A Dual-Branch Self-Boosting Framework for Self-Supervised 3D Hand Pose Estimation

Pengfei Ren, Haifeng Sun, Jiachang Hao, Qi Qi, *Senior Member, IEEE*, Jingyu Wang, *Senior Member, IEEE*, and Jianxin Liao

***Abstract*— Although 3D hand pose estimation has made significant progress in recent years with the development of the deep neural network, most learning-based methods require a large amount of labeled data that is time-consuming to collect. In this paper, we propose a dual-branch self-boosting framework for self-supervised 3D hand pose estimation from depth images. First, we adopt a simple yet effective image-to-image translation technology to generate realistic depth images from synthetic data for network pre-training. Second, we propose a dual-branch network to perform 3D hand model estimation and pixel-wise pose estimation in a decoupled way. Through a part-aware model-fitting loss, the network can be updated according to the fine-grained differences between the hand model and the unlabeled real image. Through an inter-branch loss, the two complementary branches can boost each other continuously during self-supervised learning. Furthermore, we adopt a refinement stage to better utilize the prior structure information in the estimated hand model for a more accurate and robust estimation. Our method outperforms previous self-supervised methods by a large margin without using paired multi-view images and achieves comparable results to strongly supervised methods. Besides, by adopting our regenerated pose annotations, the performance of the skeleton-based gesture recognition is significantly improved.**

***Index Terms*— 3D hand pose estimation, self-supervised training, 3D hand model, gesture recognition.**

I. Introduction

3D HAND pose estimation has become an active research topic in computer vision in recent years. This technology provides a natural way for users to interact with virtual environments and virtual objects, so it is a core requirement in human-computer interaction, virtual reality and augmented reality. In addition, 3D hand pose estimation plays a significant role in understanding and analyzing human behaviors and intentions, such as sign language recognition [1] and hand gesture recognition [2]–[6]. With the emergence of depth cameras and deep learning, depth-based 3D hand pose and mesh estimation has made significant progress in the past few years [7]–[24]. However, acquiring large-scale hand datasets with 3D annotations is time-consuming and labor-consuming, and the annotation quality is difficult to guarantee [25]–[28]. In addition, due to the limited camera perspectives and subjects of the existing datasets, deep neural networks trained on these datasets are not robust for the diverse camera perspective, hand shape, and scales in real scenarios.

Recently, the self-supervised hand pose estimation methods [29]–[31] get rid of dependence on labeled real data and show a strong generalization ability for real scenarios. In particular, "self-supervised" refers to deriving supervision from the data itself and the estimation of the network itself, such as maintaining the consistency between the input image and the rendered image, or maintaining the consistency of estimations from multiple views. Specifically, the self-supervised hand pose estimation methods introduce a 3D hand model into the deep neural network and adopt a set of differentiable model-fitting terms and kinetic prior terms to train the network with unlabeled real data. These methods have two necessary steps: (1) adopting synthetic data to pre-train the network; (2) adopting unlabeled real data for self-supervised fine-tuning. They achieve surprising results without any labeled real data, but they still have potential problems with the two necessary steps.

For the first step, previous methods [29]–[31] directly adopt synthetic data for pre-training, which makes the trained network difficult to generalize to the real scene because of the domain gap. Thus, the advantages of synthetic data with unlimited perspectives and shapes have not been fully exploited. The image-to-image translation technology is able to effectively convert the image of the source domain into the target domain, so as to reduce the domain gap. However, directly performing image-to-image translation on depth images may produce significant depth offset in the hand region and artifacts in the background. Therefore, we propose a depth-consistent loss to maintain the hand structure information and suppress artifacts. Meanwhile, we adopt a pose-guided information dropping mechanism to simulate the hole in real depth images and increase the diversity of synthetic data.

Manuscript received 1 August 2021; revised 12 March 2022 and 30 May 2022; accepted 7 July 2022. Date of publication 26 July 2022; date of current version 3 August 2022. This work was supported in part by the National Natural Science Foundation of China under Grant 62071067, Grant 62171057, Grant 62101064, and Grant 62001054; in part by the Ministry of Education and China Mobile Joint Fund under Grant MCM20200202; in part by the Beijing University of Posts and Telecommunications (BUPT)-China Mobile Research Institute Joint Innovation Center; and in part by the BUPT Excellent Ph.D. Students Foundation CX2020121. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Charith Abhayaratne. *(Corresponding authors: Qi Qi; Jingyu Wang.)*

The authors are with the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China, and also with EBUPT.COM, Beijing 100191, China (e-mail: rpf@bupt.edu.cn; hfsun@bupt.edu.cn; haojc@bupt.edu.cn; qiqi8266@bupt.edu.cn; wangjingyu@bupt.edu.cn; liaojx@bupt.edu.cn).

This article has supplementary downloadable material available at https://doi.org/10.1109/TIP.2022.3192708, provided by the authors.

Digital Object Identifier 10.1109/TIP.2022.3192708

For the second step, which is also the most critical step in self-supervised 3D hand pose estimation, these methods treat hand pose estimation and hand model estimation as the same task or combine them in a single branch, which sacrifices either the accuracy of pose estimation or the flexibility and robustness of hand model estimation. On the one hand, directly regressing abstract hand model parameters [29], such as joint angles and shape parameters, can enforce kinematic constraints and achieve flexible hand shape estimation. However, it is a highly non-linear process, which causes difficulties in network optimization, especially in the absence of explicit supervision, such as joint coordinates and hand model parameters. Therefore, the accuracy of this method is poor. On the other hand, some works [30], [31] propose to estimate the 3D hand model in a pixel-wise regression manner, which can maintain the spatial structure information of the feature map and achieve more accurate pose estimation. However, the pre-defined hand model in these methods lacks some prior constraints, such as joint angle or bone length. Besides, the hand shape of those models is fixed during training and testing. Therefore, the robustness and flexibility of the 3D hand model estimation are limited, which may further affect the accuracy of the estimation for the hand that has an inconsistent shape with the predefined shape.

To achieve flexible and robust hand model estimation and accurate pose estimation simultaneously, we adopt a dual-branch network to decouple the pose estimation and the hand model estimation. Specifically, a Pixel-Wise Estimation (PWE) branch performs pixel-wise pose regression to achieve accurate 3D pose estimation. A Model Parameters Estimation (MPE) branch regresses the parameters of a parametric hand model, including the shape and pose, to achieve robust and flexible 3D hand model estimation. These two branches have their own advantages and are complementary to each other. Therefore, we propose two loss terms to allow the two branches to boost each other during the self-supervised training. First, we propose an inter-branch loss that selectively uses the predicted results of these two branches as pseudo-labels for each other. The estimation results of the PWE branch have pixel-level localization accuracy and are robust to image transformation, which can provide explicit joint supervision for the MPE branch, thus reducing the training difficulty and instability of the MPE branch; In turn, the MPE branch can generate more reasonable results for the samples with self-occlusion, depth missing and image noise, which can improve the robustness of the PWE branch for these situations. Second, we propose a part-aware model-fitting loss, which performs semantic segmentation on the depth data according to the results of the PWE branch, thereby pushing each part of the hand model to the depth data region with the same semantics. On the basis of the original model-fitting loss [29], [30], the part-aware model-fitting loss can make the hand model perceive fine-grained errors from data to model.

Furthermore, the estimated hand model contains rich prior structure information, which can provide strong disambiguation clues for subsequent networks and reduce the uncertainty of subsequent estimations, such as the estimation for the region with depth missing. Therefore, we encode the estimated hand model information to the feature space by a pose re-parameterization process and adopt a refinement stage to produce a more accurate and robust 3D pose and model estimation. Our Dual-branch Self-boosting Framework (DSF) is shown in Fig. 1.

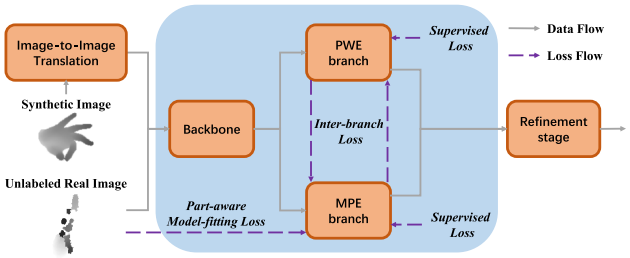

Fig. 1. Overview of our proposed dual-branch self-boosting framework. Through image-to-image translation technology, our framework can make better use of synthetic data for pre-training. The dual-branch design allows our framework to adopt a part-aware model-fitting loss for self-supervised learning on unlabeled real data while achieving high-accuracy pose estimation and flexible and robust hand model estimation simultaneously.

We conducted a quantitative and qualitative evaluation on three 3D hand pose estimation datasets (NYU [15], ICVL [7], and MSRA [32]). Experiments show that DSF outperforms previous self-supervised methods by a large margin without using any paired multi-view real data and achieves comparable results with strongly supervised methods. Then, we evaluated our method in real scenarios using the Kinect V2. Experimental results show that our method can achieve good performance in real-time and has good generalization ability. Finally, we use DSF to regenerate the hand pose data for two hand gesture recognition datasets (SHREC [33] and DHG [34]). Experiments show that the performance of the state-of-the-art (SOTA) skeleton-based gesture recognition algorithms can be significantly improved by using our generated hand pose data. The code is available at https://github.com/RenFeiTemp/DSF.

Our contributions can be summarized as follows:

- We propose a complete framework from pre-training to fine-tuning for self-supervised 3D hand pose estimation.
- We propose a simple but effective image-to-image translation technology to translate synthetic depth images into realistic depth images, which significantly reduces the domain gap between the synthetic and the real data.
- We propose a dual-branch self-boosting network to achieve accurate 3D pose estimation and flexible 3D hand model estimation simultaneously. The two branches can promote each other's performance during self-supervised learning with an inter-branch loss and a part-aware model-fitting loss.
- Our method outperforms previous self-supervised methods by a large margin without using paired multi-view images and achieves comparable results to strongly supervised methods. In addition, adopting the hand pose generated by our method significantly improves the accuracy of the skeleton-based gesture recognition algorithm.

II. Related Work

A. Depth-Based 3D Hand Pose Estimation

Depth-based 3D hand pose estimation can be categorized into three classes: model-based methods, learning-based

methods, and hybrid methods. Model-based methods use a pre-defined 3D hand model to fit the depth input by an optimization algorithm. Its effectiveness heavily relies on the construction of the hand model and the definition of the similarity function which is used to evaluate how well the 3D hand model fits the input depth data. A variety of hand models have been proposed, including sphere models [35], sphere-mesh models [36], Linear Blend Skinning [37]–[39], and Gaussian mixture models [40], [41]. Common optimization methods include Particle Swarm Optimization (PSO) [37], [42], Iterative Closest Point (ICP) [43], PSO-ICP [35] and gradient-based methods [44], [45]. This kind of method requires no labeled data, but it is sensitive to the parameters of model initialization and is easily trapped in error accumulation.

Learning-based methods use labeled data to learn a mapping between the depth image and the hand pose. Some early works [7], [46], [47] use random forests or their variants to estimate hand pose. These methods are limited by the hand-crafted features and CNN-based methods outperform it by a large margin. The deep neural network-based methods can be divided into two categories: regression-based methods and detection-based methods. Regression-based methods directly regress the parameters of the hand pose, such as 3D joint coordinates or hand model parameters. A variety of strategies are proposed to improve the performance of the regression-based method, including embedding prior knowledge [12], feedback correction [48], [49], embedding hand bone model [50], region ensemble [9], [10], multi-branch aggregation [51], stacked regression [52], adopting 3D structure network such as point cloud network [14] or 3D CNN [53], and adopting transformer [54]. Detection-based methods generate dense pixel-wise estimations such as heatmaps or offset vector fields, then obtain joint coordinates from these pixel-wise estimations. Tompson *et al.* [15] firstly apply 2D CNNs to predict heatmaps of each joint and then use model-based inverse kinematics to recover 3D hand pose. Many strategies are proposed to improve the performance of detection-based methods, such as multi-view heatmaps estimation [16], [55], dense 3D regression [18], adopting 3D structure network [17], [19], [56], [57], differentiable adaptive weighting regression [20], [21], [58], adopting graph convolution [59], etc. Although these methods have achieved excellent performance, they all rely on a large number of labeled real data.

