arXiv:2307.06038v2 [cs.CV] 10 Apr 2024

Pyramid Deep Fusion Network for Two-Hand Reconstruction from RGB-D Images

Jinwei Ren, and Jianke Zhu, *Senior Member, IEEE*

***Abstract*—Accurately recovering the dense 3D mesh of both hands from monocular images poses considerable challenges due to occlusions and projection ambiguity. Most of the existing methods extract features from color images to estimate the root-aligned hand meshes, which neglect the crucial depth and scale information in the real world. Given the noisy sensor measurements with limited resolution, depth-based methods predict 3D keypoints rather than a dense mesh. These limitations motivate us to take advantage of these two complementary inputs to acquire dense hand meshes on a real-world scale. In this work, we propose an end-to-end framework for recovering dense meshes for both hands, which employ single-view RGB-D image pairs as input. The primary challenge lies in effectively utilizing two different input modalities to mitigate the blurring effects in RGB images and noises in depth images. Instead of directly treating depth maps as additional channels for RGB images, we encode the depth information into the unordered point cloud to preserve more geometric details. Specifically, our framework employs ResNet50 and PointNet++ to derive features from RGB and point cloud, respectively. Additionally, we introduce a novel pyramid deep fusion network (PDFNet) to aggregate features at different scales, which demonstrates superior efficacy compared to previous fusion strategies. Furthermore, we employ a GCN-based decoder to process the fused features and recover the corresponding 3D pose and dense mesh. Through comprehensive ablation experiments, we have not only demonstrated the effectiveness of our proposed fusion algorithm but also outperformed the state-of-the-art approaches on publicly available datasets. To reproduce the results, we will make our source code and models publicly available at https://github.com/zijinxuxu/PDFNet.**

***Index Terms*—RGB-D fusion, 3D reconstruction, hand pose, end-to-end network.**

I. Introduction

Recovering the 3D pose and shape of human hands from a single viewpoint plays a pivotal role in a multitude of real-world applications, such as human-computer interaction [1], mixed reality [2], action recognition [3], and simulation. Over the past two decades, extensive research [4]–[13] has emerged in the field of hand reconstruction with various inputs including single color images, RGB-D images with depth maps, multi-view images, and video sequences. Due to the inherent complexity of finger joints, self-occlusions, and motion blur, an ongoing endeavor is to effectively address the challenges in 3D hand reconstruction.

At present, the prevailing methods [9], [14]–[16] for hand reconstruction predominantly focus on directly estimating both hands from a single RGB image. However, these methods encounter difficulties in real-world scenarios characterized by cluttered backgrounds, lighting variations, and motion blur, which limit their performance to environments similar to the training data. Generally, a conventional framework [4] separates detection and reconstruction, which requires extracting the hand region from the image by an off-the-shelf detector before feeding it to the reconstruction model. Consequently, these models only predict root-aligned 3D hand meshes. Instead, depth map-based methods [17], [18] often incorporate range maps as auxiliary supervisory information to compensate for inherent noise and limited resolution. Additionally, certain approaches [6], [19] employ depth maps to predict sparse 3D keypoints. The absolute scale information in depth maps is not affected by background changes as well as the rich foreground features in RGB maps, which is crucial to hand reconstruction. Fig. 1 presents a visual comparison between utilizing solely RGB input and augmenting it with depth map input.

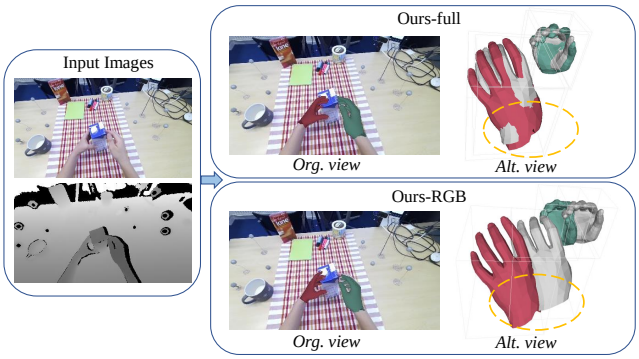

Fig. 1. Comparison between Ours-full method and Ours-RGB method. Although the results of the two methods are very similar under the original projection perspective, there is a large misalignment of the latter in the depth direction under the new perspective.

The previous fusion methods [20]–[22] have primarily relied on RGB-D cameras, which leverage both rich image information and depth measurements to accomplish tasks such as object detection and semantic segmentation. Despite extensive research efforts over an extended period, an effective fusion scheme for hand reconstruction remains elusive. This challenge can be attributed to the highly nonlinear nature of gestures [23] and the inherent variations between hands, making it arduous to achieve satisfactory results through a straightforward combination of color images and depth maps. In certain scenarios, utilizing depth maps alone can actually yield superior outcomes [24]. Hence, it becomes imperative to ascertain an effective fusion strategy specifically tailored for hand reconstruction tasks.

The simplest and most rudimentary fusion method entails directly incorporating the depth map as an additional channel

Jinwei Ren and Jianke Zhu are both with the College of Computer Science and Technology, Zhejiang University, Zheda Rd 38th, Hangzhou, China. Email: {zijinxuxu, jkzhu}@zju.edu.cn;
Jianke Zhu is the Corresponding Author.

alongside the RGB image [25]. This approach merely requires modifying the input channel of the model from three to four channels. However, the performance enhancement achieved by this simple fusion method remains quite limited. An alternative fusion approach that has gained popularity is operating at the feature level [24], [26], [27]. It is important to note that directly concatenating the two features obtained from shallow CNNs does not yield any performance improvements [24]. Accordingly, researchers have made endeavors to extract multi-scale depth features [26] or perform cross-fusion at intermediate feature layers [27].

The aforementioned fusion methods are primarily employed for the cropped single-hand images, which are limited to predicting sparse 3D keypoints rather than dense meshes. Furthermore, these methods process depth maps into 2D images, disregarding their inherent 3D characteristics. Inspired by previous work in 3D object detection [20] and 6-DoF estimation [28], [29], we adopt a different approach by converting the depth map into an unordered point cloud, and then extract point features to fuse them with RGB features derived by CNNs. Experimental results indicate that this method yields the improved performance due to the more effective feature. Additionally, we argue that it is insufficient for learning local features by relying solely on fixed-sized point feature regression. Motivated by the architecture of PointNet++ [30], we introduce a pyramid feature fusion module that enable the integration of point cloud features and RGB features at their corresponding positions across multiple scales. Moreover, existing frameworks based on sparse 3D keypoints or root-aligned mesh estimation may fall short when attempting to achieve two-hand reconstruction in real-world interactive scenarios.

In order to address the aforementioned challenges, we present an end-to-end framework that incorporates RGB and depth information to accurately reconstruct a 3D mesh of both hands from RGB-D inputs. Unlike HandPointnet [6], our approach eliminates the need for normalizing the point cloud using oriented bounding boxes, thereby avoiding misalignment between the point cloud and color image while simplifying the process. To tackle the difficulty in learning local features, we suggest a pyramid structure feature fusion module called PDFNet, which facilitates the fusion of two features at different scales in order to enable the effective integration of information. Furthermore, we introduce an adaptive weight allocation module to achieve more robust and accurate fusion, which allocates weights to different features to mitigate interference from local unreliable regions.

