Harmonious Feature Learning for Interactive Hand-Object Pose Estimation

Zhifeng Lin[1] Changxing Ding[1,2*] Huan Yao[1] Zengsheng Kuang[1] Shaoli Huang[3]
[1] South China University of Technology [2] Pazhou Lab, Guangzhou [3] Tencent AI-Lab, Shenzhen
eezhifenglin@mail.scut.edu.cn, chxding@scut.edu.cn
{mehuanyao,ftkuangzs}@mail.scut.edu.cn, shaolihuang@tencent.com

Abstract

Joint hand and object pose estimation from a single image is extremely challenging as serious occlusion often occurs when the hand and object interact. Existing approaches typically first extract coarse hand and object features from a single backbone, then further enhance them with reference to each other via interaction modules. However, these works usually ignore that the hand and object are competitive in feature learning, since the backbone takes both of them as foreground and they are usually mutually occluded. In this paper, we propose a novel Harmonious Feature Learning Network (HFL-Net). HFL-Net introduces a new framework that combines the advantages of single- and double-stream backbones: it shares the parameters of the low- and high-level convolutional layers of a common ResNet-50 model for the hand and object, leaving the middle-level layers unshared. This strategy enables the hand and the object to be extracted as the sole targets by the middle-level layers, avoiding their competition in feature learning. The shared high-level layers also force their features to be harmonious, thereby facilitating their mutual feature enhancement. In particular, we propose to enhance the feature of the hand via concatenation with the feature in the same location from the object stream. A subsequent self-attention layer is adopted to deeply fuse the concatenated feature. Experimental results show that our proposed approach consistently outperforms state-of-the-art methods on the popular HO3D and Dex-YCB databases. Notably, the performance of our model on hand pose estimation even surpasses that of existing works that only perform the single-hand pose estimation task. Code is available at https://github.com/lzfff12/HFL-Net.

1. Introduction

When humans interact with the physical world, they primarily do so by using their hands. Thus, an accurate understanding of how hands interact with objects is essential to the understanding of human behavior. It can be widely applied to a range of fields, including the development of virtual reality [36], augmented reality [33, 34], and imitation-based robot learning [35], among others. Recently, hand pose estimation [12–16] and 6D object pose estimation [17–19] based on monocular RGB images have respectively achieved remarkable results. However, the research into joint hand-object pose estimation under circumstances of interaction remains in its infancy [2, 3, 23, 26–28].

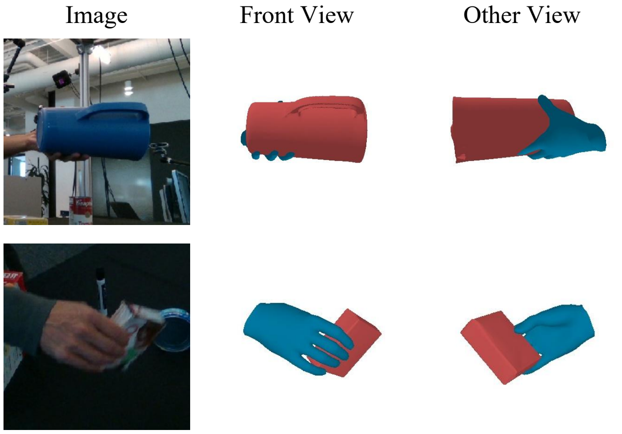


Figure 1. HFL-Net predicts the 3D hand and object poses from single monocular RGB images accurately, even in serious occlusion scenarios.

As illustrated in Figure 1, joint hand-object pose estimation from a single image is extremely challenging. The main reason for this is that when the hand and object interact with each other, serious occlusion occurs; occlusion, in turn, results in information loss, increasing the difficulty of each task.

One mainstream solution to this problem is to utilize context. Due to physical constraints, the interacting hand and object tend to be highly correlated in terms of their poses, meaning that the appearance of one can be useful context for the other [1–3]. Methods that adopt this solution typically employ a single backbone to extract features for the hand and object, respectively [2, 22, 27]. This uni-

[*] Corresponding author.

fied backbone model ensures that the hand and object features are in the same space, which facilitates the subsequent mutual feature enhancement between hand and object via attention-based methods [2].

However, the hand and object pose estimation tasks are competitive in feature learning if a single backbone model is utilized. In more detail, when the hand and object are close to each other, the backbone model treats them both as foreground, and may thus be unable to differentiate the hand features from those of the object. A straightforward solution is to utilize two backbones [1, 3, 23], one for the hand and the other one for the object; when this approach is adopted, each backbone has only one target as the foreground. The main downsides of this strategy include large model size and (more importantly) the different feature spaces between backbones, which introduce difficulties with regard to mutual feature enhancement between the hand and object.

To solve the aforementioned problems, we propose a novel Harmonious Feature Learning Network (HFL-Net). HFL-Net introduces a new framework that combines the advantages of single- and double-stream backbones. In more detail, our backbone shares the parameters of the low- and high-level convolutional layers of a common ResNet-50 model [4] for the hand and object, leaving the middle-level layers unshared. Feature maps produced by low-level layers are fed into the two sets of middle-level layers, which regard the hand and object respectively as the sole foreground target. As a result, feature learning for the hand and object is no longer competitive. Finally, through sharing the parameters of the high-level convolutional layers, the hand and object features are forced to be in similar feature spaces. In this way, our backbone realizes harmonious feature learning for the hand and object pose estimation.

We further enhance the representation power of the hand and object features through the use of efficient attention models. Several existing methods have successfully realized hand-to-object feature enhancement via cross-attention operations [1, 2]; however, object-to-hand feature enhancement usually turns out to be difficult [1, 2]. Motivated by the observation that when one pixel on the hand is occluded, the object feature in the same location usually provides useful cues, we propose a simple but effective strategy for facilitating object-to-hand feature enhancement. Specifically, we adopt ROIAlign [6] to extract fixed-size feature maps from the two output streams of our backbone respectively according to the hand bounding box. We then concatenate the two feature maps along the channel dimension and feed the obtained feature maps into a self-attention module [7]. Object-to-hand feature enhancement is automatically realized via the fully-connected and multi-head attention layers in the self-attention module. Finally, we split the output feature maps by the self-attention layer along the channel dimension, and take the first half as the enhanced hand feature maps.