Hybrid methods [35], [37], [60] use a learning-based method to initialize the hand model or re-initializing when tracking fails and perform temporal hand tracking using model-based methods.

B. Hand Pose Estimation With Synthetic Data

Synthetic hand images covering diverse camera perspectives, shapes, and pose variations can be generated with accurate 3D annotations by 3D computer graphics technology. For 3D hand pose estimation, some methods [22], [23], [29]–[31], [61]–[68] directly adopt synthetic data to train the network and assist the real-world hand pose estimation. However, this method can lead to severe overfitting when the labeled real data is insufficient. Therefore, most methods require 3D joint label [22], [23], [61], [62], [64], [66], 2D joint label [63], depth image information [67], or self-supervised training [29]–[31] to fine-tune on real data. Some methods learn a shared latent space for synthetic and real modalities [69]–[72]. However, these methods either need a large number of labeled real data and paired synthetic data [71] or a small amount of labeled data assisted by a domain discriminator [69], [70], [72] to align the feature distribution of different modalities; otherwise, the performance will be poor. For example, Rad *et al.* [71] map the feature of real samples to synthetic samples by minimizing feature distance between paired samples. Furthermore, Poier *et al.* [72] adopt a domain discriminator to reduce the domain gap between synthetic data and real data. Some works [73]–[76] adopt a deep neural network to translate synthetic images into target-like images, which can then be used for network training. However, these methods [74]–[76] are mostly used in RGB-based 3D hand pose estimation tasks and [73] is only evaluated in relatively simple cases where all the synthetic depth images have correspondences in real datasets. Our method can generate realism, depth-consistent depth images for synthetic images.

C. Unsupervised Domain Adaptation

Unsupervised Domain Adaptation (UDA) transfers knowledge from an existing labeled domain to a new unlabeled domain. The core of UDA is domain alignment, which can be addressed at multiple levels, including input space [77]–[81], feature space [82]–[84], and output space [85], [86]. In our work, we adopt a depth-aware image-to-image translation network to reduce the domain gap in the input space. Besides explicit domain alignment, several techniques have been proposed to improve the performance of the network on target domains, such as pseudo-labeling and self-training [87]–[90], curriculum learning [91], [92], self-supervised representation learning [93], [94], self-supervised training [30], [95]–[97], etc. Among these, our method is closely related to self-training and self-supervised training. Self-training generates pseudo-labels in the unlabeled target domain using the pre-trained model in the source domain and iteratively refines this model using the most reliable pseudo-labels. In our work, the PWE branch and the MPE branch take the estimated results of the other as pseudo-labels, respectively. In particular, our method selects reliable pseudo-labels for the PWE branch based on the fitness between the estimated hand model and the input depth data. Compared with other self-training methods, our method better exploits the 3D structure information of the input depth image. Self-supervised training derives supervision from the input data itself and the consistency of network estimations, which is widely used in human pose estimation [95], [98], face reconstruction [96], [99], [100], 6D object pose estimation [97], [101] and hand pose estimation [29]–[31], [102]. Next, we mainly focus on self-supervised training in hand pose estimation.

Dibra *et al.* [29] predict the joint angles of a hand model and train the network by minimizing the difference between a rendered image and the input image. However, directly regressing the abstract parameters is a highly non-linear process and tends to ignore some spatial details of the

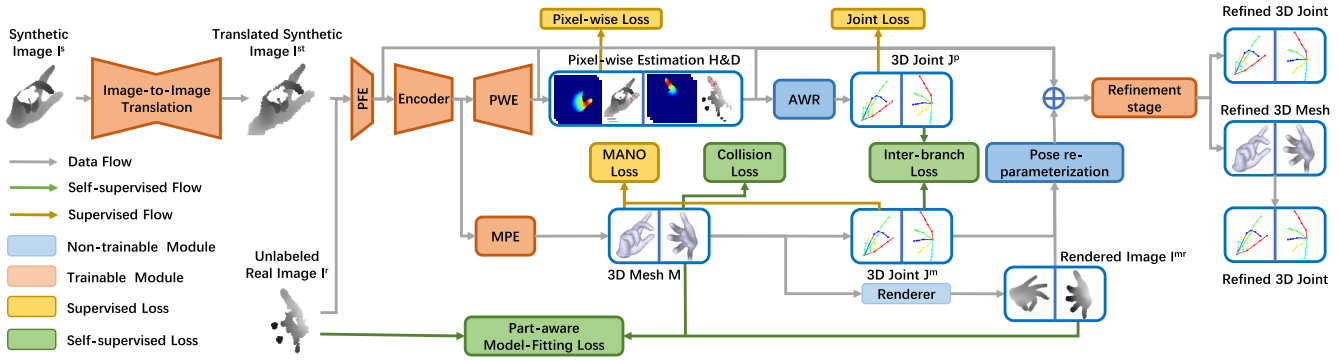

Fig. 2. Overall framework. The input of the network is the translated synthetic depth image and the unlabeled real image. A Pre-Feature Extraction (PFE) module extracts the primitive feature maps from the input depth image. An encoder generates the feature maps with high-level semantics, which are fed into the PWE branch and the MPE branch as input. The dual-branch structure performs pixel-wise pose regression and hand model estimation in parallel. Finally, a dual-branch refinement stage outputs a refined hand pose and hand model.

feature map. To solve this problem, Wan *et al.* [30] build a sphere-based model and directly predict the 2D heatmap and depth map for the center of each sphere. Adopting pixel-wise regression can better capture the spatial details of the feature map and reduce the learning difficulty of the network. Wan *et al.* [31] use a fully convolutional network to regress mesh vertices which can be seen as a 2D embedding in a 3D space. To obtain the 3D hand mesh, they adopt a differentiable re-pose mechanism to solve the similarity transform from the estimated mesh surface to a pre-defined template hand model. However, due to the fixed radius of the sphere and the fixed shape of the template mesh, the flexibility of the hand models in these methods is limited. In particular, these methods must use paired multi-view images to achieve acceptable accuracy, which greatly constrains the potential scenarios of the self-supervised 3D hand pose estimation. Chen *et al.* [102] propose to simultaneously predict 3D hand model and 2D joints, which appears to be similar to our method. However, they do not consider the different characteristics of the two branches and blindly maintain the consistency of the two branches. In order to maintain the stability of self-supervised training, they adopt an off-the-shelf 2D hand pose detector, which requires a large amount of 2D annotated data for training. In addition, two branches in their approach use different encoders. Contrary to their method, the two branches in our method share the same encoder, which facilitates the mutual promotion of the two branches in self-supervised training.

III. Methods

A. Overview

As shown in Fig. 2, we propose a Dual-branch Self-boosting Framework (DSF) to make full use of synthetic data and unlabeled real data for accurate, robust, and flexible 3D hand pose and model estimation. First, we pre-train the network with labeled synthetic data. We translate synthetic images into realistic images to reduce the domain gap between synthetic data and real data. Then, we adopt unlabeled real data to fine-tune the network in a self-supervised manner. Meanwhile, we also fed the labeled synthetic data into the network to stabilize the self-supervised training process.

B. Image Translation for Synthetic Data

Compared with capturing depth images and 3D joint coordinates simultaneously, individual 3D joint coordinates can be easily and quickly collected through a data glove [103] or some magnetic sensors [25] without considering the impact on the appearance of the depth image. Therefore, we generate synthetic depth images from individual 3D hand poses. Specifically, given a 3D hand pose, we adopt a MANO hand model [104] and use an iterative optimization method as mentioned in [105] to obtain the corresponding MANO model parameters P including hand pose $\theta \in \mathbb{R}^{45}$, hand shape $\beta \in \mathbb{R}^{10}$, global scale $S \in \mathbb{R}^{1}$, global rotation $R \in \mathbb{R}^{3}$ and global translation $T \in \mathbb{R}^{3}$. Then, we adopt a differentiable renderer [106] to render the MANO model to a depth image with accurate joint and 3D mesh coordinates labels. In particular, we randomly sample 413k hand pose data from the BigHand 2.2M hand dataset [25]. We perform an online data augmentation by sampling the hand shape with a normal distribution $N(0; 3)$, the hand scale with a uniform distribution $U(0.8, 1.2)$ and the hand global rotation with a uniform distribution $U(0, 2\pi)$.

Due to the domain gap between synthetic data and real data, the performance of the network which is trained with synthetic data will decrease significantly on real data. Therefore, we adopt an unpaired image-to-image translation technology, CycleGAN [77], to translate the synthetic depth images $\mathbf{I}^s$ into realistic target-like depth images $\mathbf{I}^{st}$. However, as shown in Fig 3, adopting the original CycleGAN may produce a significant depth offset in the hand region or artifact in the background region, which destroys the 3D geometry structural information of the depth image. Therefore, based on the original CycleGAN, we propose a depth-consistent loss to preserve the 3D structure information of the original synthetic depth image and prevent the artifact. Specifically, for each pixel i on the depth image, the depth-consistent loss is defined as:

$$\mathcal{L}_{consis} = \sum_{i \in \mathbf{I}^s} \mathbf{M}_i^{inner} \mathbf{S}_i \left| \mathbf{I}_i^{st} - \mathbf{I}_i^{s} \right| + \sum_{i \in \mathbf{I}^s} \mathbf{M}_i^{outer} \left| \mathbf{I}_i^{st} - \mathbf{I}_i^{s} \right|. \quad (1)$$

The first term in $\mathcal{L}_{consis}$ is used to maintain the depth-consistent of the hand region, and the second term is

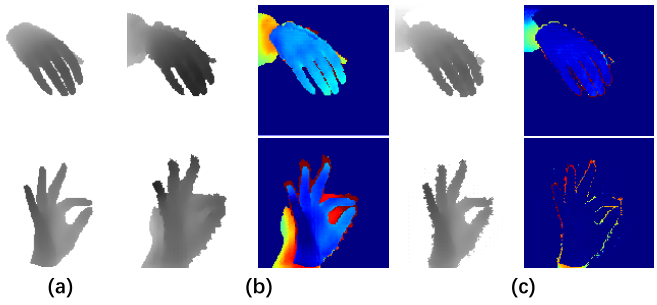

Fig. 3. (a) Synthetic images. (b) The images generated by the original CycleGAN and the difference map compared with the original synthetic images. (c) The images generated by the CycleGAN with the depth-consistent loss and the difference map compared with the original synthetic images.

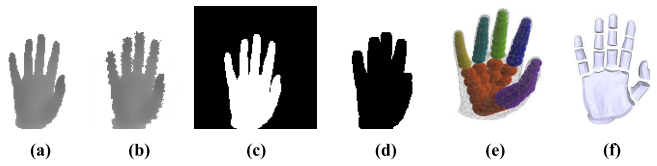

Fig. 4. (a) Original synthetic depth image $\mathbf{I}^{s}$. (b) Translated synthetic depth image $\mathbf{I}^{st}$. (c) Inner mask $\mathbf{M}^{inner}$. (d) Outer Mask $\mathbf{M}^{outer}$. (e) The spheres in 3D hand model for collision loss. (f) Different hand parts to calculate the self-intersection.

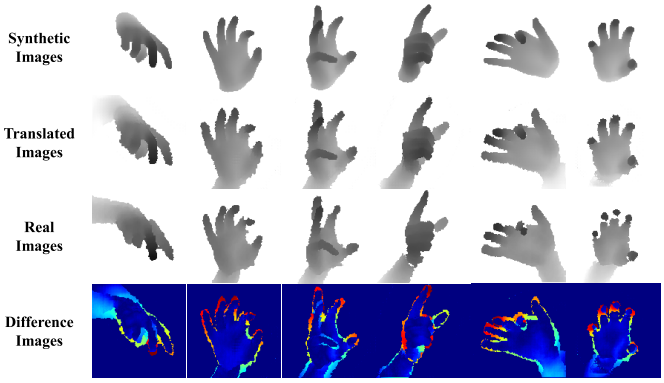

Fig. 5. Comparison between the translated image and the real image.

used to avoid artifacts in the background region. As shown in Fig. 4, the inner mask $\mathbf{M}_i^{inner}$ is set to 1 at the intersection of the two depth images' hand regions and 0 at other regions. $\mathbf{S}$ means that only a part of pixels are randomly selected, which can avoid over constraining the generation process. In practice, we randomly select 15% pixels. The outer mask $\mathbf{M}_i^{outer}$ is set to 0 in the dilated hand region of the real image and 1 at other regions. The dilated hand region is obtained by performing a 3×3 convolution on the original hand region with an all 1 value of the convolution kernel. $\mathbf{M}^{inner}$ and $\mathbf{M}^{outer}$ allows the generator to add edge noise freely without penalty for depth-consistent loss.

