-3D* [31] in Table 2. As shown, our method performs competitively against methods that assumes known object models. Then, we compare on *FHB* (both *action split* and *subject split*) in Table 3 and 4. Note that [33] is an extension to [35] which leverages photometric consistency but required known object model. As shown in Table 3, we demonstrate superior performances among all three architecturally similar networks [33, 35]. We attribute the performance gain in *action split* (*i.e.* *FHB*) to the fact that *FHB*$^-$ contains almost half of *FHB* with incom-

Figure 3. Qualitative comparison with ObMan [35]. Top two rows refers to models trained with *FHB*. Bottom two rows refers to in-the-wild settings where models are only trained with synthetic dataset *ObMan*. Our method is able to refine and sharpen object mesh under the collaborative learning framework (see blue arrows) and generalise better hand pose in both settings.

Table 1. Quantitative comparison with ObMan [35] on *ObMan*, *FHB*$^-$ and *DexYCB*$^-$ datasets. * refers to the results with contact loss. Our proposed collaborative learning strategy performs competitively without physical contact loss.

Datasets	*ObMan*			*FHB*$^-$			*DexYCB*$^-$	
Method	[35]	[35]*	Ours	[35]	[35]*	Ours	[35]*	Ours
Hand error (mm)↓	11.6	11.6	**9.1**	28.1	28.8	**25.3**	17.6	**15.3**
Object error (mm)↓	641.5	637.9	**385.7**	1579.2	1565.0	**1445.0**	549.4	**501.2**
Max. penetration (mm)↓	9.5	9.2	**7.4**	18.7	**12.1**	16.1	14.6	**12.1**
Intersection vol. (cm^3)↓	12.3	12.2	**9.3**	26.9	16.1	**14.7**	14.9	**13.4**

Table 2. Error rates of different hand pose estimation methods on *HO-3D*. Note that the reported results for [42] output hand meshes only. We outperform two other architecturally similar networks [33, 35] without known object models under our collaborative learning framework.

Method	Mesh error ↓	F-score @$5mm$ ↑	F-score @$15mm$ ↑	Known objects
[35]	11.0	46.0	93.0	✗
[31]	10.6	50.6	94.2	✓
[42]	**9.5**	**52.6**	**95.5**	✓
[33]	11.4	42.5	93.4	✓
Ours	10.9	48.5	94.3	✗

Table 3. Error rates of different algorithms. *FHB* refers to *action split* and *FHB*$^{-}$ refers to *subject split* of the dataset.

Method	*FHB* Hand Error	*FHB*$^{-}$ Hand Error
Tekin *et al*. [65]	15.8	-
Hasson *et al*. [33]	-	28.0
Hasson *et al*. [35]	18.0	27.4
Cao *et al*. [14]	14.2	-
Ours	**9.8**	**25.3**

Table 4. PCK performance over respective error threshold on *FHB*. Compared to another collaborative learning framework [79] and graph-based method [23], our method performs better and is able to reconstruct both hand-object meshes.

Method	PCK@20mm	PCK@25mm
Tekin *et al*. [65]	69.17%	81.25%
Hernando *et al*. [24]	74.73 %	82.10%
Yang *et al*. [79]	81.03%	86.61%
Doosti *et al*. [23]	92.17%	92.63%
Ours	**93.14%**	**95.65%**

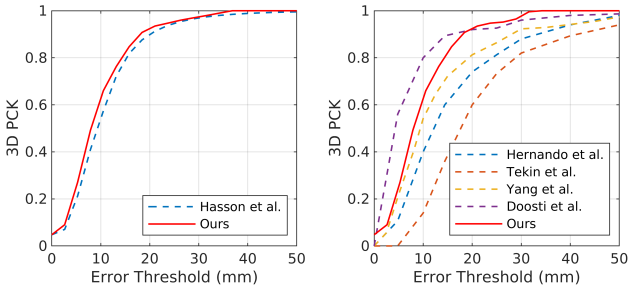

Figure 4. 3D PCK for *ObMan* (*left*) and *FHB* (*right*). Note that Hasson *et al*. refers to [35], and Doosti *et al*. [23] is a hand-object pose estimation method where known object is given.

