

# Direction-Aware Hybrid Representation Learning for 3D Hand Pose and Shape Estimation

Shiyong Liu<sup>1</sup> Zhihao Li<sup>1</sup> Xiao Tang<sup>1</sup> Jianzhuang Liu<sup>2</sup>

<sup>1</sup>Huawei Noah's Ark Lab

<sup>2</sup>Shenzhen Institute of Advanced Technology

{liushiyong3, zhihao.li, tangxiaol2}@huawei.com, jz.liu@siat.ac.cn

## Abstract

*Most model-based 3D hand pose and shape estimation methods directly regress the parametric model parameters from an image to obtain 3D joints under weak supervision. However, these methods involve solving a complex optimization problem with many local minima, making training difficult. To address this challenge, we propose learning direction-aware hybrid features (DaHyF) that fuse implicit image features and explicit 2D joint coordinate features. This fusion is enhanced by the pixel direction information in the camera coordinate system to estimate pose, shape, and camera viewpoint. Our method directly predicts 3D hand poses with DaHyF representation and reduces jittering during motion capture using prediction confidence based on contrastive learning. We evaluate our method on the FreiHAND dataset and show that it outperforms existing state-of-the-art methods by more than 33% in accuracy. DaHyF also achieves the top ranking on both the HO3Dv2 and HO3Dv3 leaderboards for the metric of Mean Joint Error (after scale and translation alignment). Compared to the second-best results, the largest improvement observed is 10%. We also demonstrate its effectiveness in real-time motion capture scenarios with hand position variability, occlusion, and motion blur.*

## 1. Introduction

Estimating 3D hand pose and shape from monocular RGB images or videos is a critical research area in computer vision and graphics. It enables human-machine and human-environment interactions for various scenarios such as AR/VR, HCI and digital human. However, it is challenging due to depth ambiguity, motion blur, severe occlusion, low resolution, etc.

There are two categories of methods for 3D hand pose estimation: model-based and model-free. Model-based methods [24, 35, 59, 60] use parametric hand models as

the proxy representation. Model-free methods [4, 17, 30, 44, 58] directly predict the 3D coordinates of hand joints. The advantage of the former over the latter is that they can use prior information from parametric models to avoid unreasonable prediction parameters. However, predicting 3D joint rotation from 2D RGB images is a highly non-linear process for neural networks, which causes pixel misalignment in model-based methods.

Some methods propose to involve 2D joint coordinates to improve 3D regression accuracy and 2D-3D consistency [2, 7, 56]. However, most of them are not end-to-end because they require pre-training a 2D hand keypoint detector on a separate dataset. The accuracy of these methods depends heavily on the performance of the pre-trained detector and the quality of the dataset. Furthermore, their 2D joint features are mainly expressed as heatmaps which have large quantization errors in small and low-resolution targets. In order to reduce quantization errors, a regression-based end-to-end high-precision 2D keypoint detector that can be jointly optimized plays an important role in improving overall motion capture accuracy.

Directly regressing joint coordinates is a translation-dependent task. The CoordConv [31] method is proposed to deal with the translation invariance problem in the coordinate regression task with convolutional networks. CoordConv improves detection accuracy by adding corresponding channels to the input feature map of the convolution. These channels represent the coordinates of the pixels in the feature map, allowing the convolution learning process to perceive coordinates to some extent. However, Top-down 3D hand motion capture is a two-stage task. The hand is center-cropped and resized before being sent to the network. The coordinate encoding method of CoordConv cannot deal with 3D ambiguity. Encoding under the feature map coordinate system loses the hand position information in the full frame of the video, and does not improve the 3D estimation much.

A common problem in monocular video motion capture is the smoothness between frames. Since the hand targets

have a relatively large range of motion in the entire frame, there may be problems with low resolution and motion blur. Time-domain filtering can effectively improve the smoothness between frames. However, the tracker may lose the target, resulting in poor-quality hand-image input to the network, which may cause false positives to the filter. A better idea is to get the confidence of each frame of the motion capture result. For those whose confidence is lower than a threshold, the result of high confidence in the previous frame can be used as the replacement. This can improve filter performance and alleviate the jittering and flipping problem.

Motivated by the above observations, we present an end-to-end representation learning with direction-aware hybrid features (DaHyF) to improve the accuracy of 3D hand pose and shape estimation. The direction-aware hybrid features are a fusion of implicit image features and explicit 2D joint coordinate features. To reduce quantization errors and obtain high accuracy 2D joint coordinates, we design an end-to-end sub-pixel coordinate predictor that can be jointly optimized. To alleviate the problem of weak spatial perception ability caused by the translation invariance of convolutional networks and to improve the model’s ability to regress 2D coordinates and 3D rotations, we develop a global direction map module. To avoid jittering and flipping caused by false positives, we propose a scheme of motion capture confidence calculation based on contrastive learning. The end-to-end training pipeline enables joint optimization using various datasets to improve the accuracy of each module and achieve more pixel-aligned 3D pose and shape estimation without an additional detector. Our method significantly outperforms current state-of-the-art methods on the FreiHAND [59], HO3Dv2 [12] and HO3Dv3 [13] datasets according to various evaluation metrics.

Our main contributions are summarized as follows:

- We design a 2D+3D end-to-end joint optimization framework with hybrid implicit image features and explicit joint coordinate features. Our sub-pixel coordinate classifier costs fewer resources and has smaller quantization errors compared to heatmap-based methods. The end-to-end pipeline allows us to jointly optimize 2D and 3D coordinates, building a complete contrastive learning strategy to generate prediction confidence and achieve mutual promotion between 2D and 3D pose estimation.
- We propose a global direction map to enhance the spatial perception of convolutional networks. This module avoids the translation-agnostic problem and significantly improves both 2D and 3D pose estimation performance.
- We present a motion capture confidence calculation scheme based on contrastive learning. It effectively utilizes non-hand patch information to reduce false positives and improve the robustness and smoothness in video motion capture.

## 2. Related Work

### 2.1. 3D Hand Pose and Shape Estimation

3D hand pose and shape estimation methods can be generally divided into two types: model-based and model-free. Model-based methods typically use a parameterized hand model [1, 20, 40, 49] as a differentiable layer in a neural network to estimate shape and pose and map them to a triangle mesh and joint coordinates. Pose and shape are weakly supervised by supervising joint coordinates and the mask of rendered results. Since the parameterized model contains prior information about hand structure, this approach has the advantages of reducing the number of parameters, improving robustness, and reducing artifacts. However, there are also disadvantages: the parameters of the parameterized model are weakly supervised, resulting in difficulties in achieving pixel-aligned results, and the optimization process is more likely to fall into local minima [26]. Model-free methods [8, 23, 34] do not require a predefined hand model and attempt to learn a mapping from the input image or depth data to pre-defined kinematic joints via an end-to-end network. They directly regress 2D/3D keypoint coordinates and calculate joint rotations using inverse kinematics. These methods generally achieve more pixel-aligned results compared to model-based methods, but they may suffer from noise and occlusion, resulting in artifacts due to the lack of hand prior information.