To further demonstrate the effectiveness of our proposed image-to-image translation technology, we compare the translated synthetic depth image with the corresponding real image. As shown in Fig. 5, our method can generate realistic noise, although it may not be completely consistent with the corresponding real depth images. Meanwhile, our method can maintain the depth consistency between the translated synthetic image and the real image.

To simulate the depth hole in the real depth image and increase the diversity of the generated synthetic image, we propose a pose-guided information dropping mechanism. We randomly select n joints and drop the depth pixels near these joints according to the 3D Euclidean distance. Specifically, for a selected joints j, we drop the depth pixels within $D \in \mathbb{R}^1$ distance from a center point $\mathbf{J}_j^{xyz} + \mathbf{o}^{xyz}$, where D is sampled with the normal distribution $N(0, 0.3)$ and $\mathbf{o}^{xyz} \in \mathbb{R}^3$ is sampled with the normal distribution $N(0, 0.15)$. This mechanism is also conducive to the pose estimation network to explore context cues and make a more accurate and robust estimation.

C. Network Structure

1) Dual-Branch Structure: As shown in Fig. 2, we adopt a dual-branch network structure, which mainly includes a Pixel-Wise Estimation (PWE) branch for pixel-by-pixel dense pose estimation, and a Model Parameter Estimation (MPE) branch for hand model parameters regression, and a backbone composed of the PFE module to extract primitive feature maps and an encoder to extract semantic feature maps.

The MPE branch adopts a fully connected layer to predict the parameters P of a MANO model, from which we can obtain a 3D hand mesh $\mathbf{M} \in \mathbb{R}^{N \times 3}$ and 3D joint coordinates $\mathbf{J}^m \in \mathbb{R}^{K \times 3}$, where N and K are the number of mesh vertices and the number of joints, respectively. Different from the previous self-supervised methods using a 3D hand model with a fixed shape, the MPE branch has stronger flexibility and adaptability for different subjects by predicting the shape β and scale S of the MANO model. At the same time, benefiting from the prior shape and pose information of the parametric hand model, the MPE branch is able to generate a more robust and reasonable hand pose, especially for the image with self-occlusion, depth holes, and image noise. However, directly regressing MANO parameters is highly nonlinear and difficult to capture fine-grained spatial information, so the accuracy of the estimated joint coordinates is unsatisfactory, especially in the absence of direct supervision information, such as joint coordinates or mesh coordinates.

The PWE branch adopts a fully convolutional structure [107] which contains three deconvolution layers to generate high-resolution semantic feature maps and two convolution layers to perform dense pose regression. Similar to previous works [18], [20], the 3D joint coordinates are re-parameterized to a set of pixel-wise estimations, i.e., 3D heat maps $\mathbf{H} \in \mathbb{R}^{K \times H \times W}$ and unit 3D directional vector fields $\mathbf{D} \in \mathbb{R}^{3K \times H \times W}$, where H and W are the height and width of the feature maps after deconvolution. This structure can better maintain the local structure information of the feature maps and make the estimate translation-invariant. In addition, we adopt the Adaptive Weighting Regression (AWR) [20] to generate 3D joint coordinates $\mathbf{J}^p \in \mathbb{R}^{K \times 3}$ from the pixel-wise estimations in a differentiable way.

2) Refinement Stage: Benefit from the prior constraint of the MANO model, the estimated hand model is robust to the image noise and depth hole in the depth image, which can provide strong disambiguation clues for subsequent estimation. Therefore, we adopt a pose re-parameterization process [18], [52]

to encode the 3D hand model information, including the 3D joint coordinates $\mathbf{J}^m$ and the rendered depth image $\mathbf{I}^{mr}$, into pixel-level features, i.e., 3D heat maps $\widetilde{\mathbf{H}} \in \mathbb{R}^{K\times H\times W}$ and unit 3D directional vector fields $\widetilde{\mathbf{D}} \in \mathbb{R}^{3K\times H\times W}$. Then, we adopt a refinement stage to generate refined results. The refinement stage has the same dual-branch structure and supervision as the previous stage. The only difference is that in addition to the primitive features from the PFE module, its input also includes the semantic features from the previous PWE branch, the pixel-wise estimation from the previous PWE branch and the encoded hand model features from the previous MPE branch.

D. Supervision of Network

1) Supervision of the Labeled Synthetic Data: For synthetic data, since the depth image is obtained according to the 3D hand model through the online rendering, we have accurate joint coordinate annotations $\hat{\mathbf{J}} \in \mathbb{R}^{K\times 3}$ and mesh coordinate annotations $\hat{\mathbf{V}} \in \mathbb{R}^{N\times 3}$. At the same time, we can obtain the ground-truth value of the 3D heat maps $\hat{H}$ and unit 3D directional vector fields $\hat{D}$.

For the PWE branch, we adopt a joint loss $\mathcal{L}_{joint}$ and a pixel-wise loss $\mathcal{L}_{pixel}$ as follows:

$$\mathcal{L}_{joint} = \sum_{j=1}^{K} Smooth_{L1}\left(\mathbf{J}_j^p, \hat{\mathbf{J}}_j\right), \tag{2}$$

$$\mathcal{L}_{pixel} = \sum_{j=1}^{K} Smooth_{L1}\left(\mathbf{H}_j, \hat{\mathbf{H}}_j\right) + \sum_{j=1}^{K} Smooth_{L1}\left(\mathbf{D}_j, \hat{\mathbf{D}}_j\right). \tag{3}$$

For the MPE branch, we adopt a MANO loss $\mathcal{L}_{mano}$ as follow:

$$\mathcal{L}_{mano} = \sum_{v=1}^{N} Smooth_{L1}\left(\mathbf{V}_v, \hat{\mathbf{V}}_v\right) + \sum_{j=1}^{K} Smooth_{L1}\left(\mathbf{J}_j^m, \hat{\mathbf{J}}_j\right). \tag{4}$$

Here, we adopt the $Smooth_{L1}$ loss [9], [10], [52] to make the loss terms robust to the outliers. In addition, in order to avoid extreme mesh deformations, we adopt the shape loss $\mathcal{L}_{shape}$ to constrain the predicted 3D mesh shape β as close as possible to the average shape. The shape loss term is defined as:

$$\mathcal{L}_{shape} = \|\beta\|_2^2. \tag{5}$$

In particular, this loss is only used during the pre-training.

2) Supervision of the Unlabeled Real Data: As shown in Fig. 2, we propose three carefully designed self-supervised losses for the dual-branch architecture, including a part-aware model-fitting loss $\mathcal{L}_{fitting}$, an inter-branch loss $\mathcal{L}_{inter}$, and an adaptive collision loss $\mathcal{L}_{coll}$.

$$\mathcal{L}_{MPE} = w_{fitting}\mathcal{L}_{fitting} + w_{inter}\mathcal{L}_{inter} + w_{coll}\mathcal{L}_{coll}. \tag{6}$$

where $w_{fitting}$, w_{inter}, w_{coll} are the loss weights, set to 1, 1, 1, respectively. We will first introduce the model-fitting loss used in previous work, and then we will introduce the new loss proposed in our work.

a) Model-fitting loss: For unlabeled real data $\mathbf{I}^r$, we update the model by penalizing the distance between the estimated hand surface to 3D points from the input depth image. Similar to the previous method [29]–[31], we adopt a model-to-data term $\mathcal{L}_{m2d}$ and a data-to-model term $\mathcal{L}_{d2m}$ to measure the global fitness between the model and the depth data. $\mathcal{L}_{m2d}$ penalizes the model from lying outside the hand region and aligns the hand model as close as possible to the hand depth image. We adopt a differentiable renderer [106] to render the estimated hand mesh to a depth image $\mathbf{I}^{mr}$.

$$\mathcal{L}_{m2d} = \sum_{i\in\mathbf{I}^{mr}} \left|\mathbf{I}_i^{mr} - \mathbf{I}_i^r\right|. \tag{7}$$

$\mathcal{L}_{d2m}$ tries to find the best correspondence on the whole hand mesh surface for the 3D point corresponding to pixel i. Specifically, we approximate the point-to-hand model surface distance by finding the nearest vertex from the estimated 3D hand mesh $\mathbf{V}$. $\mathcal{L}_{d2m}$ is defined as follows:

$$\mathcal{L}_{d2m} = \sum_{i\in\mathbf{I}^{xyz}} \min_{v\in\mathbf{V}} \left(Smooth_{L1}\left(\mathbf{I}_i^{xyz}, \mathbf{V}_v\right)\right), \tag{8}$$

where $\mathbf{I}_i^{xyz} \in \mathbb{R}^3$ are the corresponding 3D point of the pixel i, which can be obtained by a projection from the image plane coordinates of the pixel i.

b) Part-aware model-fitting loss: Part-aware model-fitting loss improves the data-to-model terms in the previous model-fitting loss. Specifically, a 3D point and its nearest vertex on hand mesh are not guaranteed to have the same semantics. Therefore, blindly bringing them closer may lead to incorrect fitting results. In order to alleviate this problem, we propose a part-aware data-to-model term $\mathcal{L}_{pd2m}$, which only considers minimizing the distance between the 3D point and mesh vertex with same semantics. $\mathcal{L}_{pd2m}$ is defined as follows:

$$\mathcal{L}_{pd2m} = \sum_{j=1}^{K} \sum_{i\in\mathbf{I}_j^{xyz}} \min_{v\in\mathbf{V}_j} \left(Smooth_{L1}\left(\mathbf{I}_{j,i}^{xyz}, \mathbf{V}_{j,v}\right)\right), \tag{9}$$

where $\mathbf{I}_j$ and $\mathbf{V}_j$ represent the pixels and vertices belonging to joint j in depth image and mesh respectively. We determine the semantics of each mesh vertex according to the skin weight between the mesh vertex and the joint. Specifically, when the corresponding skin weight is greater than 0.1, we determine that the mesh vertex belongs to the joint. We use the estimation results of the PWE branch to generate joint-wise semantic segmentation for the depth image. Specifically, given the estimated hand pose $\mathbf{J}^p$, the segmentation result for pixel i can be obtained as:

$$c_i = \underset{j\in[1,K]}{argmin} \left|\left\|\mathbf{J}_j^p - \mathbf{I}_i^{xyz}\right\|_2 - r_j\right|, \tag{10}$$

where r_j represents the radius of the joint j. Compared with $\mathcal{L}_{d2m}$, $\mathcal{L}_{pd2m}$ can better reflect the joint-level offset and not easily influenced by unrelated joints. As shown in Fig. 6, when the index finger moves laterally or even completely occluded by the palm of the hand, the change of $\mathcal{L}_{d2m}$ from Fig. 6(c) to Fig. 6(d),(e) is relatively small, while $\mathcal{L}_{pd2m}$ and $\mathcal{L}_{pd2m,index}$ increase significantly from Fig. 6(c) to Fig. 6 (d),(e). Therefore, adopting $\mathcal{L}_{pd2m}$ on the basis of $\mathcal{L}_{d2m}$ can make the network better perceive the fine-grained difference between the hand model and the depth data.

c) Inter-branch loss: Due to different network structures and regression tasks, the two branches have their unique advantages. On the one hand, the PWE branch has pixel-level localization accuracy and is more robust to image translation. On the other hand, the MPE branch benefits from the prior structure information of the parametric hand model, which can produce more reasonable estimation results for images with self-occlusion, depth holes, or noise. Therefore, we propose an inter-branch loss to enable the two branches to promote each other during the training process. Specifically, the inter-branch loss contains two items, a supervision item $\mathcal{L}_{P2M}$ from the PWE branch to the MPE branch, and a supervision item $\mathcal{L}_{M2P}$ from the MPE branch to the PWE branch.