To attain a more comprehensive representation of hand structures, as opposed to merely sparse 3D keypoints, we opt to employ a graph convolutional network (GCN) as our decoder as in [9]. Instead of directly using the image-wide features, we introduce a center map for precise hand localization. Additionally, we conduct experiments using the parameterized model MANO [5] and multiple fully connected layers as alternative decoders across various two-hand datasets. The results demonstrate the convincing performance enhancement achieved through our fusion algorithm.

From above all, the main contributions of our work can be summarized as follows.

(1) We propose an efficacious end-to-end single-stage framework that reconstructs 3D hand meshes from a solitary RGB-D input. To the best of our knowledge, this is the first RGB-D fusion framework for two-hand reconstruction.
(2) We devise a novel fusion module named PDFNet that effectively harnesses both color information and depth maps. Empirical studies validate the substantial enhancement that this module imparts upon the baseline model.
(3) Both quantitative and qualitative evaluations clearly demonstrate that our proposed approach achieves state-of-the-art performance on publicly available two-hand datasets [16], [31].

II. Related Work

Rapid progress has been made on hand pose estimation [4], [6], [32]–[35] and 3D hand mesh [5], [18], [36], [37] reconstruction over recent years, giving rise to various categories such as single-handed [7], [38] and multi-handed reconstruction [39], fully supervised [8], [40] and weakly supervised methods [41]–[43], etc. In this paper, our primary research focus lies in exploring different types of inputs. Consequently, previous studies can be classified into three distinct groups, namely color image, depth map, and RGB-D image.

A. Hand Reconstruction from Color Image

Due to the lack of depth information, it is very challenging to recover 3D hand pose from a single color image. Zimmermann *et al.* [4] trained a deep neural network to learn the 3D articulation prior of hands on a synthetic dataset. Guo *et al.* [38] proposed a feature interaction module to enhance the joint and skeleton feature. In addition to predicting 3D pose, Boukhayma *et al.* [7] further predicted the shape of the hand and optimized the 3D parameterized model MANO [5] through a re-projection module. To improve performance, the subsequent model-based methods introduced iterative optimization [44], neural rendering [45], spatial mesh convolution [41], adaptive 2D-1D registration [46], etc. In addition, novel image-to-pixel prediction networks [8], graph-convolution-reinforced transformer [40], and contrastive learning [42] have also been applied in this field. In order to address the scarcity of 3D annotations for real hands, Zimmermann *et al.* [47] proposed the first single-hand dataset containing 3D pose and shape labels. Hampali *et al.* [48] proposed a dataset with similar annotations, which focuses on hand-object interaction scenes. Considering the situation of multiple hands in a picture, multi-stage methods [49] [50] that separate hand detection and pose estimation, as well as single-stage methods [39] [10] that jointly detect and reconstruct, have been proposed. Moon *et al.* [51] proposed a large-scale real-captured interacting hand dataset using a multi-view system. Based on this dataset, several subsequent works [9], [14]–[16], [52] have conducted more in-depth research on left and right hand interaction and designed exquisite network structures to better extract features.

B. Hand Reconstruction from Depth Map

Compared to using only RGB images, it is more intuitive to recover the hand pose and shape from depth maps, as partial geometric information can be directly obtained. According to the different processing methods for input data, these methods can be roughly divided into two categories, including image-based methods and point cloud-based approaches. The former mostly directly employs CNNs to process depth maps like RGB images through feedback loop [53], dense per pixel compression [54], forward kinematics [55], adaptive weighting regression [56], and auxiliary latent variable [57], etc. The latter processes the depth map into a point cloud and directly extracts point features from it to regress the hand pose. Ge *et al.* [58] proposed 3D CNN for point feature extraction and regress full hand pose in volumetric representation. In order to effectively utilize information in depth images and reduce network parameters, Ge *et al.* [6] adopted the network structure of PointNet [30], [59] to extract point cloud features. The point cloud regularization module is also introduced to improve the robustness of the method. Subsequent work adopted similar frameworks while introducing the intermediate supervisory information such as heatmap and unit vector field [34], semantic segmentation [60] to enhance the performance of the model. With the deepening of research, permutation equivariant layer (PEL) [19], self-organizing map (SOM) [61] and Transformer [62] have also been introduced into hand pose estimation. As for interacting hand, Taylor *et al.* [63] trained a segmentation network to construct a 3D point cloud from depth maps and designed a signed distance field to minimize model fitting errors. Muller *et al.* [64] estimated a vertex-to-pixel correspondence map first and proposed an energy minimization framework, which can optimize the pose and shape parameters by fitting the point cloud. However, optimization-based model fitting methods rely more on precise 3D point cloud inputs. On the other hand, the learning-based methods may obtain more prior information from other depth maps to mitigate the impact of sensor noise, which has not yet been explored.

C. Hand Reconstruction from RGB-D Image

RGB-D fusion has been extensively studied in fields such as 3D object detection [20], object pose estimation [28], and semantic segmentation [22] [21]. However, there has been few in-depth research on hand reconstruction tasks. [66] [67] [68] requested RGB-D sensor as input while the RGB image was only used to segment the hand part in the depth map. Cai *et al.* [18] used depth maps as regularization terms during training to reduce dependence on 3D annotations, and only used RGB images during testing. Yuan *et al.* [69] pre-trained a depth-based network and froze the parameters of the network during joint RGB-D training. The gap between the RGB-based method and the depth-based approach is narrowed by minimizing the intermediate features of the two branches. Kazakos *et al.* [24] designed a double-stream architecture for RGB-D fusion, and tried input-level fusion, feature-level fusion, and score-level fusion. Unfortunately, their experiments indicated that adding RGB information did not help with performance gains. Mueller *et al.* [25] directly used the 4-channel RGB-D input and trained two CNNs to locate and regress the 3D position of the hand. They chose to project RGB pixels onto a depth map to obtain a colored depth map and then predicted the absolute coordinates of the hand center and the 3D offset of each joint separately. Lin *et al.* [26] scaled the depth map to multiple sizes to aggregated features at different resolutions, and then adopted feature attention structures [70] to fuse RGB features. Sun *et al.* [27] adopted a similar dual stream structure, where the depth map branch used a shallower network to avoid overfitting. The features of the two branches were first cross fused in the middle part, and then concatenated together. This resulted in better results compared to direct concatenation.

Each of the aforementioned approaches exhibits certain limitations. Primarily, they solely focus on regressing hand pose without undertaking shape reconstruction. While point cloud structures offer a more accurate depiction of geometric information compared to depth maps, there is a dearth of research in this area. Additionally, simply stitching the global features may pose challenges in effectively capturing local structures. A potential solution lies in multi-scale feature fusion. By taking into account these aforementioned limitations, we introduce a novel framework for dense hand mesh reconstruction, built upon our pyramid fusion module (PDFNet).