Our method combines the advantages of both approaches by additionally fusing coordinate features with image features to assist in regressing parameters of the parameterized hand model

### 2.2. 2D Hand Keypoint Estimation

2D hand keypoint estimation aims to locate the 2D coordinates of hand keypoints from images. This is usually done in a top-down manner because hand targets are generally small in the whole images. The top-down paradigm employs a two-step procedure that first detects hand bounding boxes and then performs single hand keypoint estimation for each bounding box. Top-down approaches can be categorized into regression-based [25, 39, 50] and heatmap-based [46, 51, 52, 54]. Regression-based methods directly regress the keypoint coordinates from the image, which are efficient and show promising potential in real-time applications, but they fail to provide a probability distribution for multiple candidate positions. To overcome the shortcomings of direct coordinate regression and make coordinates more suitable for regression by a convolutional neural network, heatmap-based joint representations have been proposed [38], which output the keypoint positions as the peak values of heatmaps or confidence maps. The advantage of this type of methods is that it can provide a probability distribution for multiple candidate positions, but due to the

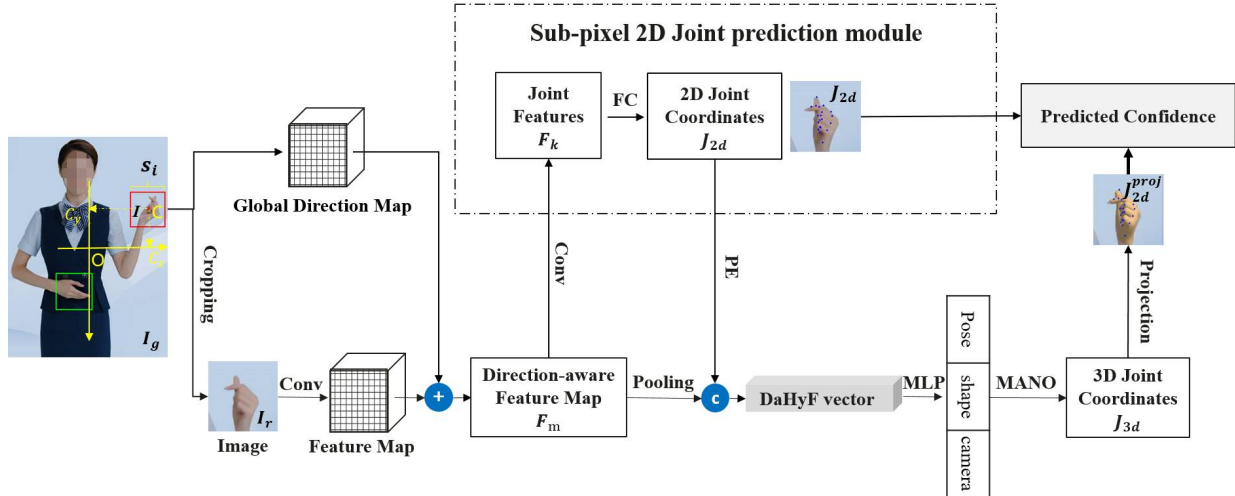


Figure 1. **Overview of DaHyF.** A hand region is cropped, resized and encoded as a local implicit feature map, which is then fused with a global direction map calculated from the hand bounding box. The fused features are used to detect 2D hand keypoints  $J_{2d}$  in sub-pixel accuracy, which are positionally encoded (PE) with sinusoidal signals to build a joint feature vector. The pooled implicit  $F_m$  and the encoded explicit  $J_{2d}$  are concatenated to form our direction-aware hybrid feature (DaHyF) vector. This DaHyF vector is used to regress hand pose, shape, and camera parameters. Finally, 3D keypoints are obtained based on the MANO model [40] and projected to 2D coordinates  $J_{2d}^{proj}$  for confidence computation with  $J_{2d}$ .

quantization errors, heatmap-based methods do not perform well in low-resolution scenarios [27], especially for small and severe motion blur targets such as hands.

Our method is regression-based and uses a sub-pixel coordinate classifier to reduce quantization errors.

### 2.3. 3D Pose Confidence

Many works have proposed predicting pose confidence to promote the robustness and accuracy of networks. [3] uses a 2D pose detector to provide a confidence value for each joint, with undetected joints having a confidence of zero. [48] builds a multi-view 2D part confidence map to track 3D skeletons in the presence of missing detections, substantial occlusions, and large calibration errors. [43] generates confidence-scored 3D proposals for several body joints by reprojecting the classification result and finding local modes. [11] targets the problem of inaccurate confidence values predicted by CNNs and takes pairs of pose masks rendered from a 3D model and crops regions in the original image as input to calibrate the confidence scores of the pose proposals. Most of these methods operate at the granularity of each joint, with confidence used to find the highest scoring result among many candidates or to judge the probability that the current joint is occluded.

In video motion capture, the accuracy of individual joints is important, but the overall hand pose has the greatest impact on the visual effect because it affects the smoothness of the motion capture process and whether there are sudden jittering or other abnormal mutations. Our method is

designed to take into account all hand joints and output the confidence of the current frame’s motion capture result to alleviate jitter and flip caused by false positives, which can improve the robustness and smoothness of video motion capture.

## 3. Method

This paper presents a single-view 2D+3D end-to-end joint optimization framework with direction-aware hybrid features for hand motion capture. It provides comprehensive spatial information for the model-based approach and improves the pixel-alignment of hand motion capture. To achieve this, as shown in Figure 1, the framework has three components: (1) a global direction map, which encodes global direction information into an implicit image feature map to boost spatial perception capabilities of convolution layers, (2) sub-pixel joint coordinate classification, which predicts sub-pixel coordinate for each joint to reduce the quantization error, and (3) Positional Encoding (PE), which maps each coordinate into a high-dimensional space. The encoded 2D joint coordinates are fused with the implicit image features, which are then fed to the regressor to estimate MANO [40] parameters. We also present a scheme for motion capture confidence calculation based on contrastive learning, which can improve the robustness and smoothness of the output video.

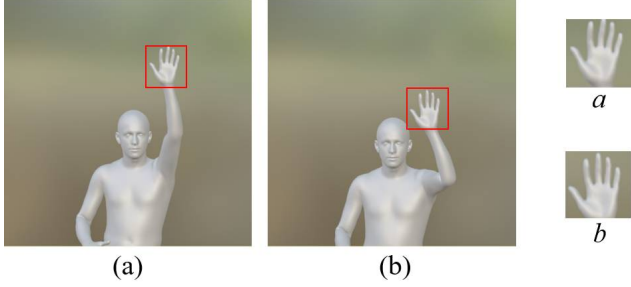


Figure 2. The input image features of  $a$  and  $b$  are very similar after cropping and resizing the hand patches. From their local direction maps, we can see that they are identical under the input image feature coordinate space. Thus, the local direction maps do not help CNN distinguish  $a$  from  $b$ , and it is difficult to regress the global 3D pose information of the hands with respect to the camera coordinate system.