plete object list and unseen test subjects during test time. We analyse our hand pose estimation performance using the PCK metric in Table 4. Note that Yang *et al*. [79] takes sequential images as input and leverages action recognition task in their collaborative framework. We achieve state-of-the-art performance to in hand pose estimation with the advantage of object reconstruction. 3D PCK curves are shown in Fig. 4. Finally, we compare with a supervised version of Spurr *et al*. [62] which won the HANDS 2019 Challenge [1] on *DexYCB* [17]. In Table 5, the numbers are obtained from [17] where [62] has a HRNet32 [64] backbone.

Ablation study. To motivate our design choices, we present a quantitative comparison of our method with various components disabled. We validate that the combination of our design choices outperforms the naïve collaborative learning baseline (see supplementary), which predicts the embeddings directly and perform 3D reconstruction last.

Table 5. Error rates on *DexYCB* and [62] is the winner of HANDS 2019 Challenge [1]. Table indicates hand error (mm) with AUC values in parentheses. S0-S3 are the official dataset splits [17].

	S0	S1	S2	S3
[62]	17.34(0.698)	22.26(0.615)	**25.49(0.530)**	18.44(0.686)
Ours	**16.05(0.722)**	**21.22(0.620)**	27.01(0.521)	**17.93(0.698)**

Table 6. Performances of different network design choices on *FHB*$^{-}$. We experiment on network iterations P, associative loss $\mathcal{L}_{asso}$ and different convolution operators. The baseline on the first row is same as ObMan [35].

Method	w $\mathcal{L}_{asso}$		w/o $\mathcal{L}_{asso}$	
	Hand Error	Object Error	Hand Error	Object Error
Baseline	-	-	28.4	1655.2
Baseline ($P = 1$)	26.9	1600.3	27.4	1625.9
Baseline ($P = 2$)	**25.3**	**1445.0**	26.3	1618.4
Baseline ($P = 3$)	25.4	1448.2	26.4	1620.5
Baseline ($P = 4$)	25.3	1447.9	26.3	1612.9
Baseline ($P = 5$)	25.3	1445.6	26.2	1618.8
GCN [40] ($P = 1$)	27.1	1587.6	27.8	1629.8
GCN [40] ($P = 2$)	27.0	1590.8	28.2	1635.1
Spiral [28, 41] ($P = 1$)	26.8	1581.8	27.6	1630.1
Spiral [28, 41] ($P = 2$)	26.9	1600.2	27.6	1629.5

Impact of the number of network iterations (P): Table 6 shows the results of varying P with associative loss and demonstrate that associative loss contributes to improving hand and object error. This can be expected since hand-object reconstruction are highly correlated such that learning in a collaborative manner enables performance boost to each other. The effectiveness of our proposed dynamic graph convolution can be demonstrated by the fast performance saturation at $P = 2$. Note that we took [35] as our baseline and graph convolution is enabled from $P = 1$.

Comparison with static graph convolution: To motivate our proposed dynamic graph convolution, we experiment with two commonly used graph convolution in Table 6, *i.e*. GCN [40] and spiral mesh convolution [28, 41]. As the graph convolutions weights are only updated after P iterations, increasing network iterations will have zero effects. It can be seen that static graph convolution does not benefit from increasing network iterations. We also observed that our unsupervised associative loss ($\mathcal{L}_{asso}$) consistently improves hand-object error across Table 6.

Effectiveness of associative loss ($\mathcal{L}_{asso}$): To further study the effect of our unsupervised $\mathcal{L}_{asso}$, we plot the training loss for the collaborative framework, with and without associative loss in Fig. 5. Unsurprisingly, we find that increasing network iterations P contributes to a higher convergence rate (right of Fig. 5). We also observe that our unsupervised associative loss ($\mathcal{L}_{asso}$) is able to stabilise the training across all iterations (left of Fig. 5). This shows that training with $\mathcal{L}_{asso}$ is crucial for this framework.