For $\mathcal{L}_{P2M}$, we directly take the estimated joints $\mathbf{J}^p$ from the PWE branch as pseudo-labels of the MPE branch. Given the estimated joints $\mathbf{J}^p$, $\mathcal{L}_{P2M}$ is defined as:

$$\mathcal{L}_{P2M} = \sum_{j=1}^{K} Smooth_{L1}\left(\mathbf{J}_j^m, \mathbf{J}_j^p\right). \tag{11}$$

$\mathcal{L}_{P2M}$ makes the MPE branch perceive pixel-level estimation errors, so as to improve the estimation accuracy. More importantly, by adopting the explicit supervision of joints, we can adopt the model-fitting loss to optimize the network without fixing the model's shape, which significantly improves the flexibility of hand model estimation.

Because the PWE branch lacks additional constraint information, i.e., the model-fitting loss, directly using the estimated results of the MPE branch as the pseudo label can easily cause the PWE branch to overfit the inaccurate pseudo label, which affects the MPE branch through the $\mathcal{L}_{P2M}$ in turn, and finally leads to the collapse of the whole network. Therefore, for $\mathcal{L}_{M2P}$, we propose a filtering mechanism that only passes highly reliable joints to the PWE branch as pseudo-labels. Given the estimated joints $\mathbf{J}^m$ from the MPE branch, $\mathcal{L}_{M2P}$ is defined as:

$$\mathcal{L}_{M2P} = \sum_{j=1}^{K} \mathbf{M}_j^{filter} Smooth_{L1}\left(\mathbf{J}_j^p, \mathbf{J}_j^m\right). \tag{12}$$

We determine the value of $\mathbf{M}_j^{filter}$ according to the global fitness and the part-wise fitness between the estimated hand model and the input data. $\mathbf{M}_j^{filter}$ is defined as follows:

$$\mathbf{M}_j^{filter} = \begin{cases} 1 & \mathcal{L}_{m2d} < 0.04, \mathcal{L}_{d2m} < 0.001, \\ & \mathcal{L}_{pd2m,j} < 0.001, \\ 0 & \text{otherwise.} \end{cases} \tag{13}$$

d) Adaptive collision loss: In order to make the 3D joint and 3D hand mesh predicted by the MPE branch more plausible, we introduce an adaptive collision loss $\mathcal{L}_{coll}$ to avoid unreasonable collisions. Specifically, we place multiple spheres in the hand mesh and then penalize overlaps between these spheres, as shown in Fig. 4. Different from the previous methods [29], [30] that adopt a set of colliders with a fixed radius, the radius of these spheres can be automatically adjusted according to the distance between the corresponding joints and its related mesh vertices, so it has a strong adaptive ability. $\mathcal{L}_{coll}$ penalizes collisions between the m-th and n-th sphere as follows:

$$\mathcal{L}_{coll} = \sum_{m,n} \mathbf{M}_{m,n}^{sphere} max(r_m + r_n - \|c_m - c_n\|_2, 0), \tag{14}$$

where r and c represents the radius and the 3D coordinates of the sphere, respectively; We adopt $\mathbf{M}^{sphere}$ to discard some collisions that do not need to be considered, such as the collisions between the palm spheres and the collisions between the spheres in the same bone.

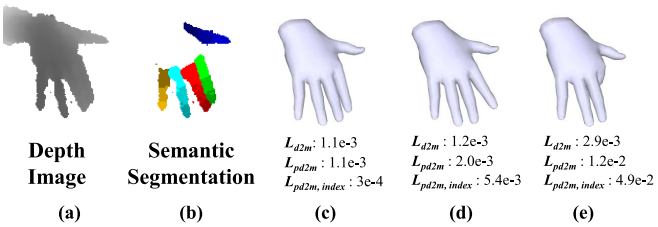

Fig. 6. Comparison of different data-to-model terms. The three columns on the right show three hand models with different pose, each of which contains three loss values between the model and the depth data. $\mathcal{L}_{pd2m,index}$ represents the average part-aware data-to-model loss of the four joints of the index finger.

IV. Experiment

A. Dataset and Evaluation Metrics

1) 3D Hand Pose Dataset: **NYU dataset** [15] has three paired camera perspectives, including a frontal perspective (View 1) and two side perspectives (View 2 and View 3), captured by the PrimeSense 3D camera. For each perspective, it contains 72K training images collected from a single subject and 8.2K testing images collected from two subjects, one of which is the same as the subject in the training set. Following previous methods, we only use images of View 1 for training, but we tested our method in all three perspectives. **ICVL dataset** [7] consists of 22K training and 1.6K testing depth images captured by the Intel RealSense SR300. The training images are collected from 10 subjects and the testing images are collected from 2 subjects. The annotation of the hand pose contains 16 joints. **MSRA dataset** [32] contains 76.5K images captured by the Intel RealSense SR300 from 9 subjects. Each subject contains 17 hand gestures with 21 annotated joints. As mentioned in [27], [28], ICVL and MSRA datasets have a lot of annotation errors. Therefore, similar to [28], we only perform qualitative analysis on ICVL and MSRA.

We evaluate our approach using two widely used metrics: the Average Joint Error (AJE) and the Success Rate (SR). The AJE is the average Euclidean distance between the predicted coordinates and the ground-truth coordinates for each joint over the whole test set. The SR is defined as the proportion of good frames in all testing frames. If the maximum value of the joint error in a frame is less than a certain threshold, it will be judged as a good frame. In addition, we adopt a Cross-view Average Joint Error (CAJE), which represents the average joint error of all frames in all three camera perspectives, which is able to measure the performance of our method in multiple camera perspectives. In order to more accurately and reliably evaluate our method, we run all experiments five times and

TABLE I

COMPARISON OF DIFFERENT PRE-TRAINING STRATEGIES. THE CAJE (mm) OF THE PWE BRANCH AND THE MPE BRANCH ON NYU DATASET ARE SHOWN IN THIS TABLE

Name	Multi-view	CycleGAN	Consistent	Dropping	PWE	MPE
Synth					23.85 ± 0.88	32.65 ± 1.12
Synth-MultiView	√				21.78 ± 0.83	28.91 ± 0.97
Synth-Cycle	√	√			17.31 ± 0.43	28.44 ± 0.71
Synth-Consis	√	√	√		16.92 ± 0.45	22.06 ± 0.51
Synth-Trans	√	√	√	√	15.50 ± 0.51	21.93 ± 0.55

TABLE II

COMPARISON OF DIFFERENT SELF-SUPERVISION LOSS ITEMS. THE CAJE (mm) OF THE PWE BRANCH AND THE MPE BRANCH ON NYU DATASET ARE LISTED IN THIS TABLE

Name	$\mathcal{L}_{P2M}$	$\mathcal{L}_{m2d}$	$\mathcal{L}_{d2m}$	$\mathcal{L}_{pd2m}$	$\mathcal{L}_{M2P}$	$\mathcal{L}_{coll}$	PWE	MPE
Synth-Trans							15.50 ± 0.51	21.93 ± 0.55
SelfSup-P2M	√						13.98 ± 0.17	17.23 ± 0.24
SelfSup-Model	√	√	√				13.58 ± 0.16	16.86 ± 0.21
SelfSup-Part	√	√	√	√			13.56 ± 0.16	16.65 ± 0.23
SelfSup-M2P	√	√	√	√	√		12.80 ± 0.07	16.10 ± 0.13
SelfSup	√	√	√	√	√	√	12.79 ± 0.13	16.03 ± 0.17

take the last epoch for evaluation in each experiment. For the self-supervised training, we choose the model that is closest to the average performance as the pre-trained model.

2) Gesture Recognition Dataset: **DHG dataset** [34] includes 14 gestures with 2800 sequences provided by 20 subjects. They are performed in two ways: using one finger and the whole hand. So it has two benchmarks: 14-gestures for coarse classification and 28-gestures for fine-grained classification. The 3D coordinates of 22 hand joints in real-world space is captured by the Intel RealSense F200. We use the leave-one-subject-out experimental protocol for training and testing. **SHREC dataset** [33] contains 2800 gesture sequences performed 1 and 10 times by 28 subjects in two ways like the DHG dataset. It contains 2800 sequences, which are divided into 1960 sequences for training and 840 sequences for testing.

B. Implementation Details

We train and evaluate our method on a computer with an AMD Ryzen 9 3900X CPU and an NVIDIA RTX 3090 GPU having 24GB of GPU memory. For the whole system, first, we train the image-to-image translation network separately and freeze it in subsequent training. Then, we pre-train all parts of the pose estimation network end-to-end, including the backbone, PWE branch, and MPE branch, with labeled synthetic images. Finally, we fine-tune all parts of the pose estimation network end-to-end on unlabeled real images. In particular, during fine-tuning, we also adopt labeled synthetic data to stabilize the training. The image-to-image translation network is trained using Adam [108] with an initial learning rate of 0.0002. From the 20th epoch to the 40th epoch, the learning rate gradually and linearly decreases to 0. The dual-branch network is trained using AdamW [109] with an initial learning rate of 0.001. The learning rate is divided by 10 at 10 epochs and the training stops at 15 epochs for both pre-training and fine-tuning. To crop the hand image from the original depth image, we use the hand center provided by [17]. The cropped image is resized to 128×128 and the depth value is normalized to $[-1, 1]$. For real data, we perform data augmentation including random rotation ($[-180, 180]$), random scaling ($[0.8, 1.2]$) and random translation ($[-10, 10]$).

C. Ablation Study

Ablation experiments are conducted on the NYU dataset [15] since it has a larger variance in pose and accurate annotations in different camera perspectives. In order to ensure the simplicity of the experiments, we use a dual-branch network without refinement stage by default.

1) Impact of Image-to-Image Translation: We investigate the impact of different pre-training strategies on the network in this section. As shown in Table I and Fig. 7, when we use synthetic data with a random sampling of shape and scale to pre-train the network, the performance of the network on the real data is unsatisfactory. Then, we add a random sampling of hand rotation for synthetic data ("Synth-MultiView", which consistently improves the performance of the two branches in all three views. In order to reduce the domain gap, we adopt the CycleGAN to convert the synthetic images to realistic target-like images ("Synth-Cycle"). As shown in Table I, adopt the CycleGAN improves the performance of two branches, but the accuracy of the network, especially the MPE branch, is still poor. Compared with the original CycleGAN, adopting the depth-consistent loss decreases the CAJE of the PWE branch and the MPE branch by 2.3% (from 17.31 mm to 16.92 mm) and 22.4% (from 28.44 mm to 22.06 mm), respectively. Meanwhile, the depth-consistent loss increases SR under all thresholds from all three views. This shows that the depth offset and artifacts significantly impair the effect of the image-to-image translation. By adding the pose-guided information dropping mechanism ("Synth-Trans"), the performance of the MPE branch is improved on all metrics. In later experiments, we use the Synth-Trans as default.