### 3.1. Hand Encoding

Given an image or a video frame  $I_g \in R^{H \times W}$  containing hands, we first crop square hand patches  $I \in R^{s_i \times s_i \times 3}$  and resize them to  $s_p \times s_p$ . Here,  $s_i$  is the patch size and  $s_p$  is the network input size equals to 224. We also flip all the left hand patches to the right. Then, we use convolution layers to extract shared low-level semantic features. Thus, the hand is represented by an implicit image feature map  $F$ .

### 3.2. Direction Map

Convolutional neural networks (CNNs) are very successful in various visual tasks due to their translation invariance. However, this advantage becomes a defect in tasks that involve translation dependence, such as coordinate modeling, and potentially affects the final model performance [31]. One way to improve the accuracy of coordinate regression is to add a direction map to the feature map, which enables the latter convolution process to perceive the spatial information of the features. The traditional approach generally builds the direction map under the local coordinate space of image features. We define  $s_f$  as the size of the image feature map  $F$ . The local direction map  $L_m$  has the same size as  $F$ . Each channel contains either row coordinate  $i \in [0, s_f]$  or column coordinate  $j \in [0, s_f]$  under the local space of  $F$ . The origin is located at the upper left corner of  $F$ . The fused feature map is the concatenation of  $F$  and  $L_m$ .

In our experiments, we find that the local direction encoding improves the estimation of 2D joints, but it does not enhance the 3D joint accuracy due to the ambiguity of the camera pose in 3D space. Compared with the body, the hand movement in the video is more flexible. As illustrated in Figure 2, the hand can appear in different positions of the frame, after cropping and resizing, the calculated local direction maps are the same, and they cannot help CNN dis-

tinguish them. Therefore, it is hard to regress the 3D information of these hands with respect to the camera coordinate system.

To address this issue, we propose the global direction map, as illustrated in Figure 3. First, given a pixel coordinate  $P_f^i$  in the image feature map, we calculate its corresponding pixel coordinate  $P_l^i$  under the resized hand patch:

$$P_l^i = P_f^i \times sc_p + \frac{sc_p}{2}, \quad (1)$$

$$sc_p = \frac{s_p}{s_f}. \quad (2)$$

Then, we calculate the corresponding global coordinate  $P_g^i$  under the frame space:

$$P_g^i = P_l^i \times sc_o + P_{ulc} - O, \quad (3)$$

$$sc_o = \frac{s_i}{s_p}, \quad (4)$$

$$O = \left( \frac{W}{2}, \frac{H}{2} \right), \quad (5)$$

where  $P_{ulc}$  is the upper left corner's coordinate of the hand patch  $I$  in the global frame space, and  $O$  is the origin located at the center of the frame.

To better estimate the joint rotation angle, we normalize the global coordinate  $P_g^i$  into a direction vector  $P_d^i$  in the camera coordinate system based on the focal length  $f$ , i.e.,  $P_d^i = P_g^i / f$ , similar to the definition of the direction vector in NeRF [33]. In practice, if  $f$  is unknown, we set it to  $\sqrt{H^2 + W^2}$ , where  $H$  and  $W$  are the height and width of the frame.

Give the image feature map with all pixel coordinates  $\{P_f^i\}$ , we can obtain all their corresponding direction vectors  $\{P_d^i\}$ . Let  $P_d^i = (x_d^i, y_d^i)$ . We construct the global direction map as follows: We first form two channels: one with all the  $\{x_d^i\}$  and the other with all the  $\{y_d^i\}$ . Then we copy them to form the global direction map such that it has the same channel number as that of the image feature map  $F$  (see Figure 3).

### 3.3. Joint Feature Fusion

**2D Joint Estimation.** The resolution of hand patches is usually low. This makes the traditional Gaussian heatmap-based methods unsuitable for coordinate regression due to quantization error. Therefore, we draw inspiration from SIMCC [27] and transform the coordinate estimation into a classification task. This can improve the localization accuracy of hand keypoints under low resolution. We apply a series of convolution operations on the image feature  $F_m$  (see Figure 1) to obtain the representation  $F_k \in R^{21 \times 56 \times 56}$  with 21 hand keypoints. Define  $n_x$  (or  $n_y$ ) as the number of class labels, where  $n_x = s_p \times s$ ,  $n_y = s_p \times s$ ,

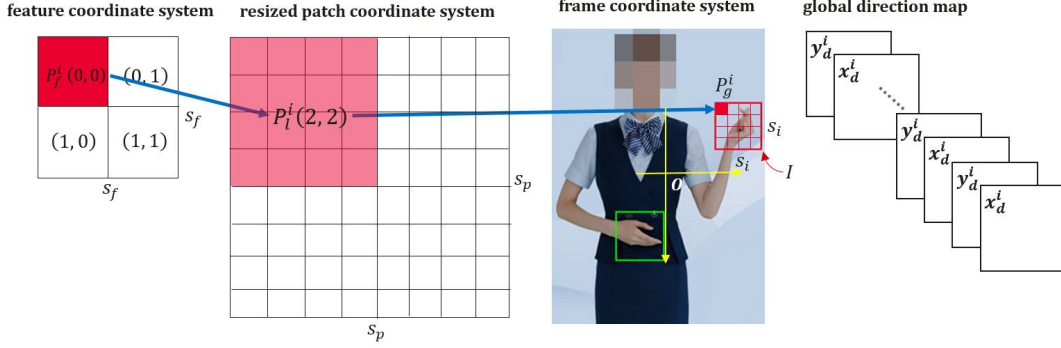


Figure 3. **Global direction map construction.** First, we calculate  $P_f^i$  from  $P_f^i$ , then  $P_g^i$  from  $P_f^i$  and  $P_d^i$  (not shown here) from  $P_g^i$ , and finally form the global feature map from  $P_d^i = (x_d^i, y_d^i)$ .

and  $s \geq 1$  is a pre-defined scale for controlling the quantization error to improve the sub-pixel positioning accuracy ( $s$  is set to 3 in this work). The numbers of bins for the horizontal and vertical axes are  $n_x$  and  $n_y$ , respectively. The horizontal coordinate classifier and the vertical coordinate classifier predict the  $k$ -th keypoint  $\{f_x^k, f_y^k\}$ , where  $k = 1, 2, \dots, 21$ , and  $f_x^k, f_y^k \in R^{n_x}$ . Let the ground-truth labels be  $\{\hat{J}_x^k, \hat{J}_y^k\}$ . We train the network by smoothing the classification labels with 1D Gaussian and minimizing the KL divergence between  $\{\bar{J}_x^k, \bar{J}_y^k\}$  and  $\{f_x^k, f_y^k\}$ , where  $\{\bar{J}_x^k, \bar{J}_y^k\}$  is the smoothed result of  $\{\hat{J}_x^k, \hat{J}_y^k\}$ . Then we convert the network output  $\{f_x^k, f_y^k\}$  to  $J_{2d} \in R^{2 \times 21}$  through soft-argmax to obtain the predicted 2D keypoint coordinates:  $J_{2d} = \{\frac{1}{s} \text{soft-argmax}(f_x^k), \frac{1}{s} \text{soft-argmax}(f_y^k)\}$ .