III. Methodology

The goal of this paper is to restore a dense 3D mesh of both hands within real-world scenes through a single RGB-D image. Our framework takes both RGB images and point cloud generated from depth maps as inputs, and extracts features using classic ResNet50 [65] and PointNet++ [30], respectively. Subsequently, the extracted features are fed into PDFNet for deep fusion to improve the performance of our model. The fused features are then fed into the GCN-based decoder to output the dense 3D mesh of both hands. By ingeniously fusing the modalities of RGB and depth, we are able to accurately reconstruct a 3D hand mesh with real depth and scale in camera space. For interactive scenarios in AR/VR applications, it is imperative to restore the absolute position within the camera coordinate system, surpassing the limitations of previous root-aligned outputs. Apart from the root position, the depth map also conveys the relative geometric relationship among hand joints, which yields significant contributions to the accuracy of reconstruction in local coordinate systems.

A. Overview

The overall structure of our approach is a classical encoder-decoder architecture, as depicted in Fig. 2. Our method can be divided into three integral components, including feature extraction, feature fusion, and feature decoding. Within the feature extraction module (Section III-B), we extract 2D image features utilizing ResNet50, while simultaneously extracting 3D point cloud features using PointNet++. In the feature fusion phase (Section III-C), the corresponding RGB features and point cloud features are fused at the pixel level through point cloud indexing. Finally, in the feature decoding phase

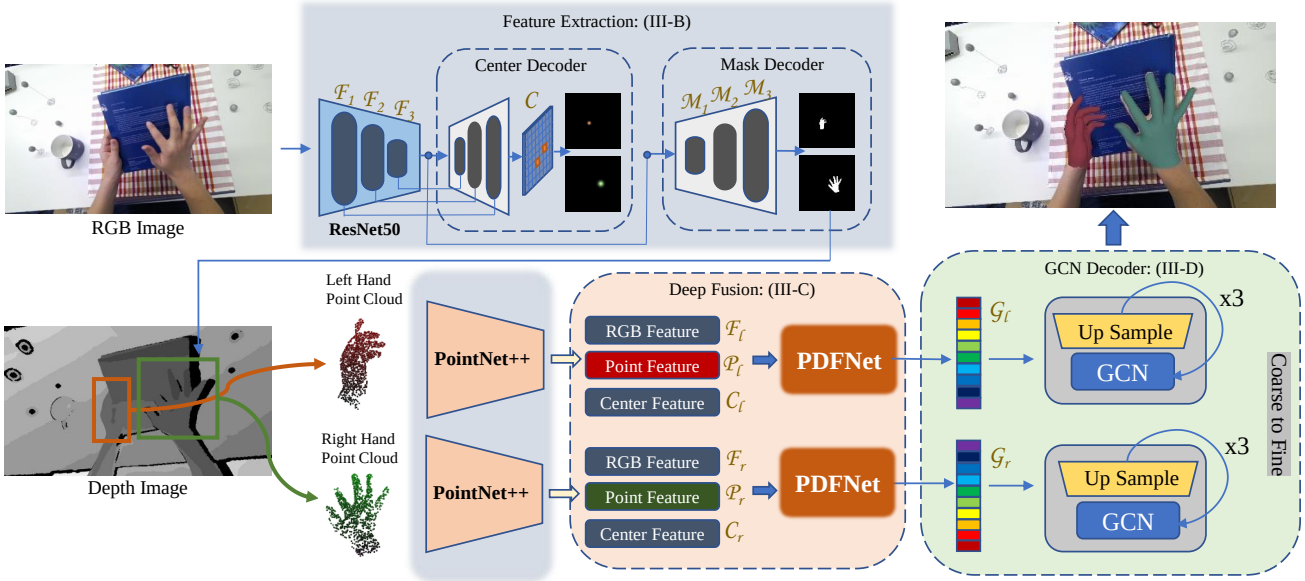

Fig. 2. Overview of the proposed framework. Given an RGB-D image, we adopt ResNet50 [65] and PointNet++ [30] as the backbone to extract features (Section III-B) and decode RGB features into center maps and masks using two simple decoders. The deep fusion module (Section III-C) is responsible for the deep fusion of RGB features and point features. The GCN-based decoder (Section III-D) takes the fused global feature and outputs dense hand mesh of both hands in a coarse to fine way. The whole pipeline is trained in an end-to-end manner.

(Section III-D), we employ multi-layer Graph Convolutional Networks (GCN) and upsampling operations to decode the input global features into a finely detailed 3D dense mesh representation with two distinct hands. In the following sections, we will provide a detailed description of each module in the framework.

B. Dual-stream Encoder

In the feature extraction module, we need to fully extract the features from the RGB-D image containing both hands from the first perspective to restore accurate pose and shape.

RGB Feature Extraction. Firstly, given an unprocessed monocular RGB image $\mathcal{I}_c \in \mathbb{R}^{H\times W\times 3}$, we use the classic ResNet50 to extract 2D pyramid features as follows: $F = \{\mathcal{F}_1 \in \mathbb{R}^{H\times W\times 3}, \mathcal{F}_2 \in \mathbb{R}^{\frac{H}{2}\times\frac{W}{2}\times 64}, \mathcal{F}_3 \in \mathbb{R}^{\frac{H}{4}\times\frac{W}{4}\times 256}\}$. Then we adopt two simple decoder networks to regress the center $P_{ct} = \{P_l \in \mathbb{R}^2, P_r \in \mathbb{R}^2\}$ and mask $M = \{M_l \in \mathbb{R}^{H\times W}, M_r \in \mathbb{R}^{H\times W}\}$ of the left and right hands. The predicted center position of each hand will be used to initialize the 3D position of the hand mesh, and the predicted mask will be used to segment the hand area in the depth map.

Point Cloud Preprocessing. Given an unprocessed depth map $\mathcal{I}_d \in \mathbb{R}^{H\times W\times 1}$ and predicted mask M for both hands, we first convert the 2D image into a 3D point set using the camera's parameters. By calculating the mean depth of the point set, we filtered out outliers that exceed the threshold range [-0.08,+0.08] mm to reduce noise interference. Then, we randomly selected 1024 points from the remaining point set as the initial point cloud. Based on the generated initial point cloud, we can directly extract point cloud features using a specially designed network.

Review of PointNet++. Compared to directly extracting features from 2D depth maps using CNNs, PointNet [59] pioneered the extraction of high-dimensional features directly from point cloud through a per-point multi-layer perceptron (MLP) network. However, there is a lack of mining for local structural features due to the fixed number of points in PointNet. Therefore, PointNet++ [30] proposed a hierarchical feature extraction architecture to address this issue. Specifically, it includes multiple point set abstraction levels by selecting a fixed number of points in each layer as the center of the local area. The K neighbors around each center point will be aggregated and high-dimensional features will be extracted through the classic PointNet network. The center point and high-dimensional features will be fed into the next layer and the aggregation operation will be repeated. Finally, global features are extracted from all points in the last layer through the PointNet network. It is worthy of noting that previous work often used PointNet for point cloud classification, and it is still an unexplored field to predict dense 3D meshes from sparse point cloud.