We demonstrate the effectiveness of HFL-Net through comprehensive experiments on two benchmarks: HO3D [9] and Dex-YCB [10], and find that our method consistently outperforms state-of-the-art works on the joint hand-object pose estimation task. Moreover, benefiting from the learned harmonious hand and object features, the hand and object pose estimation tasks in HFL-Net are mutually beneficial rather than competitive. In our experiments, the performance of HFL-Net on the hand pose estimation task surpasses even recent works [12, 15, 32] that only estimate hand poses in both the training and testing stages.

2. Related Work

2.1. RGB-based 3D Hand Pose Estimation

Existing RGB-based 3D hand pose estimation methods can be roughly grouped into the following two categories: hand model-free methods [14, 16, 29, 38, 39] and hand model-based methods [12, 15, 16, 44–47].

Hand model-free methods predict coordinates of 3D hand joints or 3D hand mesh vertices directly from a single RGB image [14, 16]. Predicting 3D hand joints is easier, but the joints lack geometry information of the hand surface. In comparison, the 3D mesh contains rich geometric topology of the hand. To obtain reasonable topology between mesh vertices, graph convolution [14, 16, 29] is usually adopted to refine the vertex features. However, methods that directly predict hand mesh require dense and accurate 3D vertex annotations, which are difficult to obtain.

Hand model-based methods make use of hand priors to simplify the task of mesh prediction [12, 15, 44–46]. They are usually based on parameterized hand models, e.g., MANO [31], that are pre-trained on a large number of manually-scanned hand meshes in various pose and shapes. With these priors, hand model-based methods only need to estimates a small number of pose and shape coefficients, by which they obtain the hand mesh. Good results can be found in a number of recent works [12, 15, 45].

The above methods focus on the hand pose estimation task and achieve superior performance when there is no hand-object interaction. However, in real-world scenarios, the hand interacts with objects frequently. Hand-object interaction brings in unique challenges to the pose estimation task. In the following, we will review recent works on interactive hand-object pose estimation.

2.2. RGB-based 3D Hand-object Pose Estimation

Existing RGB-based 3D hand-object pose estimation techniques can be divided into two categories: the optimization-based methods [21, 24, 25, 41, 42] and the learning-based methods [1, 2, 20, 22, 23, 27, 43]. The optimization-based methods refine the hand and object pose

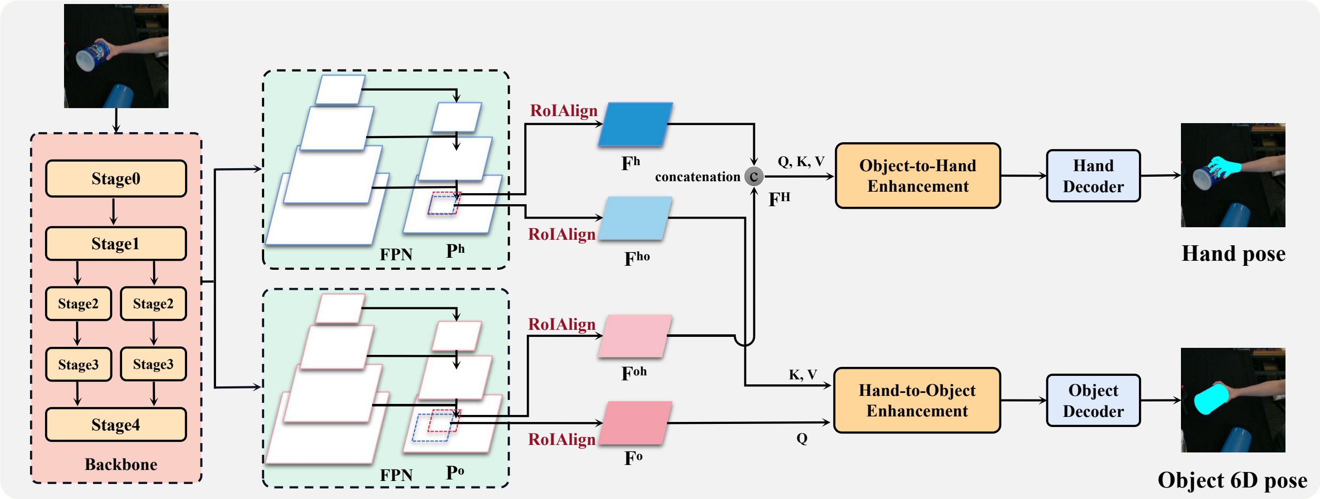


Figure 2. Overview of our HFL-Net framework, which includes an elaborately designed backbone based on ResNet-50, hand and object interaction modules, and hand and object decoders. By adopting independent stage-2 and stage-3 convolutional layers, our backbone model avoids competition in feature learning between the hand and object. Furthermore, due to the shared stage-4 layers, the hand and object features are forced to be in a similar space, enabling us to conduct effective object-to-hand and hand-to-object feature enhancement. Finally, the enhanced hand and object features are fed into their respective decoder for pose estimation. Best viewed in color.

according to their contact surface with physical constraints, i.e., attraction and repulsion. Estimating the contact surface between the hand and object is usually time-consuming [42]. To relieve this problem, Tse et al. [25] proposed a graph-based network to speed up the contact surface estimation.

The learning-based methods design unified models for joint hand and object pose estimation. They typically adopt an off-the-shelf hand model, e.g., MANO [31], and also assume the 3D object model is available. Therefore, they can directly predict the hand and object pose based on these priors. Early works [1, 23] adopt double-stream backbones for independent hand and object pose estimation, at the cost of higher model complexity. Recent works [2, 22, 27] adopt a single-stream backbone to extract hand and object features. However, they ignore the hand and object feature learning are competitive if a single-stream backbone is adopted.

Due to physical constraint, the pose of interactive hand and object are highly correlated with each other. Therefore, the hand and object appearance can be useful context for each other. Existing works have proposed various approaches to utilize this context information. For example, Tekin et al. [43] and Chen et al. [1] integrated hand and object features with recurrent neural networks. Liu et al. [2] utilized a cross-attention layer to enhance the object feature with that of the hand. Hampali et al. [27] employed transformer to model the correlation between the hand or object keypoints, so as to improve the accuracy of hand-object pose estimation.

In this paper, we propose a novel framework that extracts harmonious hand and object features, which not only relieves the competition between the hand and object pose estimation tasks, but also enables effective mutual enhancement between the hand and object features.