**2D-3D Joint Features Fusion.** To better combine the 2D joints with the direction-aware feature map  $F_m$ , we use a position encoding (PE) method similar to NeRF [33], which maps the coordinates to a high dimensional space and enables our regressor to more easily approximate higher frequency information. The encoding result is as follows:

$$\{\gamma_x^k(\mu_x^k), \gamma_y^k(\mu_y^k)\}, \quad \{\mu_x^k, \mu_y^k\} = \frac{(J_{2d} - \{\frac{s_p}{2}, \frac{s_p}{2}\})}{f}, \quad (6)$$

$$\begin{aligned} \gamma_x^k(p) &= \gamma_y^k(p) = \\ &(\sin(2^1 \pi p), \cos(2^1 \pi p), \dots, \sin(2^L \pi p), \cos(2^L \pi p)), \quad (7) \end{aligned}$$

where  $L$  is set to 4 in this work. So the encoded coordinates are  $\{\gamma_x^k(\mu_x^k), \gamma_y^k(\mu_y^k)\} \in R^{2 \times 21 \times (2L)}$ , which are then converted to a vector  $V_{PE} \in R^{336}$ . We also perform feature map pooling on  $F_m$  to obtain another vector  $V_{F_m}$ . Finally, we concatenate  $V_{PE}$  and  $V_{F_m}$  to obtain the DaHyF feature vector (see Figure 1), which is fed to an MLP to regress the hand pose and shape, and camera parameters.

### 3.4. Contrastive Learning for Pose Confidence

The contrastive learning aims to reduce the impact of non-hand regions on hand pose estimation. We use the regressor (MLP in Figure 1) to generate the hand pose  $\theta \in R^{16 \times 3}$  (represented in axis angle), shape  $\beta \in R^{10}$ , and weak-perspective projection camera parameters  $P_{weak}(s, tx, ty) \in R^3$  with respect to the cropped hand patch  $I$ . Then using MANO with  $\theta$  and  $\beta$  as the input, we obtain the 3D joint coordinates  $J_{3d}$ , which are projected to 2D joint coordinates  $J_{2d}^{proj}$  by the global perspective projection parameters that are obtained based on  $P_{weak}$  [25].

To measure the similarity between  $J_{2d}$  and  $J_{2d}^{proj}$ , we use cosine similarity as the contrastive learning criterion. We first normalize  $J_{2d}$  and  $J_{2d}^{proj}$  by:

$$v_{2d} = (J_{2d} \times (\frac{s_i}{s_p}) - (\frac{s_i}{2}, \frac{s_i}{2}))/s_i, \quad (8)$$

$$v_{2d}^{proj} = (J_{2d}^{proj} - C)/s_i, \quad (9)$$

where  $C$  is the center coordinate of the hand patch  $I$  under the coordinate system of the frame  $I_g$  (see Figure 1). Then all the coordinates in  $v_{2d}$  (or  $v_{2d}^{proj}$ ) are concatenated to form a vector  $\bar{v}_{2d}$  (or  $\bar{v}_{2d}^{proj}$ ). Finally,  $\bar{v}_{2d}$  and  $\bar{v}_{2d}^{proj}$  are used to compute their cosine similarity.

When the patch  $I$  contains the hand,  $\bar{v}_{2d}$  and  $\bar{v}_{2d}^{proj}$  are considered as a pair of positive samples. We also randomly crop patches from the frame, and when a patch does not contain the hand, the resulting  $\bar{v}_{2d}$  and  $\bar{v}_{2d}^{proj}$  are considered as a pair of negative samples. Our goal is to make a positive pair as close as possible and make a negative pair as far away as possible. Figure 4 shows the main procedure of this contrastive learning. During inference, we use this cosine similarity value as the confidence measure for tidying the motion capture result. A patch whose confidence is lower than a threshold is considered as a false positive, and its hand pose parameters are replaced by those of its previous nearest hand patch.

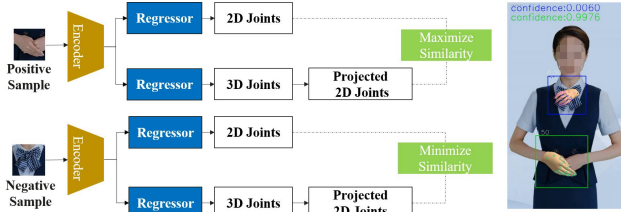


Figure 4. Our contrastive learning procedure.

### 3.5. Training Objective

The overall training loss, denoted as  $\mathcal{L}$ , consists of two components:  $\mathcal{L}_{backbone}$  for backbone training and  $\mathcal{L}_c$  for contrastive learning. The former includes 5 sub-losses: 2D branch loss ( $\mathcal{L}_{2d}$ ), 3D loss ( $\mathcal{L}_{3d}$ ), projected 2D loss ( $\mathcal{L}_{2d}^p$ ), MANO loss ( $\mathcal{L}_m$ ), and bone loss ( $\mathcal{L}_b$ ).

$$\mathcal{L} = \mathcal{L}_{backbone} + \mathcal{L}_c. \quad (10)$$

When 3D labels are available, we optionally apply  $\mathcal{L}_m$  and  $\mathcal{L}_{3d}$  and use the homoscedastic uncertainty strategy [56] to adaptively learn the weights of all sub-losses.  $\mathcal{L}_{backbone}$  is defined as:

$$\mathcal{L}_{backbone} = \frac{\mathcal{L}_{2d}}{\sigma_{2d}^2} + \frac{\mathcal{L}_{3d}}{\sigma_{3d}^2} + \frac{\mathcal{L}_{2d}^p}{(\sigma_{2d}^p)^2} + \frac{\mathcal{L}_m}{\sigma_m^2} + \frac{\mathcal{L}_b}{\sigma_b^2}, \quad (11)$$

where  $\sigma_{2d}$ ,  $\sigma_{3d}$ ,  $\sigma_{2d}^p$ ,  $\sigma_m$ , and  $\sigma_b$  are learned by the strategy automatically. We use the L1 loss for  $\mathcal{L}_{2d}$ ,  $\mathcal{L}_{3d}$ , and  $\mathcal{L}_{2d}^p$ , and the L2 loss for  $\mathcal{L}_m$  and  $\mathcal{L}_c$ . The bone loss  $\mathcal{L}_b$  refers to [24].