Depth Feature Extraction. Given a set of point cloud data with both hands $X_h = \{X_l \in \mathbb{R}^{N\times C}, X_r \in \mathbb{R}^{N\times C}\}$, we refer to the structure of PointNet++ to extract pyramid point cloud features as follows: $P = \{\mathcal{P}_1 \in \mathbb{R}^{2\times N\times C}, \mathcal{P}_2 \in \mathbb{R}^{2\times N_1\times C_1}, \mathcal{P}_3 \in \mathbb{R}^{2\times N_2\times C_2}\}$. In our implementation, N=1024, C=3, N_1=512, C_1=131, N_2=128, C_2=259. At each level of point set abstraction, our approach involves employing ball queries to locate neighboring points within a predefined radius range. These identified points are subsequently fed into Multi-Layer Perceptrons (MLPs), enabling the extraction of high-dimensional features that correspond to the number of central points. These features are then concatenated with the features of the central point, yielding the point cloud features specific to that particular layer. Note that the pyramid point cloud features obtained at this stage exhibit a structure similar to the pyramid RGB features obtained earlier. In other words, as the scale decreases, the number of

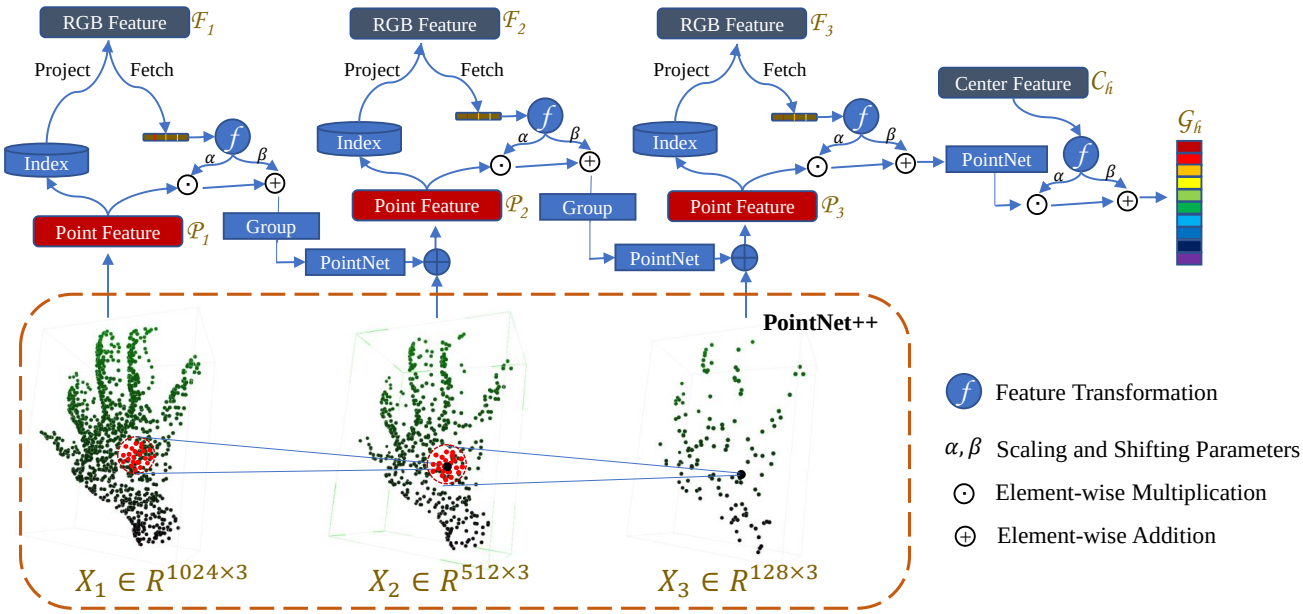

Fig. 3. Details of our proposed Pyramid Deep Fusion Network (PDFNet).

channels deepens, representing a continuous process of feature abstraction. By integrating pyramid features at different scales, we delve further into the feature characteristics of distinct modalities, thereby mutually reinforcing and complementing each other.

C. Pyramid Deep Fusion Network

At this stage, we have successfully obtained pyramid features for both modalities. Now, the crucial step is to fuse these features effectively. While the most simplistic approach involves a single layer of MLP to generate global features from the two modalities, followed by their concatenation, such a method neglects the local discrepancies present between the modalities. Factors such as motion blur, occlusion, and noise are significant local characteristics that might impair the ability of global features to complement each other. To address this issue, we have adopted a pixel-level feature fusion technique, aligning the corresponding RGB features with the point cloud features through 3D-2D projection of the point cloud. Notably, unlike the approach employed in DenseFusion [28], we perform pixel-by-pixel fusion on multi-scale pyramid features. In contrast to simply concatenating two distinct features, we incorporate a feature space transformation module. This module dynamically allocates weights to avoid the influence of local biases on the overall performance.

Specifically, we have designed a three-layer pyramid feature fusion structure, as shown in Fig. 3. With the help of PointNet++ [30] network, we downsample the initial point cloud $\mathcal{X}_1 \in \mathbb{R}^{1024\times3}$ to more sparse point cloud $\mathcal{X}_2 \in \mathbb{R}^{512\times3}$ and $\mathcal{X}_3 \in \mathbb{R}^{128\times3}$ through central point aggregation. Each set of point cloud finds K neighboring points as a local point set through the ball query of the center point. Through the PointNet network, higher-dimensional features are extracted from each local point set and subsequently consolidated into a single point representation via max-pooling. The resulting aggregated high-dimensional features are concatenated with the center point features from the original point cloud to obtain the point features P of that layer. For easier comprehension, the corresponding pseudocode can be seen in algorithm 1.

To acquire the RGB features corresponding to specific positions, we retain the index vector of the point cloud with respect to the depth map. Through this index, we project each point cloud onto a 2D feature map and gather the corresponding features, as illustrated in the projection-fetch process depicted in Fig. 3. The collected RGB features and point cloud features possess similar dimensions to facilitate seamless feature stitching, which is deemed as the conventional and effective approach for feature fusion. However, disparities exist in the data distribution and magnitude order between the two feature vectors, prompting the need for adaptive allocation of feature weights in order to attain enhanced outcomes. Motivated by the Spatial Feature Transform technique introduced by [71], we have tailored a shallow MLP network to learn scale and shift parameters individually for the aforementioned two features. The RGB features serve as the conditioning factor to acquire the scale and shift parameters. This enables a feature affine transformation that maps the point features into a novel feature space, as illustrated below

$$\hat{\mathcal{P}} = \mathcal{P} \odot \boldsymbol{\alpha} + \boldsymbol{\beta}, (\boldsymbol{\alpha}, \boldsymbol{\beta}) = \psi(\mathcal{F}). \tag{1}$$

$\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are learned affine transformation parameters scale and shift, whose dimension is the same as $\mathcal{P}$. $\odot$ refers to element-wise multiplication, while ψ refers to our feature transformation network. The transformed feature $\hat{\mathcal{P}}$ will be aggregated into a more sparse high-dimensional feature point cloud through the point set abstraction layer of PointNet++. The point cloud features of the last layer are fused to generate a single global feature $\mathcal{G} \in \mathbb{R}^{2\times1024\times1}$ through PointNet network. After obtaining the fused features, we aim to merge them with the center features derived from CNNs. The center features represent the global characteristics of the entire hand,