Mesh generation within iterations: We target the problem of **mutual occlusion** of interacting hand and object by sharing 3D information at each iteration via graph convo-

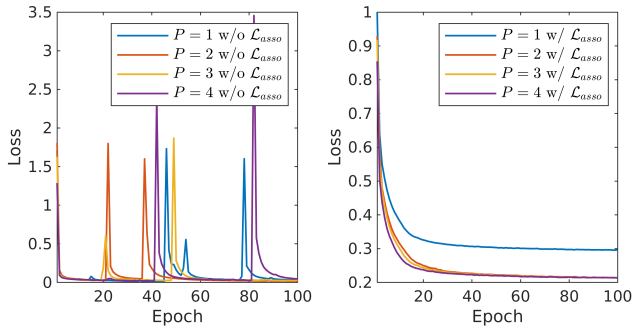

Figure 5. Progression of training losses for iterations $P = \{1, \ldots, 4\}$, without (*left*) and with (*right*) associative loss $\mathcal{L}_{asso}$.

Table 7. Ablation studies on collaborative learning framework design. We experiment on both *FHB*$^-$ and the default *DexYCB* (S0) dataset split. * refers to the naïve collaborative learning baseline.

Method		*FHB*$^-$		*DexYCB* (S0)	
		Hand Error	Object Error	Hand Error	Object Error
$P = 1$	Ours*	28.0	1759.4	17.9	563.4
	Ours	**26.9**	**1600.3**	**17.6**	**529.3**
$P = 2$	Ours*	27.6	1726.8	17.5	554.6
	Ours	**25.3**	**1445.0**	**16.1**	**461.1**
$P = 3$	Ours*	27.1	1678.1	17.3	542.1
	Ours	**25.4**	**1448.2**	**16.0**	**464.2**

lution. To validate this design choice, we construct a simpler collaborative learning framework which directly predicts embeddings ϕ_θ and reconstruct meshes m_θ at the final stage (see supplementary diagram). As *FHB* has limited backgrounds and visible magnetic sensors, we compare the two design on *FHB* and *DexYCB*. Table 7 shows that our final design consistently outperforms the naïve composition baseline across both datasets. We observe that sharing 3D mesh information across hand and object branches improves both reconstruction performance. At the bottom right of Fig. 2, we provide a qualitative example of how reconstruction changes with graph convolution. It can be confirmed that our attention-guided graph convolution combined with collaborative learning enables better mesh quality as well as more accurate pose estimation. We provide additional qualitative results in Fig. 6.

5. Conclusion

In this paper, we have proposed a novel collaborative learning framework which allows the sharing of mesh information across hand and object branches iteratively. The main idea behind this study was to demonstrate that mutual occlusion can be tackled in a learning-based strategy. We designed an attention-guided graph convolution which captures long-range dependencies from dynamic graph in a single layer. However, training with increasing network iterations can be highly unstable. Therefore, we proposed an unsupervised associative loss to stabilise the training and improve the feature transferring process. Our method demonstrated superior performance when compared to other existing approaches on multiple widely-used datasets.

Limitations. Our work relied on AtlasNet for object reconstruction, and we observed that the object reconstruction quality varies with the size of training data. Furthermore, we have only considered static objects, hence future works should consider the interaction between hands and articulated objects.

Potential negative societal impact. Our method can facilitate hand-based interaction in various applications including augmented and virtual reality. In general, advances in hand-based interaction can potentially introduce a barrier to or discourage people having difficulty in using their hands. This could be mitigated when accompanied by technical advances in other modes of interaction, *e.g.* eye or mouse tracking, or body gesture-based interaction.