3. Methodology

This section presents the framework of our Harmonious Feature Learning Network (HFL-Net) for joint hand and object pose estimation. As shown in Figure 2, HFL-Net comprises an elaborately designed backbone model, two interaction modules between the hand and object, a hand decoder, and an object decoder. The backbone model produces harmonious feature maps for the hand and object, respectively, which enables these feature maps to enhance each other in the subsequent interaction modules. Finally, the hand and object decoders estimate the hand and object poses, respectively. In the below, we will describe these components sequentially.

3.1. Feature Extraction Backbone

We take the popular ResNet-50 [4] model as an example to illustrate the structure of our backbone. Most existing works adopt a single ResNet-50 model [2, 22, 27] as their backbone, which regards both of the hand and object as foreground targets. When occlusion occurs due to hand-object interaction, feature learning for the hand and object becomes competitive, as illustrated in Figure 3. In what follows, we modify the original ResNet-50 model to learn harmonious features for the hand and object, respectively.

The layers in ResNet-50 are divided into five stages, which are denoted as stage-0 to stage-4, according to size of the feature maps [4]. As illustrated in Figure 2, our back-

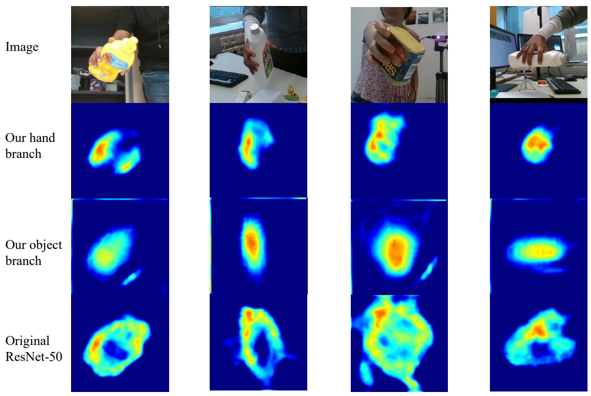
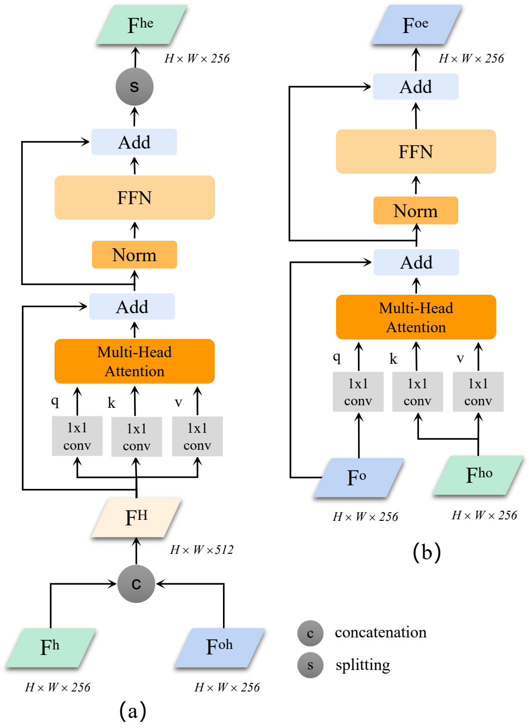


Figure 3. Heatmap illustration of the feature maps output by our backbone and ResNet-50. Heatmaps for our backbone model highlight the hand and object quite clearly under circumstances of occlusion, thanks to our harmonious feature learning scheme. In comparison, the original ResNet-50 backbone results in entangled and obscure hand and object features.

bone keeps the structure of the stage-0, stage-1, and stage-4 layers of the ResNet-50 model unchanged, but adopts independent stage-2 and stage-3 layers for the hand and object, respectively. The feature maps output by the stage-1 layers are fed into the two sets of stage-2 and stage-3 layers, respectively. In this way, each set of stage-2 and stage-3 layers has only one foreground target, and can therefore focus on the feature learning of either the hand or the object.

Finally, the two sets of feature maps output by the stage-3 layers are fed into the same stage-4 layers. Due to the sharing of the stage-4 layers, the feature spaces for the hand and object streams are forced to be unified, facilitating the subsequent interaction operations between their features. Using the same approach as in [2], we adopt Feature Pyramid Network (FPN) [5] to combine the features in stages 1 to 4. Since the hand and object have independent stage-2 and stage-3 layers, they adopt different FPNs. The final feature maps produced by the two FPNs are denoted as $\mathbf{P}^h \in \mathbb{R}^{\frac{H}{4}\times\frac{W}{4}\times 256}$ for the hand stream and $\mathbf{P}^o \in \mathbb{R}^{\frac{H}{4}\times\frac{W}{4}\times 256}$ for the object stream; here H and W denote the height and width of the input image, respectively.

We illustrate the heatmap of the feature maps output by our backbone and ResNet-50 in Figure 3. As the figure shows, the feature maps produced by our backbone model highlight the hand and object quite clearly under circumstances of occlusion, thanks to our harmonious feature learning scheme. By contrast, in the feature maps output by ResNet-50, the hand and object features tend to be competitive and obscure. In the experimentation section, we show that our backbone model achieves significantly better performance than the common single- and double-stream ResNet-50 backbones.



(a)

(b)

Figure 4. Structure of our hand and object interaction modules. The two figures illustrate (a) object-to-hand and (b) hand-to-object feature enhancement, respectively.

3.2. Interaction Modules

Due to physical constraints, interacting hands and objects are highly correlated in terms of their pose, meaning that the appearance of one can provide useful context for the other. In this section, we introduce our hand-to-object and object-to-hand feature enhancement modules, based on the harmonious features produced by our backbone.

As illustrated in Figure 2, we adopt ROIAlign [6] to obtain feature maps $\mathbf{F}^h \in \mathbb{R}^{\frac{H}{8}\times\frac{W}{8}\times 256}$ from $\mathbf{P}^h$ according to the hand bounding box and $\mathbf{F}^o \in \mathbb{R}^{\frac{H}{8}\times\frac{W}{8}\times 256}$ from $\mathbf{P}^o$ according to the object bounding box. Moreover, we employ the same ROIAlign layer to obtain $\mathbf{F}^{oh}$ from $\mathbf{P}^o$ according to the hand bounding box and obtain $\mathbf{F}^{ho}$ from $\mathbf{P}^h$ according to the overlapped area between the hand and object bounding boxes. $\mathbf{F}^{oh}$ and $\mathbf{F}^{ho}$ are utilized for object-to-hand and hand-to-object feature enhancement, respectively.