Algorithm 1: Algorithm for PDFNet Procedure

Input: pyramid RGB feature map F; center feature map C; initial point cloud X_h; camera intrinsic matrix K; num_layers; $BallRadius$; $NumPoints$.
Output: global fused feature $\hat{G}$.
P_1, $X_1 \leftarrow X_h$ ▷ Initialize point feature and point set
for i *in* $[1, num_layers]$ **do**
  $(u, v) \leftarrow K^{-1}X_i$ ▷ Find image coordinates
  $\hat{F}_i \leftarrow Fetch(F_i|u, v)$ ▷ Fetch corresponding RGB features
  $(\alpha, \beta) \leftarrow \psi_i(\hat{F}_i, P_i)$ ▷ Calculate affine transformation parameters
  $\hat{P}_i \leftarrow (P_i \odot (\alpha + 1) + \beta)$ ▷ Feature transformation
  if $i < num_layers$ **then**
    $S_i \leftarrow$ find$(\hat{P}_i, NumPoints_i, BallRadius_i)$ ▷ Find local structure
    $P_{i+1} \leftarrow cat(P_i, PointNet(group(S_i)))$ ▷ Point set abstraction
  else
    $G \leftarrow PointNet(\hat{P}_i)$
    $(\alpha, \beta) \leftarrow \psi_{i+1}(C, G)$
    $\hat{G} \leftarrow (G \odot (\alpha + 1) + \beta)$ ▷ Affine transformation
return $\hat{G}$

while the fused features consist of sparse local features. This design combines global and local elements, which maximizes the representation of input images and leads to improved results. Our subsequent ablation experiment further corroborates this discovery. Once we have acquired the final fused features, they are fed into our GCN-based decoder to output dense 3D meshes of both hands.

D. GCN-based Decoder

To fully leverage the extracted feature information, our decoder has been primarily constructed on the foundation of a state-of-the-art (SOTA) method [9]. Unlike primarily addressing interactive hands positioned at the center, our method accommodates hands appearing in any position within the field of view. Thus, we draw inspiration from the design of CenterNet [72] and utilize the center point as a representation of the hand. When extracting the corresponding image features, we collect global features from the central point position within the feature map, instead of directly flattening the entire map. Subsequent comparative experiments have substantiated the advantages of our approach, as it effectively focalizes the features on the hand regions rather than the background areas, ultimately yielding superior results.

We employ the Chebyshev Spectral Graph Conventional Network [73] to construct our 3D hand mesh, following the classic Coarse-to-fine structure. As in [9], we construct a three-layer submesh with designated vertex quantities, N_1=63, N_2=126, N_3=252. The final mesh is consistent with the topology of MANO [5], containing 778 vertices. Leveraging multiple upsampling layers, we successfully refine the hand mesh from the initial coarser submesh to the ultimate full MANO mesh.

Similar to PointNet, GCN learns the geometric structure of 3D meshes by directly optimizing the features on each vertex. Given the fused global feature $\mathcal{G}$, we map it into a more compact feature vector through a fully connected layer and concatenate it with the position encoding of vertices to obtain our initial graph features $\mathcal{G}_{\mathcal{V}} \in \mathbb{R}^{N\times C}, (N = 63, 126, 252), (C = 512, 256, 128)$. Similar to [9], our graph convolution operation on each graph feature is defined as follows:

$$G_{out} = \sum_{k=0}^{K-1} C_k(\hat{L}) G_{in} W_k. \tag{2}$$

where C_k is Chebyshev polynomials of degree k and $\hat{L} \in \mathbb{R}^{N\times N}$ is the scaled Laplacian matrix. $W_k \in \mathbb{R}^{C_{in}\times C_{out}}$ is a learnable weight matrix. $G_{in} \in \mathbb{R}^{N\times C_{in}}$ and $G_{out} \in \mathbb{R}^{N\times C_{out}}$ are input and output features in graph convolution operations, respectively. Through multiple regression heads composed of fully connected layers, we map the graph features of the last layer to the corresponding optimization objectives, such as root node coordinates, root-aligned MANO mesh, GCN mesh, etc.

E. Loss Functions

To facilitate end-to-end training of the entire model, we design a series of loss functions to constrain the learning process of parameters. In contrast to the original GCN approach [9], we augment our model with a localization module for both hands. This module incorporates an initialization scheme for the root node position, leveraging the hand center, and facilitates feature extraction at each central position. All our loss functions are provided in comprehensive detail below.

Center Loss is used to supervise our hand center learning. In essence, it is a pixel-wise binary logistic regression problem. The center points of the left and right hands are positive samples, while the rest are negative samples. Similar to CenterNet [72], we use the form of focal loss [74] to avoid the impact of imbalanced positive and negative sample sizes as follows:

$$\mathcal{L}_c = \sum_{h\in\{L,R\}} (1 - A_h)^{\gamma} \log(A_h), \tag{3}$$

where $A_h \in [0, 1]$ is the estimated confidence map for the positive class, and $1 - A_h$ is the probability for the negative class. γ is a hyperparameter and is set to 2 in our experiment.

Mask Loss is used to supervise the generation of hand masks, which is a typical semantic segmentation problem. We use smooth L_1 loss to calculate the difference between prediction and ground truth.

$$\mathcal{L}_m = ||M - \hat{M}||_1, \tag{4}$$

where $\hat{M}$ is the ground truth mask and M is our mask prediction.

Root Loss represents the L_1 distance between the predicted root node and the ground truth. In this work, we select the

TABLE I
COMPARISON WITH PREVIOUS SOTA METHODS ON H2O [16] EVALUATION DATASET. WE REPORT THE MPJPE/MPVPE AND AL-MPJPE/AL-MPVPE (MM) FOR EACH HAND.

Methods	Inputs		MPJPE↓			MPVPE↓			AL-MPJPE↓			AL-MPVPE↓		
	RGB	D	Left h.	Right h.	Mean h.	Left h.	Right h.	Mean h.	Left h.	Right h.	Mean h.	Left h.	Right h.	Mean h.
H2O [31]	✓		41.45	37.21	39.33	-	-	-	-	-	-	-	-	-
Hasson [75]	✓		39.56	41.87	40.72	-	-	-	-	-	-	-	-	-
Tekin [76]	✓		41.42	38.86	40.14	-	-	-	-	-	-	-	-	-
HOI4D [77]	✓		-	-	-	-	-	-	19.9	19.9	19.9	-	-	-
IntagHand [9]	✓		39.54	42.90	41.22	38.77	41.91	40.34	12.03	14.23	13.13	12.46	14.49	13.48
kypt-trans [16]	✓		-	-	-	-	-	-	21.36	19.57	20.47	-	-	-
Ours-RGB	✓		33.20	36.28	34.74	33.00	35.55	34.28	10.95	12.71	11.83	11.28	12.90	12.09
PointNet++ [30]		✓	17.17	17.83	17.50	16.98	17.37	17.18	7.61	9.40	8.51	7.82	9.42	8.62
IntagHand+D [9]	✓	✓	17.26	20.92	19.09	16.79	20.20	18.49	9.82	12.56	11.19	10.10	12.56	11.33
kypt-trans+D [16]	✓	✓	-	-	-	-	-	-	15.58	14.50	15.04	-	-	-
Densefusion [28]	✓	✓	21.39	25.76	23.58	21.42	25.34	23.38	18.08	23.55	20.81	18.42	23.79	21.11
Ours-full	✓	✓	**9.64**	**11.62**	**10.63**	**9.08**	**11.00**	**10.04**	**6.93**	**8.74**	**7.84**	**7.10**	**8.79**	**7.94**

first joint of the middle finger as our root node, which is the 9-th of the 21 joints.