Acknowledgements

This research was supported by the Ministry of Science and ICT, Korea, under the Information Technology Research Center (ITRC) support program (IITP-2022-2020-0-01789) supervised by the Institute for Information & Communications Technology Planning & Evaluation (IITP) and an IITP grant (2021-0-00537). The computations described in this research were performed using the Baskerville Tier 2 HPC service (https://www.baskerville.ac.uk/) that was funded by EPSRC Grant EP/T022221/1 and is operated by Advanced Research Computing at the University of Birmingham. KIK was supported by the National Research Foundation of Korea (NRF) grant (No. 2021R1A2C2012195) funded by the Korea government (MSIT).

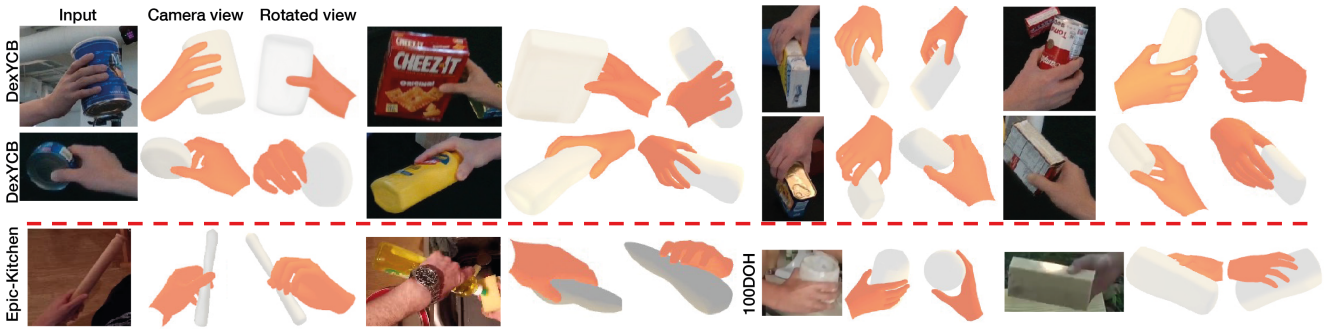

Figure 6. Qualitative results on *DexYCB* (top two rows), *EPIC-Kitchens* [20] (left of bottom row) and *100 Days of Hands* (100DOH) [59] (right of bottom row). The bottom row refers to in-the-wild settings. Our model, trained only on *DexYCB*, shows robustness to various hand poses, objects and scenes.

References

[1] Anil Armagan, Guillermo Garcia-Hernando, Seungryul Baek, Shreyas Hampali, Mahdi Rad, Zhaohui Zhang, Shipeng Xie, MingXiu Chen, Boshen Zhang, Fu Xiong, et al. Measuring generalisation to unseen viewpoints, articulations, shapes and objects for 3D hand pose estimation under hand-object interaction. In *ECCV*, 2020. 7

[2] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv:1607.06450*, 2016. 4

[3] 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. 1

[4] Seungryul Baek, Kwang In Kim, and Tae-Kyun Kim. Weakly-supervised domain adaptation via gan and mesh model for estimating 3d hand poses interacting objects. In *CVPR*, 2020. 1

[5] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *ICLR*, 2015. 4

[6] Maria Florina Balcan, Avrim Blum, Shai Fine, and Yishay Mansour. Distributed learning, communication complexity and privacy. In *Conference on Learning Theory*. JMLR Workshop and Conference Proceedings, 2012. 2

[7] Jonathan Baxter. A bayesian/information theoretic model of learning to learn via multiple task sampling. *Machine learning*, 28(1):7–39, 1997. 2

[8] Jonathan Baxter. A model of inductive bias learning. *Journal of artificial intelligence research*, 12:149–198, 2000. 2

[9] Avrim Blum, Nika Haghtalab, Ariel D Procaccia, and Mingda Qiao. Collaborative PAC learning. In *NeurIPS*, 2017. 2

[10] Adnane Boukhayma, Rodrigo de Bem, and Philip HS Torr. 3d hand shape and pose from images in the wild. In *CVPR*, 2019. 1

[11] Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. Spectral networks and locally connected networks on graphs. In *ICLR*, 2014. 2