2) Impact of Self-Supervision: In this section, We investigate the effects of different self-supervision losses. As shown in Table II, adopting $\mathcal{L}_{P2M}$ alone ("SelfSup-P2M") can significantly improve the accuracy of the MPE branch. Meanwhile, since the network learns better representations for the real data domain, the accuracy of the PWE branch is also improved. Then, we add the model-fitting loss, $\mathcal{L}_{m2d}$ and $\mathcal{L}_{d2m}$, which are widely used in previous self-supervised works. The model-fitting loss ("SelfSup-Model") improves the performance of the two branches on the basis of style transfer and $\mathcal{L}_{P2M}$. By adopting the part-aware data-to-model term $\mathcal{L}_{pd2m}$ ("SelfSup-Part"), the MPE branches obtain a further improvement. After that, we add the $\mathcal{L}_{M2P}$ ("SelfSup-M2P") to the network, which selects the most reliable part of the estimations from the MPE branch as pseudo-labels to supervise the PWE branch. $\mathcal{L}_{M2P}$ significantly improves the performance of the PWE branch the MPE branch. The above experiment shows the self-boosting characteristics of the dual-branch structure. The loss item specially designed for one branch will improve another branch in turn. Finally, we add the collision loss $\mathcal{L}_{coll}$ ("SelfSup"). Although this loss has little effect on the improvement of network performance, it effectively reduces the occurrence of unreasonable collisions, especially the self-intersection between fingers (We show some examples in the supplementary material). In particular, adopting the model-fitting loss alone leads to instabilities of training. Adopting $\mathcal{L}_{P2M}$ and $\mathcal{L}_{pd2m}$ can prevent the collapse

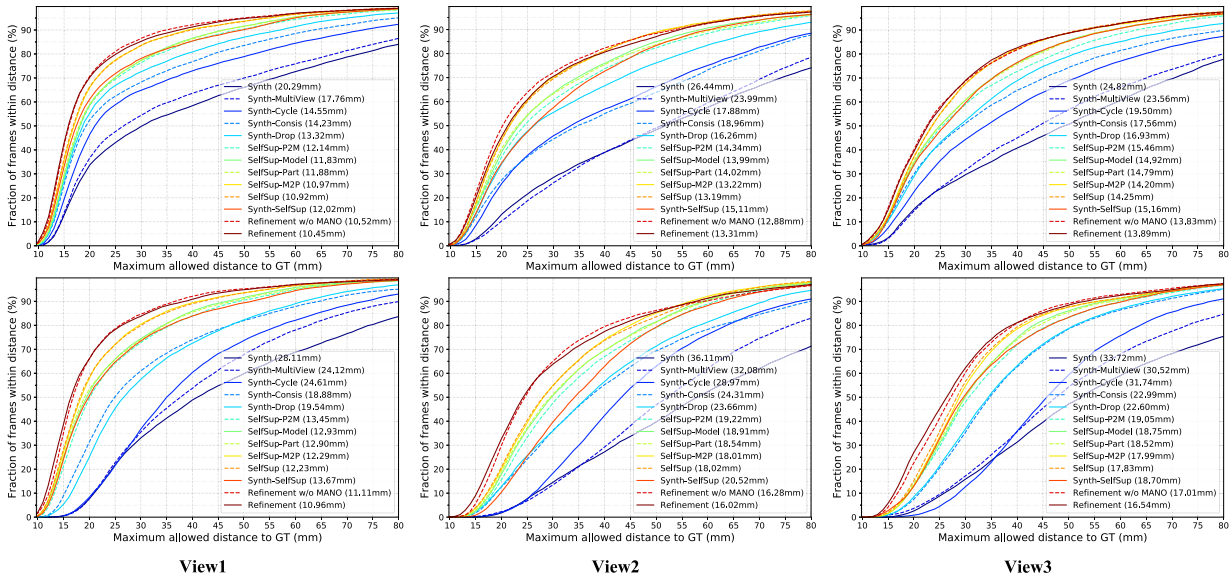

Fig. 7. Ablation experiments. The proportions of good frames over different thresholds on the NYU dataset for the three views. Top: the results of the PWE branch. Bottom: the results of the MPE branch. The AJE (mm) are presented in parentheses.

of training. Furthermore, as shown in Table II, adopting $\mathcal{L}_{M2P}$ significantly improves the stability of self-supervised training.

3) Overall Analysis: In this section, we conduct an overall analysis of the impact of the image-to-image translation, the self-supervised training, and the refinement stage. As shown in Fig. 7 and Table III, compared with the baseline model ("Synth-MultiView"), both the image-to-image translation ("Synth-Trans") and the self-supervised training ("Synth-SelfSup") alone can drastically improve the performance of the network. Specifically, Synth-Trans reduces the CAJE of the PWE branch and the MPE branch by 28.8% (from 21.78 mm to 15.50 mm) and 24.1% (from 28.91 mm to 21.93 mm); Synth-SelfSup reduces the CAJE of the PWE branch and the MPE branch by 35.2% (from 21.78 mm to 14.10 mm) and 39% (from 28.91 mm to 17.63 mm). This shows that the image-to-image translation and the self-supervised training can work well independently. Overall, the improvement brought by the self-supervised training is greater than the improvement brought by the image-to-image translation. When we adopt the image-to-image translation and the self-supervised training at the same time ("SelfSup"), the performance of the network will be further improved. As shown in Fig. 7 and Table III, the refinement stage consistently improves the performance of the two branches on almost all metrics. In particular, when we remove the re-parameterized features of the hand model ("Refinement w/o MANO"), the performance of the MPE branch decreases, especially for the camera perspectives 2 and 3. This shows that the re-parameterization of the hand model provides strong disambiguation clues for subsequent network, which helps to eliminate the ambiguity and improve the accuracy of estimation.

In addition, we analyze the impact of collision loss for the self-intersection of the estimated hand model. Specifically, we divided the hand model into 15 parts (Fig. 4 (f)) according to semantics and calculate the intersection volume between these parts. We voxelize each part of the hand mesh with voxel size 0.1 cm, and calculate the sum of the voxel volume

TABLE III

OVERALL COMPARISON. THE CAJE (mm) OF THE PWE BRANCH AND THE MPE BRANCH ON NYU DATASET ARE LISTED IN THIS TABLE

Name	Translation	Self-Sup	Refinement	PWE	MPE
Synth-MultiView				21.78 ± 0.83	28.91 ± 0.97
Synth-Trans	√			15.50 ± 0.51	21.93 ± 0.55
Synth-SelfSup		√		14.10 ± 0.33	17.63 ± 0.64
SelfSup	√	√		12.79 ± 0.13	16.03 ± 0.17
Refinement w/o MANO	√	√	√	12.41 ± 0.12	14.80 ± 0.18
Refinement	√	√	√	12.55 ± 0.13	14.51 ± 0.19

TABLE IV

THE IMPACT OF COLLISION LOSS FOR THE SV (cm^3) OF THE ESTIMATED HAND MODEL

	SelfSup w/o $\mathcal{L}_{coll}$	SelfSup	Refinement w/o $\mathcal{L}_{coll}$	Refinement
SV	16.65 ± 0.64	14.86 ± 0.32	16.3 ± 0.67	11.37 ± 0.28

shared by any two non-adjacent 3D voxels. As shown in Table IV, adopting the collision term can significantly reduce self-intersection volume (SV) of the estimated hand model. Specifically, if the collision loss is not used, the SV of "SelfSup" and "Refinement" will increase by 1.79 cm^3 and 4.93 cm^3 respectively.

4) Impact of the Training Data: Inspired by previous works [30], [31], we investigate how different training data influences the resulting network. First, we train only with the testing samples to check how well the self-supervision might "overfit" the network to training data. For the simplicity of the experiment, we train on the 8252 testing samples of view 1 and report the test results on view 1. Then, we train with a combination of both the testing and training samples of view 1. As shown in Table V, contrary to the previous works [30], [31], the performance of training directly on the testing samples alone ("SelfSup-Test") is comparable with the performance of training only on training samples ("SelfSup"). In particular, the result of the MPE branch is improved by 9.4% (11.09 mm vs 12.24 mm). We also observe that through training on the testing samples, the performance gap between the PWE branch and the MPE branch becomes very small.

TABLE V

COMPARISON OF USING DIFFERENT TRAINING DATA. THE AJE (mm) OF THE PWE BRANCH AND THE MPE BRANCH ON NYU DATASET FOR VIEW 1 ARE LISTED IN THIS TABLE

Name	Train Data	Test Data	Refinement	PWE	MPE
SelfSup	√			10.92 ± 0.14	12.24 ± 0.12
SelfSup-Test		√		11.07 ± 0.13	11.09 ± 0.13
SelfSup-Train-Test	√	√		10.88 ± 0.06	10.92 ± 0.10
Refinement	√		√	10.45 ± 0.09	10.96 ± 0.11
Refinement-Train-Test	√	√	√	10.43 ± 0.07	10.66 ± 0.10

TABLE VI

THE COMPARISON OF THE CAJE (mm) WITH UDA METHODS ON NYU DATASET

	CycleGAN	CyCADA	TASK-CYCLE	Synth-Trans	CCSSL	SelfSup
PWE	17.31 ± 0.43	18.37 ± 0.57	16.37 ± 0.18	15.50 ± 0.51	13.86 ± 0.24	12.79 ± 0.13
MPE	28.44 ± 0.71	27.30 ± 0.41	23.38 ± 0.40	21.93 ± 0.55	17.59 ± 0.32	16.03 ± 0.17

These two experiment results show that our self-supervised learning can fit the training data well, even when the training data is limited. Undoubtedly, when training with the combined training and test set, the performance of the network, including single-stage and two-stage, is further improved.

D. Comparisons With UDA Methods

We compare our method with some related UDA methods, including CyCADA [79], TASK-CYCLE [80], and CCSSL [89]. CyCADA maintains the semantic consistency of images before and after translation through a task loss and performs feature-level domain adaptation through an adversarial loss. Inspired by CyCADA, TASK-CYCLE also adds a task-related constraint during training the image-to-image translation network. In addition, they iteratively train the task network and the image-to-image translation network to gradually improve the performance of the two networks. In practice, we take hand pose estimation and hand model estimation as target tasks and perform three iterations. CCSSL is a UDA method based on pseudo-labeling and self-training. CCSSL generates reliable pseudo-labels by consistency check, including invariance consistency, equivariance consistency, and temporal consistency. In addition, CCSSL introduces the idea of curriculum learning to gradually increase the proportion of pseudo-labels in self-training. For a fair comparison, we do not use temporal information in these methods. In addition, we use "Synth-Trans" as the initialization model in CCSSL, which significantly improves the performance of CCSSL. As shown in Table VI, TASK-CYCLE outperforms CycleGAN and CyCADA but is still worse than "Synth-Trans". This shows the effectiveness of the depth-consistent loss and the pose-guided information dropping mechanism. Compared with "Synth-Trans", we find that the performance of TASK-CYCLE is more stable, which may be due to the iterative training mechanism. Since CCSSL does not exploit the 3D structure information of the depth data during self-training, although it adopts a well-designed consistency check, its performance is still far inferior to our method.

E. Comparisons With State-of-the-Arts

On NYU dataset, we compare our approach with most of the strongly supervised state-of-the-art methods: the regression-based method with a pose prior (DeepPrior) [12], the regression-based method using a feedback loop (Feedback) [48], the Lie-X method [8] applying the Lie group theory, the hand model parameter regression method (DeepModel) [50], the regression-based 3D convolutional network (3DCNN) [13], the region ensemble network (REN-9 × 6×6) [9], the pose guided hierarchical REN (Pose-REN) [10], the point-based regression method (HandPointNet) [14], the pixel-wise regression method (DenseReg) [18], the 3D voxel-to-voxel estimation network (V2V) [17], the feature mapping from synthetic images (FeatureMapping) [71], the point-wise regression network (P2P) [19], the improved Feedback (Generalized-Feedback) [49], the regression-based methods using paired multi-view image (MURAUER) [72], the multitask information sharing network (CrossInfo-Net) [110], the point-based method using a self-organizing network (SO-HandNet) [111], the anchor-to-joint regression network (A2J) [21], the adaptive weighting regression network (AWR) [20] and the pixel-wise regression method using the graph convolution (JGR-P2O) [59]. We also compare our method with, to the best of our knowledge, the only three self-supervised methods, including the 3D model-fitting method (3D refine) [29], the sphere-model based method with single-view data (SM-SV) and paired multi-view data (SM-MV) [30], and the mesh-model based method with single-view data (MM-SV) and paired multi-view data (MM-MV) [31]. Considering the different joint configurations between the MANO model and the NYU dataset, we deleted two wrist joints and one palm joint from 14 joints during the comparison. For the self-supervised methods, since they do not provide the predicted result files or the source code, we cannot recalculate their result. Therefore, we report the results in their original paper.

As shown in Fig. 8, our method outperforms previous self-supervised methods (SM-SV, MM-SV) by large margins under all thresholds. As shown in Table VII, our method reduces the AJE by 38.4% compared to the SOTA self-supervised method (10.45 mm vs 16.96 mm). Compared with the self-supervised methods adopting paired multi-view data (SM-MV and MM-MV), our method still achieves the lowest AJE (10.45 mm vs 12.26 mm). As shown in Figure 8, our method outperforms these three self-supervised methods when the error threshold is less than 30 mm, even if SM-MV and MM-MV additionally use paired multi-view data. In five experiments, the maximum AJE and minimum AJE of the PWE branch are 10.55 mm and 10.33 mm, and the maximum AJE and the minimum AJE of the MPE branch are 11.12 mm and 10.80 mm. Due to the intrinsic joint position bias between the MANO model and the NYU dataset, our method is worse than some supervised methods on View 1. However, for the unseen views 2 and 3, even in the case of joint bias, DSF reduces the AJE of the previous supervised method by 14.2% (from 15.52 mm to 13.31 mm) and 19.2 % (from 17.18 mm to 13.89 mm). Meanwhile, when the error threshold is greater than 20 mm, the SR of DSF is significantly higher than that of other supervised methods. This proves that our method has good generalization ability for the unseen camera perspectives.