## 4. Experiments

### 4.1. Implementation Details

We use HRNet-W64 [46] as our backbone to extract the feature map  $F$  from hand patches. The backbone is initialized with ImageNet [9] pre-trained weights to leverage knowledge learned from large-scale image classification. Our model is trained with Adam optimizer [21] on 4 GPUs with a batch size of 128. The hand detector is from [32] and a detected square hand patch is cropped and resized to  $224 \times 224$ . We use an initial learning rate of  $5 \times 10^{-5}$  and reduce it by a factor of 10 after 250 epochs out of 500 epochs. We only perform the contrastive learning in the last 100 epochs. To augment our training data, we apply random rotation, scaling, and cropping [42]. These techniques increase the generalization and robustness of our model to different hand poses and orientations.

### 4.2. Datasets and Evaluation Metrics

We evaluate our method using three popular 3D hand reconstruction datasets: FreiHAND [59], HO3Dv2 [12] and

HO3Dv3 [13]. We achieve the top ranking on both the HO3Dv2 and HO3Dv3 leaderboards for the metric of Mean Joint Error (after scale and translation alignment).<sup>1, 2</sup>

**FreiHAND.** FreiHAND [59] is a large-scale mixed real-world and synthetic dataset based on the MANO [40] model. It contains 32560 training samples and 3960 test samples of people performing different hand movements. This dataset is suitable for evaluating the accuracy and realism of our method to reconstruct realistic hand meshes from RGB images.

**HO3D.** HO3Dv2 [12] is a 3D hand-object dataset that contains 66,034 training samples and 11,524 evaluation samples. HO3Dv3 [13] has more accurate annotations and more data including 83,325 training images and 20,137 testing images. Evaluation on HO3Dv2 and HO3Dv3 are performed through an online submission website

**HanCo.** We also use HanCo [60], a dataset that contains 1518 short video clips captured by 8 calibrated and synchronized cameras, for training our model. It consists of 860304 individual frames and can be seen as an extended version of FreiHAND [59]. This dataset contains many non-hand frames, which are suitable for being negative samples for our contrastive learning.

**Evaluation Metrics.** We evaluate our method using 3D joint metrics and 2D joint metrics. For 3D joint metrics, we adopt procrustes-aligned mean per joint position error (PA-MPJPE), procrustes-aligned mean per vertex error (PA-MPVPE), and mean per joint position error (MPJPE). For conciseness, PA-MPJPE and PA-MPVPE are abbreviated as PJ and PV, respectively. They measure the Euclidean distances (in millimeter) of 3D joint or 3D mesh coordinates between the predictions and ground truth [19, 22, 23, 57]. Additionally, we calculate the F-Score at specific distance thresholds, denoted as F@5 and F@15, which correspond to thresholds of 5mm and 15mm, respectively. This score represents the harmonic mean of recall and precision between two meshes with respect to the given threshold [6]. For 2D joint metrics, we adopt the average endpoint error (EPE) [45] in pixels.

### 4.3. Comparison with State-of-the-Art Methods

We compare our method with current state-of-the-art (SOTA) works on 3D hand pose estimation. We use the CLIFF annotator [28] to provide MANO [40] pseudo-GT for the FreiHAND [59] dataset. This allows us to train our model with more realistic and accuracy hand shapes and poses. We evaluate our method against model-based [6, 7, 10, 18, 24, 28, 35, 36, 41, 47, 55, 59] and model-free [8, 29, 30, 34] approaches on FreiHAND. As shown in Table 1, our method achieves significant improvements over the current SOTA Mesh Graphormer [30] in all met-

<sup>1</sup>HO3Dv2: <https://codalab.lisn.upsaclay.fr/competitions/4318#results>

<sup>2</sup>HO3Dv3: <https://codalab.lisn.upsaclay.fr/competitions/4393#results>

Table 1. Performance comparison with SOTA methods on the FreiHAND [59] test set.

Method	Venue	PJ ↓	PV ↓	F@5 ↑	F@15 ↑
<b>* Model-free</b>					
Graphormer [30]	ICCV'21	5.9	6.0	76.4	98.6
METRO [29]	CVPR'21	6.3	6.5	73.1	98.4
I2L-MeshNet [34]	ECCV'20	7.4	7.6	68.1	97.3
Pose2Mesh [8]	ECCV'20	7.7	7.8	67.4	96.9
<b>* Model-based</b>					
MANO CNN [59]	ICCV'19	11.0	10.9	51.6	93.4
FrankMocap [41]	ICCV'21	9.2	11.6	55.3	95.1
PIXIE [10]	3DV'21	12.2	11.8	46.8	91.9
Tang et al. [47]	CVPR'21	6.7	6.7	72.4	98.1
Moon et al. [24]	CVPR'20	8.4	8.6	61.4	96.6
Hand4Whole [35]	CVPR'22	7.7	7.7	66.4	97.1
CLIFF [28]	ECCV'22	6.8	6.6	-	-
PyMAF [55]	CVPR'21	7.5	7.7	67.1	97.4
MobRecon [6]	CVPR'22	6.9	7.2	69.4	97.9
RoboSMPLX [36]	NIPS'24	6.9	6.7	71.5	98.1
S <sup>2</sup> Hand [7]	CVPR'21	11.8	11.9	48.0	-
AMVUR [18]	CVPR'23	6.2	6.1	76.7	98.7
HaMeR [37]	CVPR'24	6.0	5.7	78.5	99.0
DaHyF (Ours)		<b>4.0</b>	<b>4.7</b>	<b>84.8</b>	<b>99.8</b>

Table 2. Performance comparison with SOTA methods on the HO3Dv2 test set.

Method	Venue	PJ ↓	PV ↓	F@5 ↑	F@15 ↑
ObMan [15]	CVPR'19	11.0	11.0	46.4	93.9
HO3D [12]	CVPR'20	10.7	10.6	50.6	94.2
I2UV-HandNet [5]	CVPR'21	9.9	10.1	50.0	94.3
MobRecon [6]	CVPR'22	9.2	9.4	53.8	95.7
METRO [29]	CVPR'21	10.4	11.1	48.4	94.6
S <sup>2</sup> Hand [7]	CVPR'21	11.4	11.2	45.0	93.0
AMVUR [18]	CVPR'23	8.3	8.2	60.8	96.5
DaHyF (Ours)		<b>8.0</b>	<b>8.1</b>	<b>61.2</b>	<b>97.6</b>

rics. This demonstrates the effectiveness of our method for 3D hand pose estimation.