[12] Yujun Cai, Liuhao Ge, Jianfei Cai, and Junsong Yuan. Weakly-supervised 3d hand pose estimation from monocular rgb images. In *ECCV*, 2018. 1

[13] Yujun Cai, Liuhao Ge, Jun Liu, Jianfei Cai, Tat-Jen Cham, Junsong Yuan, and Nadia Magnenat Thalmann. Exploiting spatial-temporal relationships for 3d pose estimation via graph convolutional networks. In *ICCV*, 2019. 2

[14] Zhe Cao, Ilija Radosavovic, Angjoo Kanazawa, and Jitendra Malik. Reconstructing hand-object interactions in the wild. In *ICCV*, 2021. 1, 2, 6, 7

[15] Rich Caruana. Multitask learning: A knowledge-based source of inductive bias. In *ICML*, 1993. 2

[16] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv:1512.03012*, 2015. 6

[17] 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 *CVPR*, 2021. 2, 6, 7

[18] Jiecao Chen, Qin Zhang, and Yuan Zhou. Tight bounds for collaborative pac learning via multiplicative weights. In *NeurIPS*, 2018. 2

[19] Hongsuk Choi, Gyeongsik Moon, and Kyoung Mu Lee. Pose2Mesh: Graph convolutional network for 3d human pose and mesh recovery from a 2d human pose. In *ECCV*, 2020. 1, 2

[20] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. Scaling egocentric vision: The EPIC-KITCHENS dataset. In *ECCV*, 2018. 8

[21] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *NeurIPS*, 2016. 2, 4

[22] Ofer Dekel, Ran Gilad-Bachrach, Ohad Shamir, and Lin Xiao. Optimal distributed online prediction. In *ICML*, 2011. 2

[23] 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, 7

[24] 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. 1, 2, 5, 7

[25] 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 *CVPR*, 2019. 1, 2

[26] Jonas Gehring, Michael Auli, David Grangier, and Yann N Dauphin. A convolutional encoder model for neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, 2017. 4

[27] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *ICML*, 2017. 2, 4

[28] Shunwang Gong, Lei Chen, Michael Bronstein, and Stefanos Zafeiriou. SpiralNet++: A fast and highly efficient mesh convolution operator. In *ICCV Workshops*, 2019. 5, 7

[29] 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. 4

[30] Philip Haeusser, Alexander Mordvintsev, and Daniel Cremers. Learning by association–a versatile semi-supervised training method for neural networks. In *CVPR*, 2017. 5

[31] Shreyas Hampali, Mahdi Rad, Markus Oberweger, and Vincent Lepetit. Honnotate: A method for 3d annotation of hand and object poses. In *CVPR*, 2020. 2, 6

[32] Shangchen Han, Beibei Liu, Randi Cabezas, Christopher D Twigg, Peizhao Zhang, Jeff Petkau, Tsz-Ho Yu, Chun-Jung Tai, Muzaffer Akbay, Zheng Wang, et al. MEgATrack: monochrome egocentric articulated hand-tracking for virtual reality. In *SIGGRAPH*, 2020. 1

[33] 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, 5, 6, 7

[34] Yana Hasson, Gül Varol, Ivan Laptev, and Cordelia Schmid. Towards unconstrained joint hand-object reconstruction from RGB videos. In *3DV*, 2021. 1, 2, 6

[35] 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, 4, 5, 6, 7

[36] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 3, 4, 5

[37] Zhanhong Jiang, Aditya Balu, Chinmay Hegde, and Soumik Sarkar. Collaborative deep learning in fixed topology networks. In *NeurIPS*, 2017. 2

[38] Korrawe Karunratanakul, Jinlong Yang, Yan Zhang, Michael J Black, Krikamol Muandet, and Siyu Tang. Grasping field: Learning implicit representations for human grasps. In *3DV*, 2020. 2, 6

[39] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 5

[40] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *ICLR*, 2017. 2, 4, 5, 7