$$\mathcal{L}_{root} = \sum_{h\in\{L,R\}} ||Root^h - \hat{Root}^h||_1. \tag{5}$$

Mesh Loss includes our GCN mesh loss and MANO mesh loss of 3D hand vertices, and we use L_1 loss for calculation.

$$\mathcal{L}_V = \sum_{h\in\{L,R\}} ||\mathcal{M}^h_{GCN} - \hat{\mathcal{M}}^h_{GCN}||_1 + ||\mathcal{M}^h_{MANO} - \hat{\mathcal{M}}^h_{MANO}||_1. \tag{6}$$

Joint Loss. We use the predefined joint regressor $\mathcal{J}$ in MANO to generate 3D joints. Similar to mesh loss, we use $L1_1$ loss.

$$\mathcal{L}_J = \sum_{h\in\{L,R\}} ||\mathcal{J}(\mathcal{M}^h_{MANO}) - \mathcal{J}(\hat{\mathcal{M}}^h_{MANO})||_1. \tag{7}$$

Re-projection Loss. We use projection functions to project 3D meshes and key points onto 2D images to calculate the re-projection loss, which is achieved through L_2 loss.

$$\begin{aligned}\mathcal{L}_{rep} = \sum_{h\in\{L,R\}} &||(\Pi(\mathcal{M}^h_{MANO}) - \Pi(\hat{\mathcal{M}}^h_{MANO}))||_2 \\ &+ ||(\Pi(\mathcal{J}(\mathcal{M}^h_{MANO})) - \Pi(\mathcal{J}(\hat{\mathcal{M}}^h_{MANO})))||_2.\end{aligned} \tag{8}$$

Smooth Loss. To ensure the smoothness of the output mesh, we add normal vectors and edge length loss.

$$\mathcal{L}_{smooth} = \sum_{i=1}^{3} ||e_i \cdot \hat{n}||_1 + ||e - \hat{e}||_1, \tag{9}$$

where $\hat{n}$ and e_i represent the ground truth normal vector and three edges on each face in the predicted mesh, respectively. e represents the length of each edge, while $\hat{e}$ represents the corresponding ground truth.

IV. EXPERIMENT

A. Implementation Details

Our proposed framework is implemented with PyTorch [78], incorporating an asymmetric dual-stream architecture for feature encoding. Unlike IntagHand [9] necessitating centered interactive hands in their training process, our model accommodates hands from arbitrary positions from the first perspective. For instance, with the H2O dataset [31] as our exemplar, we transform the input image into a square shape using zero padding and subsequently rescale it uniformly into 384×384. Although larger resolutions can preserve more details, they place higher demands on training memory. To conduct the training, we utilize two RTX2080Ti GPUs and assign a batch size of 8 instances per card. The initial learning rate is set to 1×10^{-4} and decreases by a factor of 10 at the 30th epoch. The entire training procedure spans 80 epochs and typically takes approximately three days to complete. Common data augmentation strategies, including scaling, rotation, translation, color jittering, and horizontal flipping, are used during training.

B. Datasets and Evaluation Metrics

H2O [31] is a realistic two-handed dataset that contains multi-view RGB-D images. We only used the first perspective data, including 55,742 images in the training set, 11,638 images in the validation set, and 23,391 images in the test set. The dataset provides high-resolution images, with RGB images and depth maps being 1280×720 and pixel aligned. In addition, it provides 62-dimensional MANO annotations for each hand, which can generate corresponding 3D meshes and keypoints. With the help of the camera intrinsic matrix, we obtained the corresponding 2D landmarks. The dataset captures different objects in different desktop backgrounds, resulting in complex and varied hand poses.

H2O-3D [16] is a real captured dataset that focuses on the interaction scenarios between two hands and objects. There are a total of 17 multi-view sequences, with 5 experimenters participating and manipulating 10 different objects for recording. This dataset collected 3D annotations for 76,340 images, including 60,998 images from 69 single-camera sequences used in the training set and 15,342 images from 16 single-camera sequences used in the testing set. The dataset provides RGB-D image pairs with a resolution of 640×480, captured from a third perspective.

RHD [4] is a large-scale synthetic dataset redered from freely available characters. It provides 3D key points annotation for both hands from a third perspective, containing 41,258 training and 2,728 testing data. The RGB-D image pairs are pixel-aligned with resolution of 320×320.

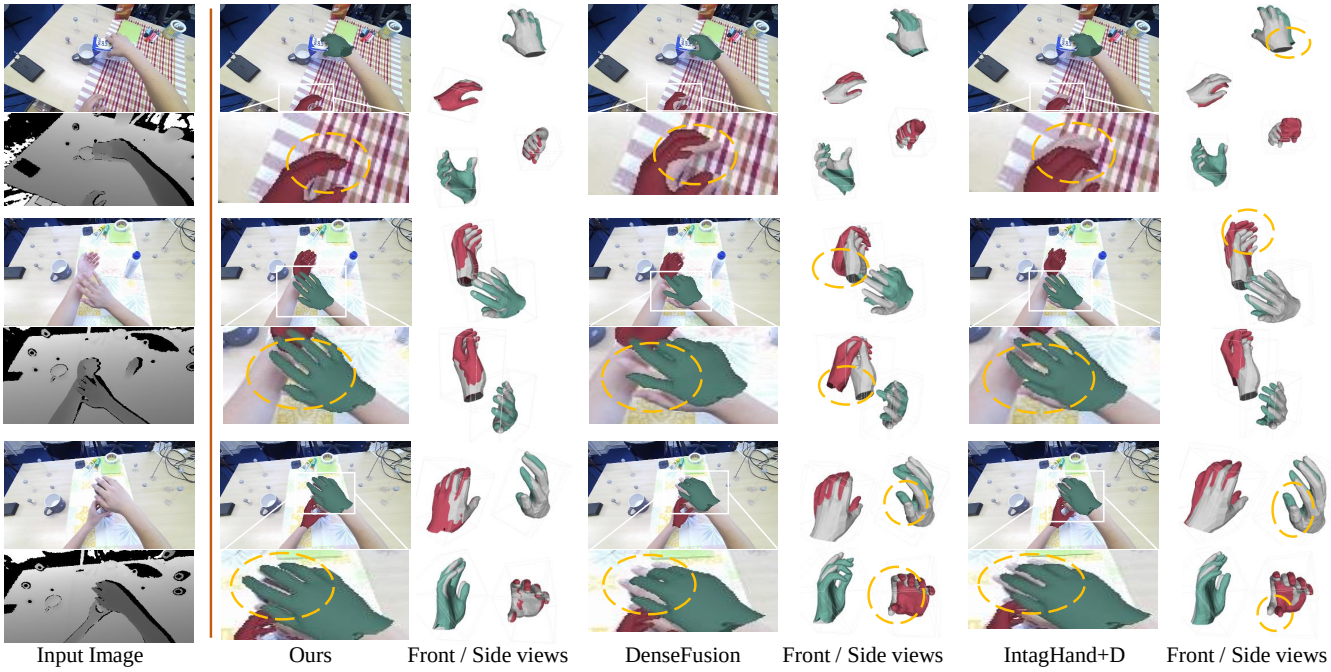

Fig. 4. Visual comparison on the H2O [16] dataset. We compared our results with DenseFusion [28] and IntagHand+D [9], and our results performed significantly better in hand-to-hand and hand-to-image alignment. We placed the predicted mesh and ground truth in the same coordinate system and color the left and right hands of the prediction in red and green respectively. From the side perspective, it can be seen that incorrect root node depth prediction can lead to significant misalignment.