Object-to-Hand Enhancement. The hand is non-rigid, flexible, and typically occluded when grasping an object; therefore, rich features are required to predict the hand pose. As illustrated in Figure 1, when a hand is seriously occluded, the object in the hand can be a strong cue regarding the hand pose. Motivated by this observation, we propose to enrich the hand feature by directly concatenating $\mathbf{F}^h$ and $\mathbf{F}^{oh}$ along the channel dimension. The obtained new feature maps are denoted as $\mathbf{F}^H \in \mathbb{R}^{\frac{H}{8}\times\frac{W}{8}\times 512}$.

We feed $\mathbf{F}^H$ into a self-attention layer to deeply fuse

$\mathbf{F}^h$ and $\mathbf{F}^{oh}$. As illustrated in Figure 4(a), fusion between $\mathbf{F}^h$ and $\mathbf{F}^{oh}$ is realized from two perspectives in the self-attention layer. First, intra-pixel feature fusion is achieved using 1×1 convolutional layers in the multi-head attention (MHA) and the feedforward network (FFN). Second, inter-pixel feature fusion is realized via the attention operations in MHA. Benefiting from the attention operation, each feature in $\mathbf{F}^{oh}$ implicitly affects the features of all pixels in $\mathbf{F}^h$.

Finally, we slice the feature maps output by the self-attention layer along the channel dimension into two halves and take the first half as the enhanced hand feature $\mathbf{F}^{he} \in \mathbb{R}^{\frac{H}{8} \times \frac{W}{8} \times 256}$.

Hand-to-Object Enhancement. Compared with the hand, the object is more rigid and less flexible. Therefore, we conduct hand-to-object enhancement according to the hand features located within the area of intersection between the hand and object. Similar to [2], we employ a cross-attention layer for hand-to-object enhancement. Specifically, we adopt $\mathbf{F}^o$ as the query and $\mathbf{F}^{ho}$ as both the key and value. The enhanced object feature maps are denoted as $\mathbf{F}^{oe} \in \mathbb{R}^{\frac{H}{8} \times \frac{W}{8} \times 256}$. $\mathbf{F}^{oe}$ is fed into the object decoder to obtain the 6D pose of the object. Different from [2], we obtain $\mathbf{F}^o$ and $\mathbf{F}^{ho}$ from two different but complementary feature maps, i.e., $\mathbf{P}^h$ and $\mathbf{P}^o$. By contrast, the inputs to the cross-attention layer in [2] are all from the same feature maps. In the experimentation section, we show that our strategy achieves better performance compared with [2].

3.3. Hand and Object Decoders

We adopt identical decoder structures to those used in [2]. For the sake of completeness, we briefly introduce these structures below.

Hand Decoder. The hand decoder takes $\mathbf{F}^{he}$ as input to an hourglass network [11], which produces both new feature maps and heatmaps $\mathbf{H} \in \mathbb{R}^{\frac{H}{8} \times \frac{W}{8} \times 21}$ for 21 2D joint points. Subsequently, the new feature maps and heatmaps are fused via 1×1 convolutions and element-wise addition. The fused feature maps are then fed into four successive residual blocks, the outputs of which are flattened into one 1024-dimensional vector. This vector is then fed into two fully connected layers to predict the hand pose and shape parameters according to the MANO model [31]; these are denoted as $\theta \in \mathbb{R}^{48}$ and $\beta \in \mathbb{R}^{10}$, respectively. With the obtained MANO parameters, we finally obtain the estimated 3D hand mesh $\mathbf{V} \in \mathbb{R}^{778 \times 3}$ and 3D coordinates of hand joints $\mathbf{J} \in \mathbb{R}^{21 \times 3}$ for the hand in the 2D image.

Object Decoder. The object decoder takes $\mathbf{F}^{oe}$ as input. It consists of 6 convolutional layers, which output a tensor $\mathbf{C} \in \mathbb{R}^{\frac{H}{8} \times \frac{W}{8} \times 3p}$. p denotes the number of control points, including one center point, 8 corner points and 12 edge midpoints of the 3D bounding box for the object. The tensor predicts the offsets of each pixel in $\mathbf{F}^{oe}$ to the 2D location of each control point in the image as well as the prediction confidence.

3.4. Overall Loss Function

Loss Functions. Since we adopt the same hand and object decoders as in [2], we also employ its loss functions. The following is the total loss function in our training phase:

$$\mathbf{L}_{total} = \mathbf{L}_{hand} + \mathbf{L}_{obj}, \tag{1}$$

where

$$\mathbf{L}_{hand} = \alpha_h \mathbf{L}_H + \alpha_{3d} \mathbf{L}_{3d} + \alpha_{mano} \mathbf{L}_{mano}, \tag{2}$$

$$\mathbf{L}_{obj} = \alpha_{p2d} \mathbf{L}_{p2d} + \alpha_{conf} \mathbf{L}_{conf}. \tag{3}$$

$\mathbf{L}_{hand}$ and $\mathbf{L}_{obj}$ represent the loss functions for the hand and object pose estimation tasks, respectively. $\mathbf{L}_H$ denotes the L2 loss for 2D joint point detection and is imposed on $\mathbf{H}$. $\mathbf{L}_{3d}$ stands for the L2 loss that is imposed on $\mathbf{V}$ and $\mathbf{J}$. $\mathbf{L}_{mano}$ is the L2 loss on MANO parameters β and θ. $\mathbf{L}_{p2d}$ and $\mathbf{L}_{conf}$ are the L1 loss imposed on $\mathbf{C}$. Finally, α_h, α_{3d}, α_{mano}, α_{p2d} and α_{conf} are coefficients that balance the weight of each loss function.

Inference Phase. In the inference phase, the hand decoder directly predicts MANO parameters to obtain $\mathbf{V}$ and $\mathbf{J}$. For the object pose, we first obtain $\mathbf{C}$ via the object decoder. Then, we adopt the Perspective-n-Point (PNP) [37] algorithm with $\mathbf{C}$ as input to predict the object pose in the same way as [2].