[41] 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, 2, 5, 7

[42] Shaowei Liu, Hanwen Jiang, Jiarui Xu, Sifei Liu, and Xiaolong Wang. Semi-supervised 3d hand-object poses estimation with interactions in time. In *CVPR*, 2021. 1, 6

[43] Vincenzo Lomonaco and Davide Maltoni. Core50: a new dataset and benchmark for continuous object recognition. In *Proceedings of the 1st Annual Conference on Robot Learning*. Proceedings of Machine Learning Research, 2017. 6

[44] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. SMPL: A skinned multi-person linear model. *ACM transactions on graphics (TOG)*, 34(6):1–16, 2015. 2

[45] Priyanka Mandikal and Kristen Grauman. Dexterous robotic grasping with object-centric visual affordances. *arXiv preprint arXiv:2009.01439*, 2020. 2

[46] Yishay Mansour, Mehryar Mohri, and Afshin Rostamizadeh. Domain adaptation with multiple sources. In *NeurIPS*, 2008. 2

[47] Yishay Mansour, Mehryar Mohri, and Afshin Rostamizadeh. Domain adaptation: Learning bounds and algorithms. In *Conference on Learning Theory*, 2009. 2

[48] Andrew N Meltzoff. Understanding the intentions of others: re-enactment of intended acts by 18-month-old children. *Developmental psychology*, 1995. 2

[49] Federico Monti, Davide Boscaini, Jonathan Masci, Emanuele Rodola, Jan Svoboda, and Michael M Bronstein. Geometric deep learning on graphs and manifolds using mixture model cnns. In *CVPR*, 2017. 2

[50] Gyeongsik Moon, Takaaki Shiratori, and Kyoung Mu Lee. DeepHandMesh: A weakly-supervised deep encoder-decoder framework for high-fidelity hand mesh modeling. In *ECCV*, 2020. 1

[51] Franziska Mueller, Florian Bernard, Oleksandr Sotnychenko, Dushyant Mehta, Srinath Sridhar, Dan Casas, and Christian Theobalt. Ganerated hands for real-time 3d hand tracking from monocular rgb. In *CVPR*, 2018. 1

[52] Franziska Mueller, Micah Davis, Florian Bernard, Oleksandr Sotnychenko, Mickeal Verschoor, Miguel A Otaduy, Dan Casas, and Christian Theobalt. Real-time pose and shape reconstruction of two interacting hands with a single depth camera. In *SIGGRAPH*, 2019. 1

[53] Yuzuko C Nakamura, Daniel M Troniak, Alberto Rodriguez, Matthew T Mason, and Nancy S Pollard. The complexities of grasping in the wild. In *Humanoids*, 2017. 1, 5

[54] Huy Nguyen and Lydia Zakynthinou. Improved algorithms for collaborative PAC learning. In *NeurIPS*, 2018. 2

[55] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. Devito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer. Automatic Differentiation in Pytorch. In *NeurIPS*, 2017. 5

[56] Akshay Rangesh, Eshed Ohn-Bar, and Mohan M Trivedi. Hidden hands: Tracking hands with an occlusion aware tracker. In *CVPR*, 2016. 1

[57] Javier Romero, Dimitrios Tzionas, and Michael J Black. Embodied hands: Modeling and capturing hands and bodies together. *ACM Transactions on Graphics (ToG)*, 36(6):245, 2017. 2, 3

[58] 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. 3

[59] Dandan Shan, Jiaqi Geng, Michelle Shu, and David Fouhey. Understanding human hands in contact at internet scale. In *CVPR*, 2020. 8

[60] Tomas Simon, Hanbyul Joo, Iain Matthews, and Yaser Sheikh. Hand keypoint detection in single images using multiview bootstrapping. In *CVPR*, 2017. 1

[61] Guocong Song and Wei Chai. Collaborative learning for deep neural networks. In *NeurIPS*, 2018. 2, 3

[62] Adrian Spurr, Umar Iqbal, Pavlo Molchanov, Otmar Hilliges, and Jan Kautz. Weakly supervised 3D hand pose estimation via biomechanical constraints. In *ECCV*, 2020. 7