Evaluation Metrics. To evaluate the accuracy of two-hand reconstruction, we used aligned mean per joint position error (AL-MPJPE) and aligned mean per vertex position error (AL-MPVPE) in millimeters to evaluate 3D key points and 3D mesh vertices after root node alignment, respectively. Practically, it becomes imperative to restore the accurate depth and scale of the reconstructed hands. Consequently, we additionally estimated the position of the root node and directly evaluated the MPJPE and MPVPE in the camera coordinate system.

TABLE II
COMPARISON WITH PREVIOUS SOTA METHODS ON H2O-3D [16] EVALUATION DATASET. WE REPORT THE AL-MPJPE (MM) AND AUC FOR EACH HAND.

Methods	Inputs RGB	Inputs D	AL-MPJPE↓	AUC↑
Densefusion [28]	✓	✓	12.7	0.747
IntagHand+D [9]	✓	✓	11.7	0.767
kypt-trans+D [16]	✓	✓	11.7	0.769
Ours-full	✓	✓	**10.7**	**0.787**

C. Comparisons with State-of-the-art Methods

Two-hand reconstruction results on H2O dataset. Firstly, we compared our method with previous SOTA two-hand reconstruction methods [9], [16], [31], [77] on H2O dataset. We also reported several single-hand pose estimation methods [75], [76], using two separate models for the left and right hand images of H2O and the results reported in the table are borrowed from H2O [31]. Due to being the first method to use RGB-D input for two-hand reconstruction, there is few existing reconstruction method to compare. Therefore, we added depth input to existing RGB-based methods to demonstrate the superiority of our fusion strategy. Since DenseFusion [28] was originally designed for the 6-DoF pose estimation of objects, we integrated it into our proposed framework for comparison. Similarly, the original PointNet++ [30] was designed for point cloud classification and segmentation, and we made corresponding modifications based on the authors' original implementation. Table I shows the evaluation results on H2O. Our method significantly surpasses the previous SOTA methods both in terms of absolute position error MPJPE/MPVPE and relative position error AL-MPJPE/AL-MPVPE. It obtains 9.64mm MPJPE of left hands and 11.62mm MPJPE of right hands under camera space. As for root-aligned position error, it obtains 6.93mm AL-MPJPE of left hands and 8.74mm AL-MPJPE of right hands. For a fair comparison, all methods in the table use the ground truth mask provided by the dataset to segment the depth map. In our subsequent ablation experiments, we also compared the results of directly using the mask estimated by the model.

The visual comparison results on the testing set of H2O can be seen in Fig. 4. We compared our method with DenseFusion [28] and IntagHand+D [9] on the H2O testing set. By projecting the predicted meshes onto the input image, we can visually compare the alignment results of hand-to-image. In addition, we placed the ground truth meshes and prediction results simultaneously in the camera coordinate system to compare the alignment results of hand-to-hand. We circled the parts with obvious differences in yellow in Fig. 4. It can be seen that our method achieved significantly better alignment in all perspectives.

Two-hand reconstruction results on H2O-3D dataset. H2O3D is a challenging two-hand dataset from a third per-

Fig. 5. Visualization results on the H2O-3D [16] test set. We compared our results with IntagHand+D [9] and DenseFusion [28], and our results performed significantly better in hand-to-image alignment.

TABLE III
ABLATION STUDY USING DIFFERENT EXPERIMENTAL CONFIGURATIONS EVALUATED ON H2O TEST SET. CHECKING FTN INDICATES THE USE OF THE DEDICATED FEATURE SPACE TRANSFORMATION NETWORK DESIGNED IN THIS ARTICLE WHILE UNCHECKING IT INDICATES THE USE OF SIMPLE FEATURE CONCATENATING. CHECKING GT MASK MEANS USING A GROUND TRUTH MASK TO SEGMENT THE DEPTH MAP WHILE UNCHECKING IT MEANS USING THE MASK PREDICTED BY THE MODEL.

	RGB	Depth	Point Feat.	RGB Feat.	Center Feat.	FTN	GT Mask	MPJPE↓	MPVPE↓	AL-MPJPE↓	AL-MPVPE↓
1	✓	-	-	-	✓	-	-	34.74	34.28	11.83	12.09
2	-	✓	✓	-	-	-	✓	17.50	17.18	8.50	8.62
3	✓	✓	-	-	✓	-	-	16.15	15.70	10.73	10.88
4	✓	✓	✓	✓	✓	-	✓	16.40	15.99	8.32	8.41
5	✓	✓	✓	✓	-	✓	✓	12.61	12.32	8.53	8.62
6	✓	✓	✓	✓	✓	✓	-	12.11	11.62	8.89	8.96
7	✓	✓	✓	✓	✓	✓	✓	**10.63**	**10.04**	**7.84**	**7.94**

TABLE IV
ABLATION STUDIES USING DIFFERENT DECODERS EVALUATED ON H2O [16] TEST SET.

Decoders	MPJPE↓	MPVPE↓	AL-MPJPE↓	AL-MPVPE↓
MANO	16.60	16.69	8.36	8.54
GCN	**10.63**	**10.04**	**7.84**	**7.94**

TABLE V
ABLATION STUDY USING DIFFERENT NUMBER OF PYRAMID LAYERS EVALUATED ON H2O DATASET.

Num. of Layers	MPJPE↓	MPVPE↓	AL-MPJPE↓	AL-MPVPE↓
L1	21.97	21.56	8.44	8.63
L2	19.58	19.11	8.95	9.12
L3	11.76	11.28	8.03	8.15
L1+L3	11.47	11.05	8.43	8.58
L1+L2+L3	**10.63**	**10.04**	**7.84**	**7.94**

spective. Our model has also been tested separately on it. As the dataset did not provide annotations for the testing set, we submitted and evaluated it on the official online platform[1]. The evaluation results are shown in Table II, and it can be seen that our method has achieved better performance under the specified evaluation indicators of the dataset. The visualization results are shown in Fig. 5.

[1]https://codalab.lisn.upsaclay.fr/competitions/4897

D. Ablation Study

We conducted a series of extensive ablation experiments on H2O and RHD datasets to confirm the contributions of different modules within our framework. Firstly, our objective is to demonstrate the performance improvement achieved by incorporating depth maps comparing to using solely RGB inputs. This aligns with the intuition of most individuals and our original intention behind the design of fusion modules. Secondly, we aim to demonstrate the advantage of our

proposed PDFNet algorithm over existing fusion strategies. To achieve this, we focus on modifying only the feature fusion module within the same encoder-decoder framework. Therefore, we show the efficacy of our pyramid design and feature transformation module. In the following, we present thorough ablation experiments and provide a detailed analysis of the corresponding outcomes.