4. Experiments

4.1. Implementation Details

We adopt the model proposed in [2] as our baseline, upon which we build our HFL-Net with the novel backbone model and interaction modules. Our backbone is based on a ResNet-50 model [4] that was pretrained on ImageNet. We crop and resize all images in each database to 256 × 256 pixels, which is smaller than the size used in [2]. Sample images can be found in Figure 1. During training, we adopt data augmentation including random scaling, rotation, translation, and color jittering. We set the batch size to 64. Following [2], we set α_h, α_{3d}, α_{mano}, α_{p2d} and α_{conf} to 100, 10000, 1, 500 and 100, respectively. We adopt the Adam optimizer and a weight decay of 5e-4 for optimization. The total number of training epochs is 70. The initial learning rate is 1e-4 and decays for every 10 epochs. All implementations are based on PyTorch.

4.2. Datasets and Metrics

HO3D. The HO3D dataset [9] consists of 77K images from 68 video sequences. It includes 10 objects and 10 subjects. We use the official splitting protocol for the training and testing sets, and submit the test results to the official

Methods	Joint↓	Mesh↓	F@5↑	F@15↑	Object
Pose2Mesh et al. [29]	12.5	12.7	44.1	90.9	No
Hasson et al. [22]	11.4	11.4	42.8	93.2	Yes
I2L-MeshNet [30]	11.2	13.9	40.9	93.2	No
Hasson et al. [23]	11.0	11.2	46.4	93.9	Yes
Hampali et al. [9]	10.7	10.6	50.6	94.2	Yes
METRO [15]	10.4	11.1	48.4	94.6	No
Liu et al. [2]	10.1	9.7	53.2	95.2	Yes
ArtiBoost [28]	11.4	10.9	48.8	94.4	Yes
Keypoint Trans. [27]	10.8	-	-	-	Yes
HandOccNet [12]	9.1	8.8	56.4	96.3	No
Ours	**8.9**	**8.7**	**57.5**	**96.5**	Yes

Table 1. Performance comparison with state-of-the-art methods on hand pose estimation on the HO3D dataset. The last column indicates whether a method performs the object 6D pose estimation task.

Methods	cleanser↑	bottle↑	can↑	average↑
Liu et al. [2]	**88.1**	61.9	**53.0**	67.7
Ours	81.4	**87.5**	52.2	**73.3**

Table 2. Performance comparison with state-of-the-art methods on object 6D pose estimation on the HO3D dataset.

Methods	MPJPE↓	PAMPJPE↓	Object
METRO [15]	15.24	6.99	No
Spurr et al. [32]	17.34	6.83	No
Liu et al. [2]	15.27	6.58	Yes
HandOccNet [12]	14.04	5.80	No
Ours	**12.56**	**5.47**	Yes

Table 3. Performance comparison with state-of-the-art methods on hand pose estimation on the Dex-YCB dataset. The last column indicates whether a method performs the object 6D pose estimation task.

website to report performance. For hand pose estimation, we report the F-scores, the mean joint error (PAMPJPE) and mean mesh error (PAMPVPE) in millimeters following Procrustes alignment. For object 6D pose estimation, we report the percentage of objects whose average vertex error is within 10% of the object diameter (ADD-0.1D) [40]. Following [2,27,28], we evaluate the performance of our model only on the objects that have been seen during training.

Dex-YCB. Dex-YCB [10] is a recently introduced large-scale dataset that includes 582K images from over 1000 video sequences. It covers 10 subjects and 20 objects. This paper presents the results according to the official s0 splitting protocol. For hand pose estimation, we report both PAMPJPE and the mean joint error in millimeters without Procrustes alignment (MPJPE). For object 6D pose estimation, we report ADD-0.1D. Following [40], since many objects in the datasets are symmetric, we use the symmetric version of ADD-0.1D.

Methods	ADD-0.1d(s)↑	
	Liu et al. [2]	Ours
Image Size	512 × 512	256 × 256
master chef can	34.2	23.3
cracker box	56.4	66.6
sugar box	42.4	35.6
tomato soup can	17.1	12.2
mustard bottle	44.3	48.1
tuna fish can	11.9	8.6
pudding box	36.4	31.2
gelatin box	25.6	26.0
potted meat can	21.9	21.1
banana	16.4	16.9
pitcher base	36.9	36.5
bleach cleanser	46.9	42.5
bowl*	30.2	36.2
mug	18.5	16.8
power drill	36.6	45.1
wood block*	38.5	45.9
scissors	12.9	13.6
large marker	2.8	3.7
extra large clamp*	38.9	44.8
foam brick*	27.5	28.8
average	29.8	**30.2**

Table 4. Performance comparison with state-of-the-art methods on object 6D pose estimation on the Dex-YCB database. We denote the objects that are considered to be symmetric by a * superscript.

4.3. Comparisons with State-of-the-Art Methods

Comparisons on HO3D. Performance comparisons on hand pose estimation are summarized in Table 1. The PAMPJPE and PAMPVPE of HFL-Net are 8.9mm and 8.7mm, respectively. Our proposed approach outperforms all state-of-the-art methods on this task. In more detail, HFL-Net outperforms its baseline model [2] by 1.2mm and 1.0mm on PAMPJPE and PAMPVPE, respectively. It even consistently outperforms a very recent work named HandOccNet [12] on all metrics. It is worth noting that here HandOccNet performs the hand pose estimation task only, without considering the object pose estimation task. We attribute the advantage of HFL-Net to the harmonious feature it extracts, which relieves the competition in feature learning between the hand and object, while also achieves effective object-to-hand feature enhancement.

We also conduct comparisons on object 6D pose estimation in Table 2. The average ADD-0.1 score of our method is 73.3%, representing a significantly improvement over [2] by 5.6%. This experiment justifies the effectiveness of our backbone model, which largely removes the interference in feature learning between the hand and the object and therefore obtains better object pose estimation performance.

Moreover, although we adopt a more complex backbone than [2], the time cost of our backbone is in fact lower. This is because the image size in this work is set to 256×256 pixels, while that in [2] is 512×512 pixels. The time cost of our backbone and that in [2] are 3.87ms and 7.5ms per image on a Titan V GPU. Therefore, our model can be more efficiently utilized in practice.

Comparisons on Dex-YCB. Performance comparisons on hand pose estimation are summarized in Table 3. The PAMPJPE and MPJPE of HFL-Net are 5.47mm and 12.56mm, respectively. Our proposed approach outperforms all state-of-the-art methods on this task. In more detail, HFL-Net outperforms its baseline model [2] by 1.11mm and 2.71mm on PAMPJPE and MPJPE, respectively; moreover, it also outperforms HandOccNet [12], which focuses on hand pose estimation. Experiments on this database further justify the superiority of our method.