[63] Adrian Spurr, Jie Song, Seonwook Park, and Otmar Hilliges. Cross-modal deep variational hand pose estimation. In *CVPR*, 2018. 1

[64] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *CVPR*, 2019. 7

[65] Bugra Tekin, Federica Bogo, and Marc Pollefeys. H+O: Unified egocentric recognition of 3d hand-object poses and interactions. In *CVPR*, 2019. 1, 5, 7

[66] Leslie G Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984. 2

[67] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 4

[68] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph Attention Networks. In *ICLR*, 2018. 4, 5

[69] Chengde Wan, Thomas Probst, Luc Van Gool, and Angela Yao. Crossing nets: Dual generative models with a shared latent space for hand pose estimation. In *CVPR*, 2017. 1

[70] Jialei Wang, Mladen Kolar, and Nathan Srerbo. Distributed multi-task learning. In *AISTATS*, 2016. 2

[71] Jiayi Wang, Franziska Mueller, Florian Bernard, Suzanne Sorli, Oleksandr Sotnychenko, Neng Qian, Miguel A Otaduy, Dan Casas, and Christian Theobalt. RGB2Hands: real-time tracking of 3D hand interactions from monocular rgb video. In *SIGGRAPH*, 2020. 1

[72] Tiancai Wang, Tong Yang, Martin Danelljan, Fahad Shahbaz Khan, Xiangyu Zhang, and Jian Sun. Learning human-object interaction detection using interaction points. In *CVPR*, 2020. 1

[73] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph CNN for learning on point clouds. In *SIGGRAPH*, 2019. 4

[74] Min-Yu Wu, Ya Hui Tang, Pai-Wei Ting, and Li-Chen Fu. Hand pose learning: combining deep learning and hierarchical refinement for 3D hand pose estimation. In *BMVC*, 2017. 1

[75] Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. In *RSS*, 2018. 6

[76] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *ICLR*, 2018. 2, 4

[77] John Yang, Hyung Jin Chang, Seungeui Lee, and Nojun Kwak. SeqHAND: RGB-sequence-based 3D hand pose and shape estimation. In *ECCV*, 2020. 1

[78] 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 *ICCV*, 2021. 2, 6

[79] Siyuan Yang, Jun Liu, Shijian Lu, Meng Hwa Er, and Alex C Kot. Collaborative learning of gesture recognition and 3D hand pose estimation with multi-order feature analysis. In *ECCV*, 2020. 2, 3, 7

[80] Qi Ye and Tae-Kyun Kim. Occlusion-aware hand pose estimation using hierarchical mixture density network. In *ECCV*, 2018. 1

[81] Shanxin Yuan, Guillermo Garcia-Hernando, Björn Stenger, Gyeongsik Moon, Ju Yong Chang, Kyoung Mu Lee, Pavlo Molchanov, Jan Kautz, Sina Honari, Liuhao Ge, et al. Depth-based 3d hand pose estimation: From current achievements to future goals. In *CVPR*, 2018. 1

[82] Shanxin Yuan, Qi Ye, Bjorn Stenger, Siddhant Jain, and Tae-Kyun Kim. Bighand2. 2m benchmark: Hand pose dataset and state of the art analysis. In *CVPR*, 2017. 1

[83] Xiong Zhang, Qiang Li, Hong Mo, Wenbo Zhang, and Wen Zheng. End-to-end hand mesh recovery from a monocular rgb image. In *ICCV*, 2019. 1

[84] Yuxiao Zhou, Marc Habermann, Weipeng Xu, Ikhsanul Habibie, Christian Theobalt, and Feng Xu. Monocular real-time hand shape and motion capture using multi-modal data. In *CVPR*, 2020. 1

[85] Christian Zimmermann and Thomas Brox. Learning to estimate 3d hand pose from single rgb images. In *ICCV*, 2017. 1

[86] 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 *ICCV*, 2019. 1