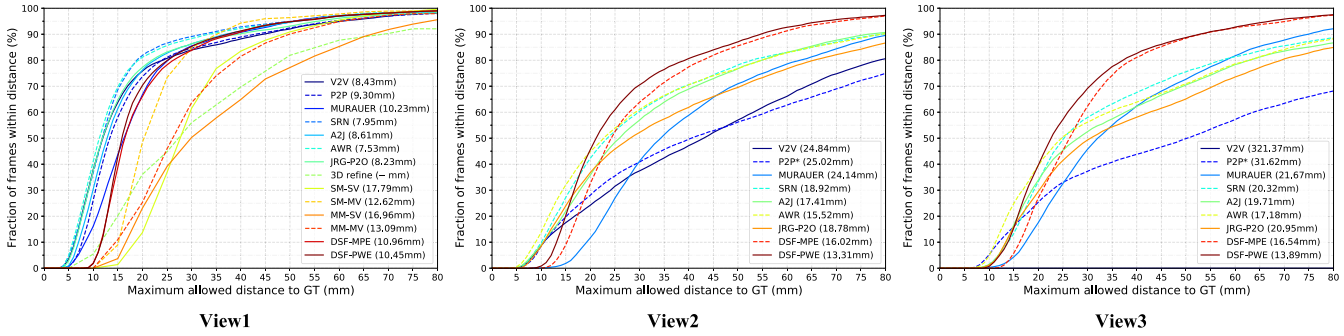

Fig. 8. Comparison of DSF with state-of-the-art methods. The proportions of good frames over different thresholds on the NYU dataset for the three views. $*$ means this method is implemented by ourselves. The AJE are presented in parentheses.

TABLE VII
THE COMPARISON OF THE AJE (mm) WITH STATE-OF-THE-ART METHODS ON NYU DATASET

Method	Publication	AJE
Supervised Method		
DeepPrior [12]	ICCVW'15	21.67
Feedback [48]	ICCV'16	17.94
Lie-X [8]	IJCV'17	19.21
DeepModel [50]	IJCAI'16	16.39
3DCNN [53]	CVPR'17	15.10
REN-9×6×6 [9]	ICIP'17	13.13
Pose-REN [10]	Neurocomputing'18	12.05
HandPointNet [14]	CVPR'18	11.02
DenseReg [18]	CVPR'18	9.60
V2V [17]	CVPR'18	8.43
FeatureMapping [71]	CVPR'18	7.44
P2P [19]	ECCV'18	9.30
Generalized-Feedback [49]	TPAMI'19	12.10
MURAUER [72]	WACV'19	10.23
CrossInfoNet [110]	CVPR'19	10.43
SRN [52]	BMVC'19	7.95
A2J [21]	ICCV'19	8.61
SO-HandNet [111]	ICCV'19	11.20
AWR [20]	AAAI'20	7.53
JGR-P2O [59]	ECCV'20	8.23
Self-supervised Method		
SM-SV [30]	CVPR'19	17.79
SM-MV [30]	CVPR'19	12.26
MM-SV [31]	ECCV'20	16.96
MM-MV [31]	ECCV'20	13.09
DSF-MPE		10.45 ± 0.09
DSF-PWE		10.96 ± 0.11

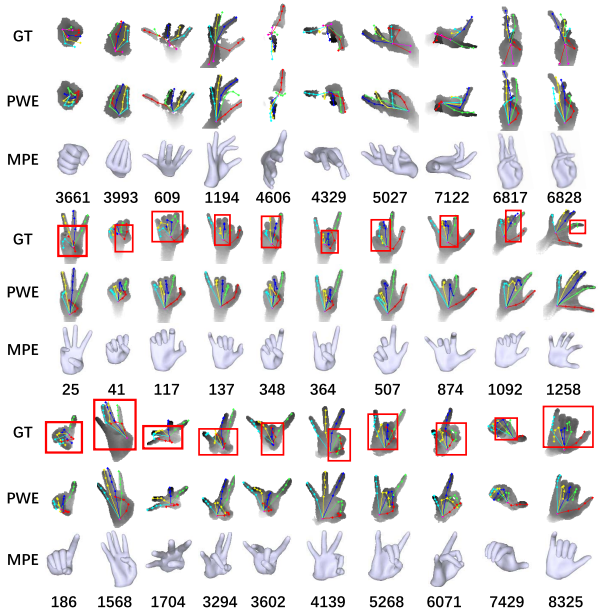

Fig. 9. Qualitative results for NYU, ICVL, and MSRA datasets from top to bottom. The number represents the frame ID in the test set for NYU, ICVL, and MSRA datasets.

F. Comparisons With State-of-the-Arts on Gesture Recognition

Accurate and robust 3D hand pose estimation plays an important role in human intention understanding and analysis, such as gesture recognition. In this section, we use DSF to perform self-supervised learning on gesture recognition datasets SHREC and DHG, and regenerate 3D skeleton coordinates for these two datasets. We adopt a SOTA skeleton-based gesture recognition method, DG-STA [6], as the baseline model and use the 3D skeleton coordinates generated by DSF for training and testing. As shown in Table VIII, using the 3D skeleton coordinates generated by DSF can significantly improve the accuracy of gesture recognition on these two datasets. For SHREC, DG-STA-DSF obtains 96.8% on the 14-gesture protocol and 95.0% on the 28-gesture protocol. For DHG dataset, DG-STA-DSF obtains 96.3% on the 14-gesture protocol and 95.9% on the 28-gesture protocol. In particular, for the most challenging setting, i.e., DHG with the 28-gesture protocol, DG-STA-DSF improves 7 points compared with the previous SOTA gesture recognition method HPEV [5]. This shows that our method can generate much more accurate bone information than the Intel RealSense SDK. This also demonstrates the importance of accurate 3D pose results for skeleton-based gesture recognition.

At the same time, we adopt a SOTA supervised 3D hand pose estimation algorithm (AWR) to regenerate 3D skeleton coordinates for SHREC and DHG datasets. Specifically, we adopt ResNet-50 as the backbone and perform supervised training on the BigHand 2.2M hand dataset [25]. As shown in Table VIII, our method consistently outperforms DG-STA-AWR, especially on more challenging DHG datasets. This reflects the superiority of our method, that is, it can quickly and efficiently estimate the hand pose of the new scenes, such as new depth cameras, perspectives, and subjects.

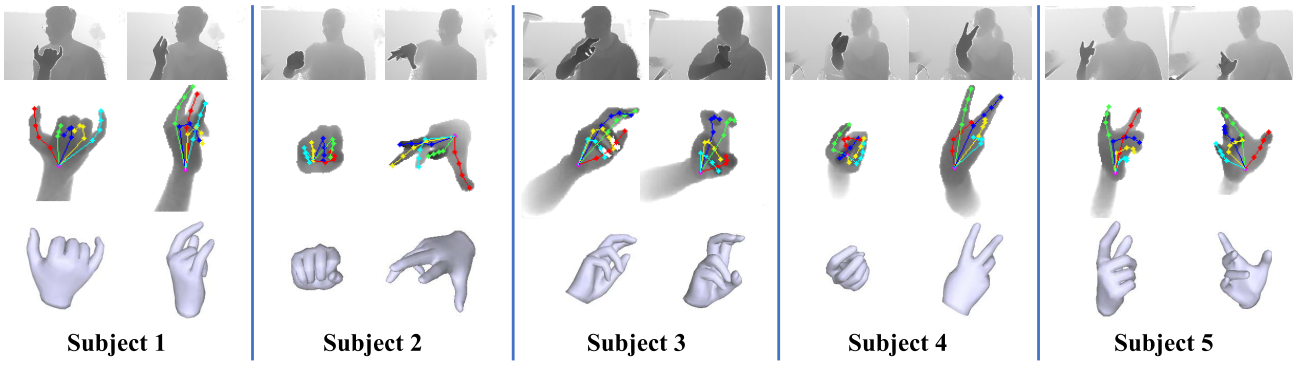

Fig. 10. Qualitative results for testing in real scenarios on different subjects with different hand sizes and hand poses. For each subject, the first line is the depth images captured by the KinectV2; the second line is the cropped hand depth images and the estimated hand pose; the third line is the estimated hand mesh.

TABLE VIII

THE COMPARISON OF THE RECOGNITION ACCURACY (%) WITH THE SOTA APPROACHES ON SHREC AND DHG DATASETS. THE UPWARD ARROWS REPRESENT THE IMPROVEMENT COMPARED TO DG-STA

Method	Modality	SHREC		DHG	
		14	28	14	28
PointLSTM [112]	Point clouds	95.9	94.7	-	-
Res-TCN [2]	Skeleton	91.1	87.3	86.9	83.6
ST-GCN [3]	Skeleton	92.7	87.7	91.2	87.1
STA-Res-TCN [2]	Skeleton	93.6	90.7	89.2	85.0
ST-TS-HGR-NET [4]	Skeleton	94.3	89.4	87.3	83.4
HPEV [5]	Skeleton	94.9	92.3	92.5	88.9
DG-STA [6]	Skeleton	94.4	90.7	91.9	88.0
DG-STA-AWR	Skeleton	$96.3\uparrow_{1.9}$	$93.3\uparrow_{2.6}$	$94.5\uparrow_{2.6}$	$92.1\uparrow_{4.1}$
DG-STA-DSF	Skeleton	$\mathbf{96.8}\uparrow_{2.4}$	$\mathbf{95.0}\uparrow_{4.3}$	$\mathbf{96.3}\uparrow_{4.4}$	$\mathbf{95.9}\uparrow_{7.9}$

G. Qualitative Results and Runtime

Some qualitative results for NYU, ICVL, and MSRA datasets are shown in Fig. 9. As shown in the estimated results for NYU, DSF can maintain the rationality of hand pose when fingers contact intensively, such as clenching (3361) and fingers close together (3993). Meanwhile, DSF is robust to extreme perspective (609, 4606) and the depth holes (1194, 4329). Although the prediction of the missing region is not completely consistent with the ground truth, it is still reasonable. For some complex poses (5027, 7122, 6817), DSF can also predict the accurate hand pose. Our method fails in the case of the complex pose with partial depth missing (6828). For ICVL and MSRA, we show that DSF is able to produce more accurate and reasonable results for the mislabeled samples. We also provide supplementary videos to completely compare our results with the annotations on the entire test set. We also evaluate our method in real scenarios with Kinect V2. As shown in Fig. 10, we experiment on five subjects, including four men and one woman, who have different hand sizes and hand poses. As can be seen, our method is tolerant of different hand poses and hand sizes. This experiment further demonstrates the excellent generalization ability of our method for the new scenarios.

Our model can switch freely for different computing devices and different scenarios during testing. Specifically, when we only need the 3D hand pose, we can only adopt the PWE branch. At this time, our method can reach 200 FPS with a batch size 1 on an NVIDIA RTX 3090 GPU. When we need the 3D hand pose and mesh at the same time, we use the two branches at the same time, and our method can also reach 105 FPS. When we need more accurate pose and mesh results, such as generating annotations for other tasks, we can adopt the refinement stage and our method can still reach 27 FPS.

V. CONCLUSION

In this paper, we propose a Dual-branch Self-boosting Framework (DSF) to achieve accurate, robust and flexible 3D hand pose and model estimation without using any labeled real data. Firstly, we propose an image-to-image translation technology to reduce the domain gap between synthetic data and real data, which significantly improves the effect of network pre-training. Secondly, we propose a dual-branch self-boosting network that can maintain the accuracy of 3D pose estimation and the robustness and flexibility of 3D hand model estimation simultaneously. Through an inter-branch loss and a part-aware model-fitting loss, we fully explore the advantages of the dual-branch structure. The two branches can promote each other continuously during the self-supervised training on unlabeled real data. Finally, we adopt a refinement stage to make better use of the prior structure information of the hand model, which further improves the accuracy and robustness of the estimation. Our method achieves comparable results to state-of-the-art fully supervised methods and shows better generalization performance. Our method outperforms previous self-supervised methods without using paired multi-view images. In addition, our method greatly improves the accuracy of skeleton-based gesture recognition, which shows that our method has strong application potential in downstream tasks.