We also conduct comparisons on the object 6D pose estimation task in Table 4. These results show that our method outperforms [2] with smaller image size. It is worth noting that object pose estimation on the Dex-YCB database is significantly more challenging than that on HO3D. This is because the scene in Dex-YCB is more complex: it usually contains multiple mutually occluded objects in the same image, which brings in severe interference to the object pose estimation task.

4.4. Ablation Study

In the following, we perform ablation studies on the HO3D dataset to demonstrate the effectiveness of our backbone and interaction modules.

Effectiveness of Our Backbone. To show the effectiveness of our backbone, we compare its performance with the commonly used single- and double-stream backbones. The model sizes of the three backbones are 34.97M, 25.08M, and 50.15M, respectively. The single-stream backbone adopts only one ResNet-50 model for both the hand and the object, while the double-stream backbone employs independent ResNet-50 models for the hand and object, respectively.

In Table 5, we first compare the performance of the three backbones without the use of any hand-to-object or object-to-hand feature enhancement modules. As is evident, the double-stream backbone significantly outperforms the single-stream backbone, indicating that the hand and object feature learning are indeed competitive if the single-stream backbone is adopted. Meanwhile, our backbone can attain nearly the same level of performance as that of the two-stream backbone, with considerably fewer model parameters. This comparison reveals that the competition between the hand the object feature learning can be largely alleviated through the adoption of unshared stage-2 and stage-3 layers. In conclusion, our backbone model effectively relieves the competition between the hand and object pose estimation tasks with only a moderate increase in model size.

Methods	Joint↓	Mesh↓	cleanser↑	bottle↑	can↑	average↑
Single-Stream	10.4	10.3	80.1	55.3	46.2	60.5
Double-Stream	9.7	9.6	82.2	74.1	49.4	68.6
Ours	9.8	9.7	84.1	70.3	48.2	67.5

Table 5. Performance comparison between different backbones on the HO3D dataset. No interaction modules are utilized in this experiment. The last column lists the average performance on three objects.

Methods	Joint↓	Mesh↓	cleanser↑	bottle↑	can↑	average↑
Single-Stream	10.2	10.0	86.2	62.1	42.3	63.5
Double-Stream	9.5	9.4	**91.2**	73.3	46.8	70.4
Ours	**8.9**	**8.7**	81.4	**87.5**	**52.2**	**73.3**

Table 6. Performance comparison between different backbones on the HO-3D dataset. The interaction modules introduced in Section 3.2 are employed on top of all backbones.

Our backbone model presents more advantages when the interaction modules are adopted. In Table 6, all three backbones are equipped with the same interaction modules described in Section 3.2. Combining the results of Table 5 and Table 6, it is clear that the performance gain of the double-stream backbone after adopting the interaction modules are quite small. This may be because the outputs of the two streams are not in the same feature space; as a result, interactions between the hand and object features are difficult. In comparison, our backbone produces harmonious hand and object features via sharing layers in stage-4, facilitating the subsequent mutual enhancement between the hand and object features. In Table 6, it is clear that the combination of our backbone and the interaction modules achieves considerably better performance than the results based on single- and double-stream backbones. The above experiments justify the effectiveness of our backbone model.

Effectiveness of Hand-to-Object Feature Enhancement. In Table 7, we study the effect of adopting different features to enhance the object feature. In these experiments, we consistently use $\mathbf{F}^o$ as the query in the cross-attention operation. The first experiment involves adopting $\mathbf{F}^{ho}$ as the key and value, which is adopted in this paper and denoted as H-to-O in Table 7. The second experiment is quite similar to the first one, except that we replace $\mathbf{F}^{ho}$ with its counterpart extracted from $\mathbf{P}^o$. This experiment is denoted as O-to-O. As shown in Table 7, the two methods are comparable in terms of hand pose estimation. However, H-to-O significantly outperforms O-to-O by as much as 8.9% on the ADD-0.1D score for the small object 'can'. This may be because $\mathbf{F}^{ho}$ provides more complementary features since it is cropped from $\mathbf{P}^h$. The above experiments demonstrate that the hand feature is indeed helpful to enhance the object feature, especially for small objects that suffer from more severe hand-object occlusions.

Effectiveness of Object-to-Hand Feature Enhance-

Model	Joint↓	Mesh↓	cleanser↑	bottle↑	ca↑	average↑
w/o inter.	9.8	9.7	84.1	70.3	48.2	67.5
O-to-O	**9.6**	9.5	**92.3**	**74.2**	42.3	69.5
H-to-O	**9.6**	**9.4**	92.1	70.4	**51.2**	**71.3**

Table 7. Ablation study on the hand-to-object feature enhancement module. H-to-O and O-to-O adopt the same query, but different values and keys for cross-attention.

Model	Joint↓	Mesh↓	cleanser↑	bottle↑	can↑	average↑
w/o enhance	9.6	9.4	**92.1**	70.4	51.2	71.3
SA	9.5	9.4	91.2	77.3	51.0	73.1
CA	9.5	9.4	84.4	82.1	44.2	70.2
ADD+SA	9.3	9.3	87.2	78.3	52.1	72.5
Ours	**8.9**	**8.7**	81.4	**87.5**	**52.2**	**73.3**

Table 8. Ablation study on the object-to-hand feature enhancement module. Hand-to-object feature enhancement is adopted by all methods in this table.

ment. In Table 8, we compare the performance of our object-to-hand enhancement module with four possible variants. The first variant does not enhance the hand feature, and is denoted as 'w/o enhance' in the table. The second one is denoted as SA, which is based on self-attention and employs $\mathbf{F}^h$ as the query, key, and value. The third one is denoted as CA, and utilizes $\mathbf{F}^h$ as the query and $\mathbf{F}^o$ as the key and value for cross attention. The fourth method is denoted as 'ADD+SA', which first fuses $\mathbf{F}^h$ and $\mathbf{F}^{oh}$ via element-wise addition and then apply a self-attention module to the fused feature maps.

As shown in the Table 8, both SA and CA boost the hand pose estimation performance slightly. This means that $\mathbf{F}^o$ can help to enhance the representation power of $\mathbf{F}^h$. In addition, ADD+SA achieves better performance than both SA and CA. Finally, our method performs significantly better than all the former three methods in hand pose estimation. In particular, it outperforms ADD+SA by 0.4mm and 0.6mm in PAMPJPE and PAMPVPE, respectively, indicating that concatenation is a more powerful strategy than element-wise addition to fuse $\mathbf{F}^h$ and $\mathbf{F}^{oh}$. These experiment experimental results justify that our hand feature enhancement approach is quite effective for hand pose estimation.