As shown in Table 2 and Table 3, we also conduct evaluation on the HO3Dv2 [12] and HO3Dv3 [13] datasets, which are more challenging than FreiHAND [59] due to severe object occlusion. Our DaHyF, trained on FreiHAND [59] and with only three additional fine-tuning epochs on each of the HO3D datasets, achieves the first place on both the leaderboards for the Mean Joint Error metric (after scale and translation alignment), demonstrating its excellent generalizability.

Figure 5 shows qualitative results obtained by our method and Mesh Graphormer [30] on the FreiHAND test set, where the reconstructed hand meshes are rendered. We can see that our results are much more accurate and pixel-aligned with the hands in the images.

Table 3. Performance comparison with SOTA methods on the HO3Dv3 test set.

Method	Venue	PJ ↓	PV ↓	F@5 ↑	F@15 ↑
ArtiBoost [53]	CVPR'22	10.8	10.4	50.7	94.6
Keypoint Trans [14]	CVPR'22	10.9	-	-	-
AMVUR [18]	CVPR'23	8.7	8.3	59.3	96.4
S <sup>2</sup> Hand [7]	CVPR'21	11.5	11.1	44.8	93.2
DaHyF (Ours)		<b>7.5</b>	<b>7.5</b>	<b>63.7</b>	<b>97.4</b>

#### 4.4. Ablation Study

To evaluate the effectiveness of our proposed framework, we conduct an ablation study using the FreiHAND dataset. Initially, we establish a Baseline model, which solely comprises the lower branch as depicted in Figure 1, employing ResNet-50 as the backbone [16]. Subsequently, we incrementally incorporate different components into the Baseline model to construct five distinct models. The qualitative results obtained from these experiments are summarized in Table 4.

These results reveal several key findings. Firstly, the integration of components such as GDM (Global Direction Map),  $\mathcal{L}_c$  (contrastive learning loss), SJCP (Sub-Pixel Joint Coordinate Prediction Module), and PE (Position Encoding) significantly enhances the performance of the Baseline model. Particularly noteworthy is the observation that the inclusion of the proposed GDM component leads to notable performance improvements across all evaluation metrics when compared with the Baseline+LDM+SJCP (+  $\mathcal{L}_c$ ) and Baseline+GDM+SJCP (+  $\mathcal{L}_c$ ) configurations.

Furthermore, the comparison between the Baseline+GDM+Heatmap (+  $\mathcal{L}_c$ ) and Baseline+GDM+SJCP (+  $\mathcal{L}_c$ ) configurations highlights the superiority of SJCP over heatmap-based joint coordinate regression methods. This indicates the effectiveness of our sub-pixel joint coordinate classification approach in accurately predicting joint coordinates.

Ultimately, our final model, denoted as Baseline+GDM+SJCP+PE (+  $\mathcal{L}_c$ ), emerges as the top-performing configuration, demonstrating superior performance across all evaluated metrics. These findings underscore the efficacy of our proposed framework in enhancing the accuracy and robustness of hand motion capture systems.

## 5. Limitation and Conclusion

**Limitation.** Our method only considers single-hand movement without occlusions by two hands. If the hand detector fails to distinguish between left and right hands, our model may give wrong results.

**Conclusion.** We have presented a novel single-view

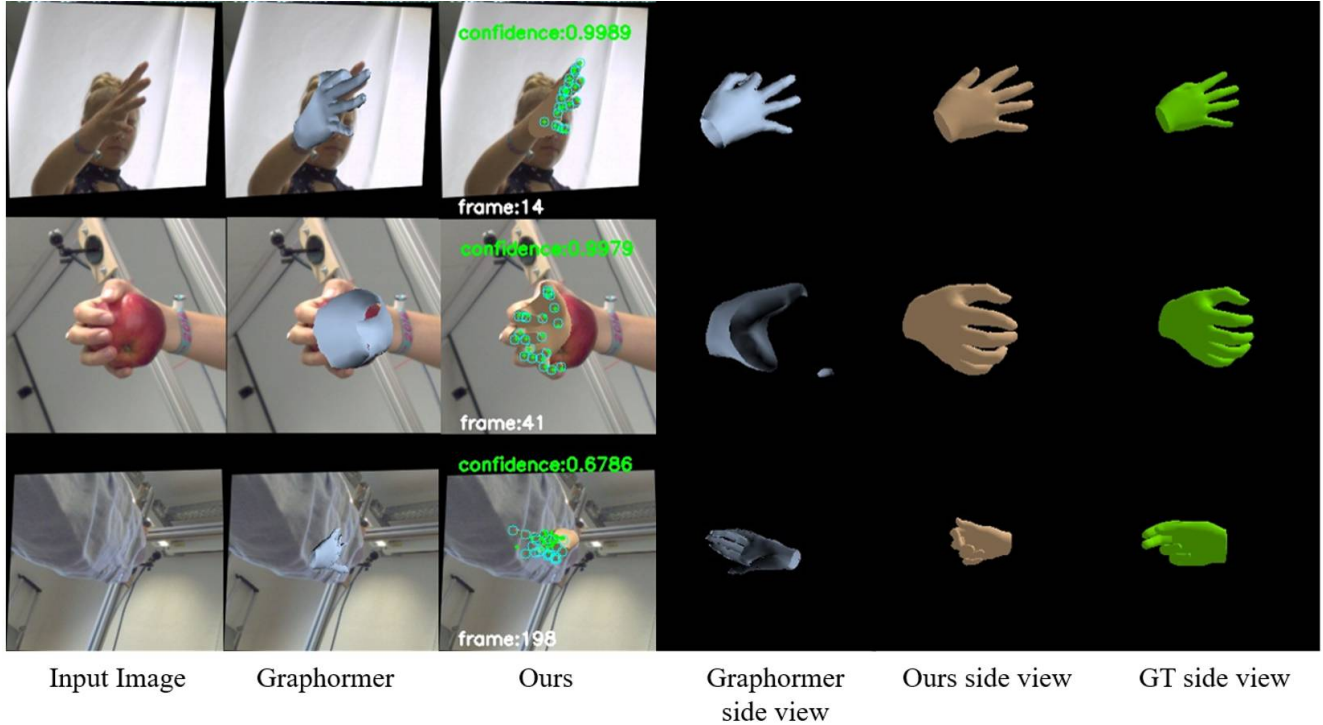


Figure 5. Qualitative comparison on the FreiHAND test set between our model and Mesh Graphormer. The last three columns visualise the results from novel views. Our results have higher accuracy and fewer artifacts in occluded scenes. Our model also provides confident scores for the pose estimation.

Table 4. Ablation studies of the components of our model. LDM: local direction map. GDM: global direction map. Heatmap: heatmap-based joint coordinate regression. SJCP: sub-pixel joint 2D coordinate prediction module. PE: position encoding.  $\mathcal{L}_c$ : contrastive learning loss.