REFERENCES

[1] Q. Xiao, M. Qin, and Y. Yin, "Skeleton-based Chinese sign language recognition and generation for bidirectional communication between deaf and hearing people," *Neural Netw.*, vol. 125, pp. 41–55, May 2020.
[2] J. Hou, G. Wang, X. Chen, J.-H. Xue, R. Zhu, and H. Yang, "Spatial-temporal attention Res-TCN for skeleton-based dynamic hand gesture recognition," in *Proc. Eur. Conf. Comput. Workshops*, 2018, pp. 1–15.
[3] S. Yan, Y. Xiong, and D. Lin, "Spatial temporal graph convolutional networks for skeleton-based action recognition," in *Proc. AAAI Conf. Artif. Intell.*, 2018, pp. 7444–7452.
[4] X. S. Nguyen, L. Brun, O. Lezoray, and S. Bougleux, "A neural network based on SPD manifold learning for skeleton-based hand gesture recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 12036–12045.

[5] J. Liu, Y. Liu, Y. Wang, V. Prinet, S. Xiang, and C. Pan, "Decoupled representation learning for skeleton-based gesture recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 5751–5760.

[6] Y. Chen, L. Zhao, X. Peng, J. Yuan, and D. N. Metaxas, "Construct dynamic graphs for hand gesture recognition via spatial-temporal attention," in *Proc. Brit. Mach. Vis. Conf.*, 2019, pp. 1–13.

[7] D. Tang, H. J. Chang, A. Tejani, and T.-K. Kim, "Latent regression forest: Structured estimation of 3D articulated hand posture," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 3786–3793.

[8] C. Xu, L. N. Govindarajan, Y. Zhang, and L. Cheng, "*Lie-X*: Depth image based articulated object pose estimation, tracking, and action recognition on lie groups," *Int. J. Comput. Vis.*, vol. 123, no. 3, pp. 454–478, Jul. 2017.

[9] H. Guo, G. Wang, X. Chen, C. Zhang, F. Qiao, and H. Yang, "Region ensemble network: Improving convolutional network for hand pose estimation," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2017, pp. 4512–4516.

[10] X. Chen, G. Wang, H. Guo, and C. Zhang, "Pose guided structured region ensemble network for cascaded hand pose estimation," *Neurocomputing*, vol. 395, pp. 138–149, Jun. 2020.

[11] M. Oberweger and V. Lepetit, "DeepPrior++: Improving fast and accurate 3D hand pose estimation," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops (ICCVW)*, Oct. 2017, pp. 585–594.

[12] M. Oberweger, P. Wohlhart, and V. Lepetit, "Hands deep in deep learning for hand pose estimation," in *Proc. Comput. Vis. Winter Workshop*, 2015, pp. 21–30.

[13] L. Ge, H. Liang, J. Yuan, and D. Thalmann, "Real-time 3D hand pose estimation with 3D convolutional neural networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 4, pp. 956–970, Apr. 2019.

[14] L. Ge, Y. Cai, J. Weng, and J. Yuan, "Hand PointNet: 3D hand pose estimation using point sets," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8417–8426.

[15] J. Tompson, M. Stein, Y. Lecun, and K. Perlin, "Real-time continuous pose recovery of human hands using convolutional networks," *ACM Trans. Graph.*, vol. 33, no. 5, pp. 169:1–169:10, Sep. 2014.

[16] L. Ge, H. Liang, J. Yuan, and D. Thalmann, "Robust 3D hand pose estimation in single depth images: From single-view CNN to multi-view CNNs," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 3593–3601.

[17] J. Y. Chang, G. Moon, and K. M. Lee, "V2V-PoseNet: Voxel-to-voxel prediction network for accurate 3D hand and human pose estimation from a single depth map," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 5079–5088.

[18] C. Wan, T. Probst, L. V. Gool, and A. Yao, "Dense 3D regression for hand pose estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 5147–5156.

[19] L. Ge, Z. Ren, and J. Yuan, "Point-to-point regression PointNet for 3D hand pose estimation," in *Proc. Eur. Conf. Comput. Vision (ECCV)*, Sep. 2018, pp. 475–491.

[20] W. Huang, P. Ren, J. Wang, Q. Qi, and H. Sun, "AWR: Adaptive weighting regression for 3D hand pose estimation," in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 11061–11068.

[21] F. Xiong *et al.*, "A2J: Anchor-to-joint regression network for 3D articulated pose estimation from a single depth image," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 793–802.

[22] X. Deng *et al.*, "Weakly supervised learning for single depth-based hand shape recovery," *IEEE Trans. Image Process.*, vol. 30, pp. 532–545, 2021.

[23] J. Malik *et al.*, "DeepHPS: End-to-end estimation of 3D hand pose and shape by learning from synthetic depth," in *Proc. Int. Conf. 3D Vis. (3DV)*, Sep. 2018, pp. 110–119.

[24] J. Malik *et al.*, "HandVoxNet: Deep voxel-based network for 3D hand shape and pose estimation from a single depth map," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 7113–7122.

[25] S. Yuan, Q. Ye, B. Stenger, S. Jain, and T.-K. Kim, "BigHand2.2 M benchmark: Hand pose dataset and state of the art analysis," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 4866–4874.

[26] J. Wang, F. Mueller, F. Bernard, and C. Theobalt, "Generative model-based loss to the rescue: A method to overcome annotation errors for depth-based hand pose estimation," in *Proc. 15th IEEE Int. Conf. Autom. Face Gesture Recognit. (FG)*, Nov. 2020, pp. 93–100.

[27] J. S. Supancic, G. Rogez, Y. Yang, J. Shotton, and D. Ramanan, "Depth-based hand pose estimation: Data, methods, and challenges," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1868–1876.

[28] M. Oberweger, G. Riegler, P. Wohlhart, and V. Lepetit, "Efficiently creating 3D training data for fine hand pose estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 4957–4965.

[29] E. Dibra, T. Wolf, C. Oztireli, and M. Gross, "How to refine 3D hand pose estimation from unlabelled depth data?" in *Proc. Int. Conf. 3D Vis. (3DV)*, Oct. 2017, pp. 135–144.

[30] C. Wan, T. Probst, L. Van Gool, and A. Yao, "Self-supervised 3D hand pose estimation through training by fitting," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 10853–10862.

[31] C. Wan, T. Probst, L. Van Gool, and A. Yao, "Dual grid net: Hand mesh vertex regression from single depth maps," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2020, pp. 442–459.

[32] X. Sun, Y. Wei, S. Liang, X. Tang, and J. Sun, "Cascaded hand pose regression," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 824–832.

[33] Q. De Smedt, H. Wannous, J.-P. Vandeborre, J. Guerry, B. Le Saux, and D. Filliat, "SHREC'17 track: 3D hand gesture recognition using a depth and skeletal dataset," in *Proc. 10th Eurograph. Workshop 3D Object Retr.*, 2017, pp. 1–6.

[34] Q. De Smedt, H. Wannous, and J.-P. Vandeborre, "Skeleton-based dynamic hand gesture recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2016, pp. 1–9.

[35] C. Qian, X. Sun, Y. Wei, X. Tang, and J. Sun, "Realtime and robust hand tracking from depth," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1106–1113.

[36] A. Tkach, M. Pauly, and A. Tagliasacchi, "Sphere-meshes for real-time hand modeling and tracking," *ACM Trans. Graph.*, vol. 35, no. 6, pp. 222:1–222:11, 2016.

[37] T. Sharp *et al.*, "Accurate, robust, and flexible real-time hand tracking," in *Proc. 33rd Annu. ACM Conf. Hum. Factors Comput. Syst.*, Apr. 2015, pp. 3633–3642.

[38] S. Khamis, J. Taylor, J. Shotton, C. Keskin, S. Izadi, and A. Fitzgibbon, "Learning an efficient model of hand shape variation from depth images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 2540–2548.

[39] J. Taylor *et al.*, "Efficient and precise interactive hand tracking through joint, continuous optimization of pose and correspondences," *ACM Trans. Graph.*, vol. 35, no. 4, pp. 143:1–143:12, 2016.

[40] M. Ye and R. Yang, "Real-time simultaneous pose and shape estimation for articulated objects using a single depth camera," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 2345–2352.

[41] S. Sridhar, F. Mueller, A. Oulasvirta, and C. Theobalt, "Fast and robust hand tracking using detection-guided optimization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3213–3221.

[42] I. Oikonomidis, N. Kyriazis, and A. Argyros, "Efficient model-based 3D tracking of hand articulations using Kinect," in *Proc. Brit. Mach. Vis. Conf.*, 2011, pp. 1–11.

[43] A. Tagliasacchi, M. Schröder, A. Tkach, S. Bouaziz, M. Botsch, and M. Pauly, "Robust articulated-ICP for real-time hand tracking," *Comput. Graph. Forum*, vol. 34, no. 5, pp. 101–114, 2015.

[44] L. Ballan, A. Taneja, J. Gall, L. Van Gool, and M. Pollefeys, "Motion capture of hands in action using discriminative salient points," in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 640–653.

[45] M. de La Gorce, D. J. Fleet, and N. Paragios, "Model-based 3D hand pose estimation from monocular video," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 9, pp. 1793–1805, Sep. 2011.

[46] D. Tang, T.-H. Yu, and T.-K. Kim, "Real-time articulated hand pose estimation using semi-supervised transductive regression forests," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2013, pp. 3224–3231.

[47] H. Liang, J. Yuan, J. Lee, L. Ge, and D. Thalmann, "Hough forest with optimized leaves for global hand pose estimation with arbitrary postures," *IEEE Trans. Cybern.*, vol. 49, no. 2, pp. 527–541, Feb. 2019.

[48] M. Oberweger, P. Wohlhart, and V. Lepetit, "Training a feedback loop for hand pose estimation," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 3316–3324.

[49] M. Oberweger, P. Wohlhart, and V. Lepetit, "Generalized feedback loop for joint hand-object pose estimation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 8, pp. 1898–1912, Aug. 2020.

[50] X. Zhou, Q. Wan, W. Zhang, X. Xue, and Y. Wei, "Model-based deep hand pose estimation," in *Proc. Int. Joint Conf. Artif. Intell.*, 2016, pp. 2421–2427.

[51] Y. Zhou, J. Lu, K. Du, X. Lin, Y. Sun, and X. Ma, "HBE: Hand branch ensemble network for real-time 3D hand pose estimation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 501–516.

[52] P. Ren, H. Sun, Q. Qi, J. Wang, and W. Huang, "SRN: Stacked regression network for real-time 3D hand pose estimation," in *Proc. Brit. Mach. Vis. Conf.*, 2019, p. 112.

[53] L. Ge, H. Liang, J. Yuan, and D. Thalmann, "3D convolutional neural networks for efficient and robust hand pose estimation from single depth images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1991–2000.

[54] L. Huang, J. Tan, J. Liu, and J. Yuan, "Hand-transformer: Non-autoregressive structured modeling for 3D hand pose estimation," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2020, pp. 17–33.

[55] L. Ge, H. Liang, J. Yuan, and D. Thalmann, "Robust 3D hand pose estimation from single depth images using multi-view CNNs," *IEEE Trans. Image Process.*, vol. 27, no. 9, pp. 4422–4436, Sep. 2018.

[56] X. Chen, G. Wang, C. Zhang, T.-K. Kim, and X. Ji, "SHPR-NET: Deep semantic hand pose regression from point clouds," *IEEE Access*, vol. 6, pp. 43425–43439, 2018.

[57] S. Li and D. Lee, "Point-to-pose voting based hand pose estimation using residual permutation equivariant layer," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 11927–11936.

[58] X. Zhang and F. Zhang, "Differentiable spatial regression: A novel method for 3D hand pose estimation," *IEEE Trans. Multimedia*, vol. 24, pp. 166–176, 2022.