4.5. Qualitative Comparisons

We make qualitative comparisons between HFL-Net and state-of-the-art methods [2, 12] in Figure 5 and Figure 6, respectively. It can be seen from the figures that HFL-Net makes more accurate pose estimation than [2, 12]. Moreover, even in serious occlusion scenarios, HFL-Net still makes reasonable hand pose prediction with our powerful object-to-hand feature enhancement module.

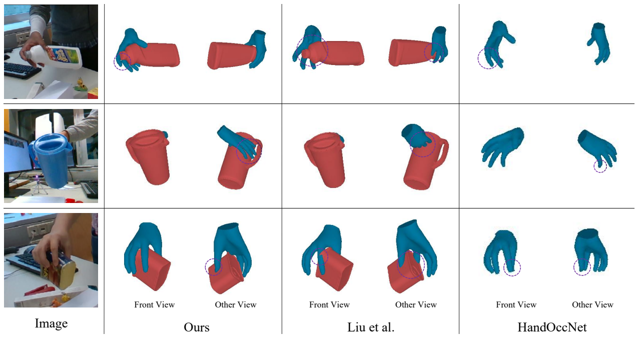


Figure 5. Qualitative comparisons between HFL-Net and [2,12] on the HO3D database. HandOccNet [12] is an approach that predicts the hand pose only.

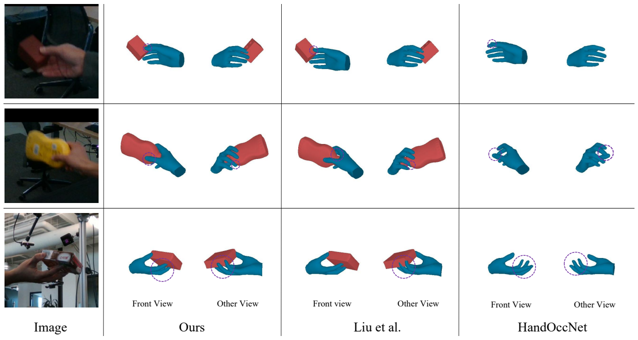


Figure 6. Qualitative comparisons between HFL-Net and [2, 12] on the Dex-YCB database. HandOccNet [12] is an approach that predicts the hand pose only.

5. Conclusion

In this work, we propose a novel Harmonious Feature Learning Network (HFL-Net) with both effective backbone and feature interaction modules. Our backbone shares the parameters of the low- and high-level convolutional layers of a common ResNet-50 model for the hand and object, leaving the middle-level layers unshared. In this way, HFL-Net not only avoids the competition in feature learning between the two, but also extracts harmonious features, facilitating the subsequent mutual enhancement between the hand and object features. In addition, our approach to object-to-hand feature enhancement is both simple and effective, enabling us to outperform methods that focus solely on hand pose estimation only. Experimental results show that our method consistently achieves state-of-the-art performance on standard benchmarks for joint hand and object pose estimation.

Acknowledgment. This work was supported by the National Natural Science Foundation of China under Grant 62076101, Guangdong Basic and Applied Basic Research Foundation under Grant 2023A1515010007, the Guangdong Provincial Key Laboratory of Human Digital Twin under Grant 2022B1212010004, and the Program for Guangdong Introducing Innovative and Entrepreneurial Teams under Grant 2017ZT07X183.

References

[1] Y. Chen, Z. Tu, D. Kang, R. Chen, L. Bao, Z. Zhang, and J. Yuan. Joint hand-object 3d reconstruction from a single image with cross-branch feature fusion. In *TIP*, 2021. 1, 2, 3

[2] S. Liu, H. Jiang, J. Xu, S. Liu, and X. Wang. Semi-supervised 3d hand-object poses estimation with interactions in time. In *CVPR*, 2021. 1, 2, 3, 4, 5, 6, 7, 8

[3] T. Tse, K. Kim, A. Leonardis, and H. Chang. Collaborative Learning for Hand and Object Reconstruction with Attention-guided Graph Convolution. In *CVPR*, 2022. 1, 2

[4] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 2, 3, 5

[5] T. Lin, Feature pyramid networks for object detection. In *CVPR*, 2017. 4

[6] K. He, G. Gkioxari, Mask r-cnn. In *ICCV*, 2017. 2, 4

[7] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. Gomez, Attention is all you need. In *In NIPS*, 2017. 2

[8] J. Ba, J. Kiros, and G. Hinton. Layer normalization. In *arXiv preprint arXiv:1607.06450*, 2016.

[9] S. Hampali, M. Rad, M. Oberweger, and V. Lepetit. Honnotate: A method for 3d annotation of hand and object poses. In *CVPR*, 2020. 2, 5, 6

[10] Y. Chao, W. Yang, Y. Xiang, P. Molchanov, A. a, J. Tremblay, Y. Narang, K. VanWyk, U. Iqbal, and S. Birchfield. DexYCB: A benchmark for capturing hand grasping of objects. In *CVPR*, 2021. 2, 6

[11] A. Newell, K. Yang, and J. Deng. Stacked hourglass networks for human pose estimation. In *ECCV*, 2016. 5

[12] J. Park, Y. Oh, G. Moon, H. Choi, and K. Lee. HandOccNet: Occlusion-Robust 3D Hand Mesh Estimation Network. In *CVPR*, 2022. 1, 2, 6, 7, 8

[13] X. Chen, Y. Liu, Y. Dong, X. Zhang, C. Ma, Y. Xiong, Y. Zhang, and X. Guo. MobRecon: Mobile-Friendly Hand Mesh Reconstruction from Monocular Image. In *CVPR*, 2022. 1

[14] X. Tang, T. Wang, and C. Fu. Towards accurate alignment in real-time 3d hand-mesh reconstruction. In *ICCV*, 2021. 1, 2

[15] K. Lin, L. Wang, and Z. Liu. End-to-end human pose and mesh reconstruction with transformers. In *CVPR*, 2021. 1, 2, 6

[16] Y. Chen, Z. Tu, D. Kang, L. Bao, Y. Zhang, X. Zhe, R. Chen, and J. Yuan. Model-based 3d hand reconstruction via self-supervised learning. In *CVPR*, 2021. 1, 2