	PA-MPJPE ↓	MPJPE ↓	EPE ( $J_{2d}^{proj}$ ) ↓	EPE ( $J_{2d}$ ) ↓
Baseline	8.91	18.14	7.37	-
Baseline + GDM	8.83	17.83	7.01	-
Baseline + GDM + Heatmap	8.40	17.30	6.80	8.10
Baseline + GDM + Heatmap + $\mathcal{L}_c$	8.20	16.99	6.33	7.36
Baseline + GDM + SJCP + $\mathcal{L}_c$	8.02	16.11	6.17	<b>6.95</b>
Baseline + LDM + SJCP + $\mathcal{L}_c$	9.59	22.7	10.03	9.06
Baseline + GDM + SJCP + PE + $\mathcal{L}_c$ (Ours)	<b>7.88</b>	<b>15.95</b>	<b>6.05</b>	7.83

2D+3D end-to-end joint optimization framework augmented with direction-aware hybrid features, aimed at enhancing the accuracy of hand motion capture. These direction-aware hybrid features are a blend of implicit image features and explicit 2D joint coordinate features, providing a comprehensive representation of hand motion.

To mitigate issues such as jittering and flipping induced by false positives, we have proposed a motion capture confidence calculation scheme based on contrastive learning. This approach helps enhance the robustness of our model

against erroneous detections.

Experiments on the FreiHAND dataset demonstrate the effectiveness of our method, revealing a significant improvement of more than 33% in accuracy compared to existing state-of-the-art techniques. Our model also achieves top ranking on both the HO3Dv2 and HO3Dv3 leaderboards for the metric of Mean Joint Error. These results underscore the potential of our framework to advance the field of hand motion capture, offering enhanced performance and reliability in real-world applications.

## References

- [1] Luca Ballan, Aparna Taneja, Jürgen Gall, Luc Van Gool, and Marc Pollefeys. Motion capture of hands in action using discriminative salient points. In *ECCV*, 2012. 2
- [2] Adnane Boukhayma, Rodrigo de Bem, and Philip HS Torr. 3d hand shape and pose from images in the wild. In *CVPR*, 2019. 1
- [3] Lewis Bridgeman, Marco Volino, Jean-Yves Guillemaut, and Adrian Hilton. Multi-person 3d pose estimation and tracking in sports. In *CVPR workshops*, 2019. 3
- [4] Yujun Cai, Lihao Ge, Jianfei Cai, and Junsong Yuan. Weakly-supervised 3d hand pose estimation from monocular rgb images. In *ECCV*, 2018. 1
- [5] Ping Chen, Yujin Chen, Dong Yang, Fangyin Wu, Qin Li, Qingpei Xia, and Yong Tan. I2uv-handnet: Image-to-uv prediction network for accurate and high-fidelity 3d hand mesh modeling. In *CVPR*, 2021. 7
- [6] Xingyu Chen, Yufeng Liu, Yajiao Dong, Xiong Zhang, Chongyang Ma, Yanmin Xiong, Yuan Zhang, and Xiaoyan Guo. Mobrecon: Mobile-friendly hand mesh reconstruction from monocular image. In *CVPR*, 2022. 6, 7
- [7] Yujin Chen, Zhigang Tu, Di Kang, Linchao Bao, Ying Zhang, Xuefei Zhe, Ruizhi Chen, and Junsong Yuan. Model-based 3d hand reconstruction via self-supervised learning. In *CVPR*, 2021. 1, 6, 7
- [8] 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. 2, 6, 7
- [9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVRP*, 2009. 6
- [10] Yao Feng, Vasileios Choutas, Timo Bolkart, Dimitrios Tzionas, and Michael J Black. Collaborative regression of expressive bodies using moderation. In *3DV*, 2021. 6, 7
- [11] Kartik Gupta, Lars Petersson, and Richard Hartley. Cullnet: Calibrated and pose aware confidence scores for object pose estimation. In *ICCV*, 2019. 3
- [12] 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, 7
- [13] Shreyas Hampali, Sayan Deb Sarkar, and Vincent Lepetit. Ho-3d\_v3: Improving the accuracy of hand-object annotations of the ho-3d dataset. *arXiv preprint arXiv:2107.00887*, 2021. 2, 6, 7
- [14] Shreyas Hampali, Sayan Deb Sarkar, Mahdi Rad, and Vincent Lepetit. Keypoint transformer: Solving joint identification in challenging hands and object interactions for accurate 3d pose estimation. In *CVPR*, 2022. 7
- [15] 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. 7
- [16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *ECCV*, 2016. 7
- [17] Umar Iqbal, Pavlo Molchanov, Thomas Breuel Juergen Gall, and Jan Kautz. Hand pose estimation via latent 2.5 d heatmap regression. In *ECCV*, 2018. 1
- [18] Zheheng Jiang, Hossein Rahmani, Sue Black, and Bryan M Williams. A probabilistic attention model with occlusion-aware texture regression for 3d hand reconstruction from a single rgb image. In *CVPR*, 2023. 6, 7
- [19] Angjoo Kanazawa, Michael J Black, David W Jacobs, and Jitendra Malik. End-to-end recovery of human shape and pose. In *CVPR*, 2018. 6
- [20] Sameh Khamis, Jonathan Taylor, Jamie Shotton, Cem Keskin, Shahram Izadi, and Andrew Fitzgibbon. Learning an efficient model of hand shape variation from depth images. In *CVPR*, 2015. 2
- [21] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6
- [22] Nikos Kolotouros, Georgios Pavlakos, Michael J Black, and Kostas Daniilidis. Learning to reconstruct 3d human pose and shape via model-fitting in the loop. In *ICCV*, 2019. 6
- [23] Nikos Kolotouros, Georgios Pavlakos, and Kostas Daniilidis. Convolutional mesh regression for single-image human shape reconstruction. In *CVPR*, 2019. 2, 6
- [24] 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, 6, 7
- [25] Ke Li, Shijie Wang, Xiang Zhang, Yifan Xu, Weijian Xu, and Zhuowen Tu. Pose recognition with cascade transformers. In *CVPR*, 2021. 2, 5
- [26] Rui Li, Zhenyu Liu, and Jianrong Tan. A survey on 3d hand pose estimation: Cameras, methods, and datasets. *Pattern Recognition*, 2019. 2
- [27] Yanjie Li, Sen Yang, Peidong Liu, Shoukui Zhang, Yunxiao Wang, Zhicheng Wang, Wankou Yang, and Shu-Tao Xia. Simcc: A simple coordinate classification perspective for human pose estimation. In *ECCV*, 2022. 3, 4
- [28] Zhihao Li, Jianzhuang Liu, Zhensong Zhang, Songcen Xu, and Youliang Yan. Cliff: Carrying location information in full frames into human pose and shape estimation. In *ECCV*, 2022. 6, 7
- [29] Kevin Lin, Lijuan Wang, and Zicheng Liu. End-to-end human pose and mesh reconstruction with transformers. In *CVPR*, 2021. 6, 7
- [30] Kevin Lin, Lijuan Wang, and Zicheng Liu. Mesh graphormer. In *ICCV*, 2021. 1, 6, 7
- [31] Rosanne Liu, Joel Lehman, Piero Molino, Felipe Petroski Such, Eric Frank, Alex Sergeev, and Jason Yosinski. An intriguing failing of convolutional neural networks and the coordconv solution. In *NeurIPS*, 2018. 1, 4
- [32] Camillo Lugaresi, Jiuqiang Tang, Hadon Nash, Chris McClanahan, Esha Uboweja, Michael Hays, Fan Zhang, Chuo-Ling Chang, Ming Guang Yong, Juhyun Lee, et al. Mediapipe: A framework for building perception pipelines. *arXiv preprint arXiv:1906.08172*, 2019. 6
- [33] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf:

- Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 2021. 4, 5
- [34] Gyeongsik Moon and Kyoung Mu Lee. I2l-meshnet: Image-to-lixel prediction network for accurate 3d human pose and mesh estimation from a single rgb image. In *ECCV*, 2020. 2, 6, 7
- [35] Gyeongsik Moon, Hongsuk Choi, and Kyoung Mu Lee. Accurate 3d hand pose estimation for whole-body 3d human mesh estimation. In *CVPR*, 2022. 1, 6, 7
- [36] Hui En Pang, Zhongang Cai, Lei Yang, Qingyi Tao, Zhonghua Wu, Tianwei Zhang, and Ziwei Liu. Towards robust and expressive whole-body human pose and shape estimation. *NIPS*, 2024. 6, 7
- [37] Georgios Pavlakos, Dandan Shan, Ilija Radosavovic, Angjoo Kanazawa, David Fouhey, and Jitendra Malik. Reconstructing hands in 3D with transformers. In *CVPR*, 2024. 7
- [38] Tomas Pfister, James Charles, and Andrew Zisserman. Flowing convnets for human pose estimation in videos. In *ICCV*, 2015. 2
- [39] Lingteng Qiu, Xuanye Zhang, Yanran Li, Guanbin Li, Xiaojun Wu, Zixiang Xiong, Xiaoguang Han, and Shuguang Cui. Peeking into occluded joints: A novel framework for crowd pose estimation. In *ECCV*, 2020. 2
- [40] Javier Romero, Dimitrios Tzionas, and Michael J Black. Embodied hands: Modeling and capturing hands and bodies together. *arXiv preprint arXiv:2201.02610*, 2022. 2, 3, 6
- [41] Yu Rong, Takaaki Shiratori, and Hanbyul Joo. Frankmocap: A monocular 3d whole-body pose estimation system via regression and integration. In *ICCV*, 2021. 6, 7
- [42] Connor Shorten and Taghi M Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of Big Data*, 2019. 6
- [43] Jamie Shotton, Toby Sharp, Alex Kipman, Andrew Fitzgibbon, Mark Finocchio, Andrew Blake, Mat Cook, and Richard Moore. Real-time human pose recognition in parts from single depth images. *Communications of the ACM*, 2013. 3
- [44] Adrian Spurr, Umar Iqbal, Pavlo Molchanov, Otmar Hilliges, and Jan Kautz. Weakly supervised 3d hand pose estimation via biomechanical constraints. In *ECCV*, 2020. 1
- [45] Adrian Spurr, Aneesh Dahiya, Xi Wang, Xucong Zhang, and Otmar Hilliges. Self-supervised 3d hand pose estimation from monocular rgb via contrastive learning. In *ICCV*, 2021. 6
- [46] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *CVPR*, 2019. 2, 6
- [47] Xiao Tang, Tianyu Wang, and Chi-Wing Fu. Towards accurate alignment in real-time 3d hand-mesh reconstruction. In *CVPR*, 2021. 6, 7
- [48] Limao Tian, Xina Cheng, Masaaki Honda, and Takeshi Ikegami. Multi-view 3d human pose reconstruction based on spatial confidence point group for jump analysis in figure skating. *Complex & Intelligent Systems*, 2023. 3
- [49] Anastasia Tkach, Mark Pauly, and Andrea Tagliasacchi. Sphere-meshes for real-time hand modeling and tracking. *ToG*, 2016. 2
- [50] Alexander Toshev and Christian Szegedy. Deeppose: Human pose estimation via deep neural networks. In *CVPR*, 2014. 2
- [51] Bin Xiao, Haiping Wu, and Yichen Wei. Simple baselines for human pose estimation and tracking. In *ECCV*, 2018. 2
- [52] Yufei Xu, Jing Zhang, Qiming Zhang, and Dacheng Tao. Vitpose: Simple vision transformer baselines for human pose estimation. *arXiv preprint arXiv:2204.12484*, 2022. 2
- [53] Lixin Yang, Kailin Li, Xinyu Zhan, Jun Lv, Wenqiang Xu, Jiefeng Li, and Cewu Lu. Artiboost: Boosting articulated 3d hand-object pose estimation via online exploration and synthesis. In *CVPR*, 2022. 7
- [54] Yuhui Yuan, Rao Fu, Lang Huang, Weihong Lin, Chao Zhang, Xilin Chen, and Jingdong Wang. Hrformer: High-resolution vision transformer for dense predict. *NeurIPS*, 2021. 2
- [55] Hongwen Zhang, Yating Tian, Xinchu Zhou, Wanli Ouyang, Yebin Liu, Limin Wang, and Zhenan Sun. Pymaf: 3d human pose and shape regression with pyramidal mesh alignment feedback loop. In *CVPR*, 2021. 6, 7
- [56] Xiong Zhang, Hongsheng Huang, Jianchao Tan, Hongmin Xu, Cheng Yang, Guozhu Peng, Lei Wang, and Ji Liu. Hand image understanding via deep multi-task learning. In *ICCV*, 2021. 1, 6
- [57] Xiaowei Zhou, Menglong Zhu, Georgios Pavlakos, Spyridon Leonardos, Konstantinos G Derpanis, and Kostas Daniilidis. Monocap: Monocular human motion capture using a cnn coupled with a geometric prior. *TPAMI*, 2018. 6
- [58] C. Zimmermann and T. Brox. Learning to estimate 3d hand pose from single rgb images. In *ICCV*, 2017. 1
- [59] 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, 2, 6, 7
- [60] Christian Zimmermann, Max Argus, and Thomas Brox. Contrastive representation learning for hand shape estimation. In *DAGM GCPR*, 2022. 1, 6