[59] L. Fang, X. Liu, L. Liu, H. Xu, and W. Kang, "JGR-P2O: Joint graph reasoning based pixel-to-offset prediction network for 3D hand pose estimation from a single depth image," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2020, pp. 120–137.

[60] S. Sridhar, F. Mueller, M. Zollhöfer, D. Casas, A. Oulasvirta, and C. Theobalt, "Real-time joint tracking of a hand manipulating an object from RGB-D input," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 294–310.

[61] L. Ge *et al.*, "3D hand shape and pose estimation from a single RGB image," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 10833–10842.

[62] A. Boukhayma, R. de Bem, and P. H. S. Torr, "3D hand shape and pose from images in the wild," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 10843–10852.

[63] Y. Zhou, M. Habermann, W. Xu, I. Habibie, C. Theobalt, and F. Xu, "Monocular real-time hand shape and motion capture using multi-modal data," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 5346–5355.

[64] J. Wang *et al.*, "RGB2Hands: Real-time tracking of 3D hand interactions from monocular RGB video," *ACM Trans. Graph.*, vol. 39, no. 6, pp. 1–16, Dec. 2020.

[65] Y. Chen *et al.*, "Joint hand-object 3D reconstruction from a single image with cross-branch feature fusion," *IEEE Trans. Image Process.*, vol. 30, pp. 4008–4021, 2021.

[66] S. Baek, K. I. Kim, and T.-K. Kim, "Pushing the envelope for RGB-based dense 3D hand pose estimation via neural rendering," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 1067–1076.

[67] Y. Cai, L. Ge, J. Cai, and J. Yuan, "Weakly-supervised 3D hand pose estimation from monocular RGB images," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 666–682.

[68] X. Deng *et al.*, "Hand pose understanding with large-scale photo-realistic rendering dataset," *IEEE Trans. Image Process.*, vol. 30, pp. 4275–4290, 2021.

[69] C. Wan, T. Probst, L. Van Gool, and A. Yao, "Crossing nets: Combining GANs and VAEs with a shared latent space for hand pose estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 680–689.

[70] M. Abdi, E. Abbasnejad, C. P. Lim, and S. Nahavandi, "3D hand pose estimation using simulation and partial-supervision with a shared latent space," in *Proc. Brit. Mach. Vis. Conf.*, 2018, pp. 1–17.

[71] M. Rad, M. Oberweger, and V. Lepetit, "Feature mapping for learning fast and accurate 3D pose inference from synthetic images," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4663–4672.

[72] G. Poier, M. Opitz, D. Schinagl, and H. Bischof, "MURAUER: Mapping unlabeled real data for label AUstERity," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2019, pp. 1393–1402.

[73] A. Shrivastava, T. Pfister, O. Tuzel, J. Susskind, W. Wang, and R. Webb, "Learning from simulated and unsupervised images through adversarial training," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2107–2116.

[74] F. Mueller *et al.*, "Ganerated hands for real-time 3D hand tracking from monocular RGB," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 49–59.

[75] L. Chen *et al.*, "TAGAN: Tonality-alignment generative adversarial networks for realistic hand pose synthesis," in *Proc. Brit. Mach. Vis. Conf.*, 2019, pp. 1–15.

[76] Z. Wu *et al.*, "MM-Hand: 3D-aware multi-modal guided hand generation for 3D hand pose synthesis," in *Proc. 28th ACM Int. Conf. Multimedia*, Oct. 2020, pp. 2508–2516.

[77] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2223–2232.

[78] K. Bousmalis, N. Silberman, D. Dohan, D. Erhan, and D. Krishnan, "Unsupervised pixel-level domain adaptation with generative adversarial networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 3722–3731.

[79] J. Hoffman *et al.*, "CyCADA: Cycle-consistent adversarial domain adaptation," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 1989–1998.

[80] M. Qi, E. Remelli, M. Salzmann, and P. Fua, "Unsupervised domain adaptation with temporal-consistent self-training for 3D hand-object joint reconstruction," 2020, *arXiv:2012.11260*.

[81] Y. Li, L. Yuan, and N. Vasconcelos, "Bidirectional learning for domain adaptation of semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 6936–6945.

[82] M. Long, Y. Cao, J. Wang, and M. Jordan, "Learning transferable features with deep adaptation networks," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 97–105.

[83] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, "Adversarial discriminative domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 7167–7176.

[84] Y. Ganin *et al.*, "Domain-adversarial training of neural networks," *J. Mach. Learn. Res.*, vol. 17, no. 1, pp. 2030–2096, Jan. 2016.

[85] Y.-H. Tsai, W.-C. Hung, S. Schulter, K. Sohn, M.-H. Yang, and M. Chandraker, "Learning to adapt structured output space for semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7472–7481.

[86] Y. Luo, L. Zheng, T. Guan, J. Yu, and Y. Yang, "Taking a closer look at domain shift: Category-level adversaries for semantics consistent domain adaptation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 2507–2516.

[87] J. Liang, R. He, Z. Sun, and T. Tan, "Exploring uncertainty in pseudo-label guided unsupervised domain adaptation," *Pattern Recognit.*, vol. 96, Dec. 2019, Art. no. 106996.

[88] Q. Wang and T. Breckon, "Unsupervised domain adaptation via structured prediction based selective pseudo-labeling," in *Proc. AAAI Conf. Artif. Intell.*, 2020, vol. 34, no. 4, pp. 6243–6250.

[89] J. Mu, W. Qiu, G. D. Hager, and A. L. Yuille, "Learning from synthetic animals," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 12386–12395.

[90] Z. Zheng and Y. Yang, "Rectifying pseudo label learning via uncertainty estimation for domain adaptive semantic segmentation," *Int. J. Comput. Vis.*, vol. 129, no. 4, pp. 1106–1120, 2021.

[91] Y. Zhang, P. David, H. Foroosh, and B. Gong, "A curriculum domain adaptation approach to the semantic segmentation of urban scenes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 8, pp. 1823–1841, Aug. 2020.

[92] F. Pan, I. Shin, F. Rameau, S. Lee, and I. S. Kweon, "Unsupervised intra-domain adaptation for semantic segmentation through self-supervision," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 3764–3773.

[93] Y. Sun, E. Tzeng, T. Darrell, and A. A. Efros, "Unsupervised domain adaptation through self-supervision," 2019, *arXiv:1909.11825*.

[94] L. Jing and Y. Tian, "Self-supervised visual feature learning with deep neural networks: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 11, pp. 4037–4058, Nov. 2020.

[95] J. N. Kundu, S. Seth, V. Jampani, M. Rakesh, R. V. Babu, and A. Chakraborty, "Self-supervised 3D human pose estimation via part guided novel image synthesis," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 6152–6162.

[96] A. Tewari *et al.*, "Self-supervised multi-level face model learning for monocular reconstruction at over 250 Hz," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2549–2559.

[97] G. Wang, F. Manhardt, J. Shao, X. Ji, N. Navab, and F. Tombari, "Self6D: Self-supervised monocular 6D object pose estimation," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2020, pp. 108–125.

[98] H.-Y. Tung, H.-W. Tung, E. Yumer, and K. Fragkiadaki, "Self-supervised learning of motion capture," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1–11.
[99] Y. Chen, F. Wu, Z. Wang, Y. Song, Y. Ling, and L. Bao, "Self-supervised learning of detailed 3D face reconstruction," *IEEE Trans. Image Process.*, vol. 29, pp. 8696–8705, 2020.
[100] Y. Wen, W. Liu, B. Raj, and R. Singh, "Self-supervised 3D face reconstruction via conditional estimation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 13289–13298.
[101] F. Manhardt *et al.*, "CPS++: Improving class-level 6D pose and shape estimation from monocular images with self-supervised learning," 2020, *arXiv:2003.05848*.
[102] Y. Chen *et al.*, "Model-based 3D hand reconstruction via self-supervised learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 10451–10460.
[103] C. Xu, A. Nanjappa, X. Zhang, and L. Cheng, "Estimate hand poses efficiently from single depth images," *Int. J. Comput. Vis.*, vol. 116, no. 1, pp. 21–45, Jan. 2016.
[104] J. Romero, D. Tzionas, and M. J. Black, "Embodied hands: Modeling and capturing hands and bodies together," *ACM Trans. Graph.*, vol. 36, no. 6, pp. 245:1–245:17, 2017.
[105] A. Armagan *et al.*, "Measuring generalisation to unseen viewpoints, articulations, shapes and objects for 3D hand pose estimation under hand-object interaction," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 85–101.
[106] N. Ravi *et al.*, "Accelerating 3D deep learning with PyTorch3D," 2020, *arXiv:2007.08501*.
[107] B. Xiao, H. Wu, and Y. Wei, "Simple baselines for human pose estimation and tracking," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 466–481.
[108] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2015, *arXiv:1412.6980*.
[109] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *Proc. Int. Conf. Learn. Represent.*, 2019, pp. 1–19.
[110] K. Du, X. Lin, Y. Sun, and X. Ma, "CrossInfoNet: Multi-task information sharing based hand pose estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 9896–9905.
[111] Y. Chen, Z. Tu, L. Ge, D. Zhang, R. Chen, and J. Yuan, "SO-HandNet: Self-organizing network for 3D hand pose estimation with semi-supervised learning," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 6961–6970.
[112] Y. Min, Y. Zhang, X. Chai, and X. Chen, "An efficient PointLSTM for point clouds based gesture recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 5761–5770.

Pengfei Ren received the B.S. degree from the Beijing University of Posts and Telecommunications in 2018, where he is currently pursuing the Ph.D. degree. His research interests include deep learning, hand pose estimation, and gesture recognition.

Haifeng Sun received the Ph.D. degree from the Beijing University of Posts and Telecommunications in 2017. He is currently a Lecture with the Beijing University of Posts and Telecommunications. His research interests include data mining, information retrieval, and next generation networks.

Jiachang Hao received the bachelor's degree from the Beijing University of Posts and Telecommunications in 2018, where he is currently pursuing the postgraduate degree. His research interests include video understanding, action recognition, and object detection.

Qi Qi (Senior Member, IEEE) received the Ph.D. degree from the Beijing University of Posts and Telecommunications, in 2010. She is currently an Associate Professor with the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications. She has published more than 30 articles in international journals. She received two awards from the National Natural Science Foundations of China. Her research interests include ubiquitous services, deep learning, transfer learning, deep reinforcement learning, edge computing, and the Internet of Things.

Jingyu Wang (Senior Member, IEEE) received the Ph.D. degree from the Beijing University of Posts and Telecommunications, Beijing, China, in 2008. He is currently a Professor with the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications. He was a recipient of four awards from the National Natural Science Foundations of China. He has published more than 50 papers in international journal and famous conferences, including the *IEEE Communications Magazine* (COMM), IEEE TRANSACTIONS ON SERVICES COMPUTING (TSC), IEEE TRANSACTIONS ON MULTIMEDIA (TMM), IEEE TRANSACTIONS ON CLOUD COMPUTING (TCC), IEEE TRANSACTIONS ON AUDIO, SPEECH, AND LANGUAGE PROCESSING (TASLP), IEEE TRANSACTIONS ON EMERGING TOPICS IN COMPUTING (TETC), IEEE COMPUTER VISION AND PATTERN RECOGNITION (CVPR), Association for the Advancement of Artificial Intelligence (AAAI), Association for Computational Linguistics (ACL), and International Conference on Data Engineering (ICDE). His main research interests include broad aspects of intelligent networks, edge computing, machine learning, knowledge-defined networks, the Internet of Vehicles, and intent-based networks.

Jianxin Liao received the Ph.D. degree from the University of Electronics Science and Technology of China in 1996. He is currently the Dean of the Network Intelligence Research Center and a Full Professor with the State Key laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications. He has published more than 100 research papers and several books, and has been granted dozens of patents for inventions. He has won a number of prizes, which include the Premier's Award of Distinguished Young Scientists from the National Natural Science Foundation of China in 2005, and the Specially-Invited Professor of the "Yangtse River Scholar Award Program" by the China Ministry of Education in 2009. His main creative contributions include mobile intelligent networks, service network intelligent, networking architectures and protocols, and multimedia communication. These achievements were conferred the "National Prize for Progress in Science and Technology" twice in 2004 and 2009, respectively.