[17] H. Chen, P. Wang, F. Wang, W. Tian, L. Xiong, and H. Li. EPro-PnP: Generalized End-to-End Probabilistic Perspective-n-Points for Monocular Object Pose Estimation. In *CVPR*, 2022. 1

[18] Y. He, H. Huang, H. Fan, Q. Chen, and J. Sun. Ffb6d: A full flow bidirectional fusion network for 6d pose estimation. In *CVPR*, 2021. 1

[19] G. Wang, F. Manhardt, F. Tombari, and X. Ji. Gdr-net: Geometry-guided direct regression network for monocular 6d object pose estimation. In *CVPR*, 2021. 1

[20] B. Doosti, S. Naha, M. Mirbagheri, and D. all. Hopenet: A graph-based model for hand-object pose estimation. In *CVPR*, 2020. 2

[21] Z. Cao, I. Radosavovic, A. Kanazawa, and J. Malik. Reconstructing hand-object interactions in the wild. In *ICCV*, 2021. 2

[22] Y. Hasson, B. Tekin, F. Bogo, I. Laptev, M. Pollefeys, and C. Schmid. Leveraging photometric consistency over time for sparsely supervised hand-object reconstruction. In *CVPR*, 2020. 1, 2, 3, 6

[23] Y. Hasson, G. Varol, D. Tzionas, I. Kalevatykh, M. Black, I. Laptev, and C. Schmid. Learning joint reconstruction of hands and manipulated objects. In *CVPR*, 2019. 1, 2, 3, 6

[24] Y. Hasson, G. Varol, C. Schmid, and I. Laptev. Towards unconstrained joint hand-object reconstruction from RGB videos. In *3DV*, 2021. 2

[25] T. Tse, Z. Zhang, K. Kim, A. Leonardis, F. Zheng, and H. Chang. S ˆ2 Contact: Graph-Based Network for 3D Hand-Object Contact Estimation with Semi-supervised Learning. In *ECCV*, 2022. 2, 3

[26] Z. Chen, Y. Hasson, C. Schmid, and I. Laptev. AlignSDF: Pose-Aligned Signed Distance Fields for Hand-Object Reconstruction. In *ECCV*, 2022. 1

[27] S. Hampali, S. Sarkar, M. Rad, and V. Lepetit. Keypoint Transformer: Solving Joint Identification in Challenging Hands and Object Interactions for Accurate 3D Pose Estimation. In *CVPR*, 2022. 1, 2, 3, 6

[28] L. Yang, K. Li, X. Zhan, J. Lv, W. Xu, J. Li, and C. Lu. ArtiBoost: Boosting Articulated 3D Hand-Object Pose Estimation via Online Exploration and Synthesis. In *CVPR*, 2022. 1, 6

[29] H. Choi, G. Moon, and K. Lee. Pose2mesh: Graph convolutional network for 3d human pose and mesh recovery from a 2d human pose. In *ECCV*, 2020. 2, 6

[30] G. Moon, and K. Lee. I2l-meshnet: Image-to-lixel prediction network for accurate 3d human pose and mesh estimation from a single rgb image. In *ECCV*, 2020. 6

[31] J. Romero, D. Tzionas, and M. Black. Embodied hands: modeling and capturing hands and bodies together. In *TOG*, 2017. 2, 3, 5

[32] A. Spurr, U. Iqbal, P. Molchanov, O. Hilliges, and J. Kautz. Weakly supervised 3d hand pose estimation via biomechanical constraints. In *ECCV*, 2020. 2, 6

[33] T. Piumsomboon, A. Clark, M. Billinghurst, and A. Cockburn. User-defined gestures for augmented reality. In *IFIP Conference on Human-Computer Interaction*, 2013. 1

[34] W. Hürst, and C. Van Wezel. Gesture-based interaction via finger tracking for mobile augmented reality. In *Multimedia Tools and Applications*, 2013. 1

[35] A. a, K. VanWyk, W. Yang, J. Liang, Y. Chao, Q. Wan, S. Birchfield, N. Ratliff, and D. Fox. Dexpilot: Vision-based teleoperation of dexterous robotic hand-arm system. In *ICRA*, 2020. 1

[36] F. Mueller, M. Davis, F. Bernard, O. Sotnychenko, M. Verschoor, M. Otaduy, D. Casas, and C. Theobalt. Real-time pose and shape reconstruction of two interacting hands with a single depth camera. In *TOG*, 2019. 1

[37] V. Lepetit, F. Moreno-Noguer, and P. Fua. Epnp: An accurate o (n) solution to the pnp problem. In *International journal of computer vision*, 2009. 5

[38] U. Iqbal, P. Molchanov, T. Gall, and J. Kautz. Hand pose estimation via latent 2.5 d heatmap regression. In *ECCV*, 2018. 2

[39] M. Li, Y. Gao, and N. Sang. Exploiting learnable joint groups for hand pose estimation. In *AAAI*, 2021. 2

[40] Hodaň, Tomáš and Matas, Jiří and Obdržálek, Štěpán. On evaluation of 6D object pose estimation. In *ECCV Workshops*, 2016. 6

[41] L. Yang, X. Zhan, K. Li, W. Xu, J. Li, and C. Lu. Cpf: Learning a contact potential field to model the hand-object interaction. In *ICCV*, 2021. 2

[42] P. Grady, C. Tang, C. Twigg, M. Vo, S. Brahmbhatt, and C. Kemp. Contactopt: Optimizing contact to improve grasps. In *CVPR*, 2021. 2, 3

[43] B. Tekin, F. Bogo, and M. Pollefeys. H+ o: Unified egocentric recognition of 3d hand-object poses and interactions. In *CVPR*, 2019. 2, 3

[44] L. Yang, J. Li, W. Xu, Y. Diao, and C. Lu. Bihand: Recovering hand mesh with multi-stage bisected hourglass networks. In *arXiv preprint arXiv:2008.05079*, 2020. 2

[45] K. Lin, L. Wang, and Z. Liu. Mesh graphormer. In *ICCV*, 2021. 2

[46] X. Zhang, H. Huang, J. Tan, H. Xu, C. Yang, G. Peng, L. Wang, and J. Liu. Hand image understanding via deep multi-task learning. In *ICCV*, 2021. 2

[47] A. Boukhayma, R. Bem, and P. Torr. 3d hand shape and pose from images in the wild. In *CVPR*, 2019. 2