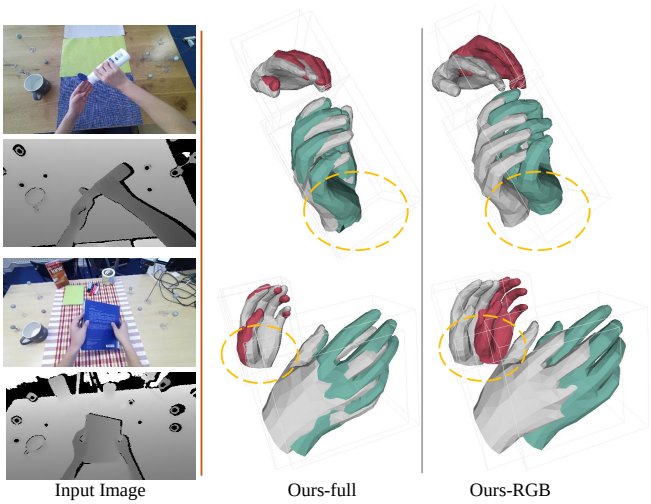

Fig. 6. Visual comparison of Ours-full model with Ours-RGB model in Table I on the H2O [16] dataset. We placed the predicted meshes and ground truth in the same camera space and color the predicted left and right hands with red and green respectively.

Comparison of different input modalities. We conducted a comprehensive analysis of our model's performance under different input scenarios, and the experimental results evaluated on H2O dataset are presented in Table III. Row-1 in the table denotes the baseline method, where only RGB images are employed for feature extraction. In Row-2, only depth maps are utilized to extract point features, with the aid of ground truth masks to generate the initial point cloud. Row-3 illustrates the model's direct usage of 4-channel RGB-D images as input. Comparing rows 1-3 in the table, it becomes evident that the integration of depth maps leads to a significant reduction in error by 50%. This aligns with our initial expectations, as predicting depth solely from RGB input inherently presents challenges due to the ill-posed nature of the problem. Visual comparisons are showcased in Fig. 6.

Row-3 attains lower absolute position error compared to Row-2, however, it does not possess an advantage in terms of relative position error. This indicates that RGB images contribute to improving the accuracy of the final predictions while also introducing some background interference information. Moreover, it signifies that a simplistic and coarse 4-channel input is not an ideal solution. This emphasizes our preference to fully leverage the complementary nature of the two input modalities.

Comparison of different fusion strategies. By comparing the results of DenseFusion [28] and Ours-full in Table I, it can be found that our devised pyramid feature fusion method confers significant performance advantages. The absolute position error (MPJPE) has decreased from 23.58mm to 10.63mm, while the relative position error (AL-MPJPE) has decreased from 20.81mm to 7.84mm.

Furthermore, we delved further into the impact of different modules in PDFNet on the final performance, as shown in Table III. Through a comparison between Row-4 and Row-7, we observe that the feature transformation network (FTN) plays a pivotal role in reducing model error, underscoring the necessity of adaptive weight allocation for the two input modalities. Otherwise, the introduction of undesired background interference features, as seen in Row-3, would adversely affect the final model performance.

In Row-5, we removed the center feature and directly utilized the fusion result of the point feature and RGB feature as the final feature. It is noteworthy that this approach leads to an increase of approximately 2mm in MPJPE and 1mm in AL-MPJPE compared to the full model. We posit that incorporating global center features proves advantageous, as it prevents the model from falling into local optima or overfitting by effectively leveraging the informative global-local features.

In Row-6, we attempted to employ the model's estimated mask for segmenting the depth map. However, this resulted in an increase of nearly 1mm in all error metrics. This phenomenon can be attributed to the inherent challenges in semantic segmentation itself, where there is inevitably a disparity between the predicted mask and the ground truth. Luckily, our model's performance only experienced a minor decrease while still surpassing the state-of-the-art (SOTA) methods significantly.

In addition, we also explored the impact of different pyramid layers on feature fusion and model performance on the H2O dataset. The experimental results are shown in Table V. It can be seen that by integrating pyramid features, stronger representation ability can be achieved, which also improves the final performance of the model.

Comparison of different decoders. To demonstrate the efficacy of our framework and the ease deployment of PDFNet, we replaced our GCN-based decoder with a MANO-based decoder. The experimental results are shown in Table IV, indicating that the GCN module used in our framework has achieved significant performance advantages. In addition, compared to the results of IntagHand and IntagHand+D in Table I, our MANO version still achieved improvements of 24.62mm MPJPE and 2.49mm MPJPE, respectively. This indicates that our PDFNet is an effective feature fusion algorithm that can extract more effective features for MANO-based decoders.

Besides, we use the RHD dataset to test whether our PDFNet module can work in complex scenarios, as this dataset is a synthetic two-hand dataset from a third perspective. It is very challenging because the position of the hand varies greatly, from a very close range of over 10mm to a distance of over 2 meters. Our results in Table VI confirm the robustness of our PDFNet. Besides, as this dataset only provides annotations for sparse 3D key points, we replaced the GCN-based decoder with a simple three-layer fully connected layer to directly output the coordinates of 21 key points. This experiment also proves that our PDFNet algorithm has a certain universality and benefits multiple decoders.

Limitations. The precision and generalizability of the model's

TABLE VI
PERFORMANCE COMPARISON BETWEEN USING THE PDFNET MODULE AND NOT USING IT ON THE RHD [4] TEST SET. WE REPORT THE MPJPE AND AL-MPJPE (MM) FOR EACH HAND HERE.

Methods	MPJPE↓		AL-MPJPE↓	
	Left h.	Right h.	Left h.	Right h.
w/o PDFNet	419.89	451.03	55.81	50.81
w/ PDFNet	**215.34**	**218.29**	**36.90**	**35.99**

mask predictions are not yet optimal. It is worthy of contemplating the utilization of expansive pre-trained models, such as SAM [79], to achieve greater adaptability across a wider application scenarios. In real-world implementations, the incorporation of temporal information from consecutive frames is imperative in acquiring consistent estimations. Regrettably, our current methodology solely supports single-frame RGB-D images as input, indicating room for further improvement.

V. CONCLUSION

This paper presents a comprehensive end-to-end framework for reconstructing both hands from a single RGB-D input. We adopt a well-designed dual-stream architecture to extract depth and RGB features, separately. Moreover, a novel pyramid feature fusion algorithm, named PDFNet, is introduced to synergistically leverage the strengths of these two complementary input modalities. The model successfully generates dense two-hand meshes in the camera coordinate system by employing our GCN-based decoder. Experiments have shown that the fusion algorithm and reconstruction framework proposed in this paper can accurately reconstruct two-hand meshes with real depth and scale. Compared to the state-of-the-art methods, our approach obtains a remarkable enhancement in performance. In future work, we aim to explore hand-object interaction and human-environment interaction to broaden the scope of application scenarios. Furthermore, both temporal and multi-perspective information can be considered to improve the usability of the model.

REFERENCES

[1] Y. Song, D. Demirdjian, and R. Davis, "Continuous body and hand gesture recognition for natural human-computer interaction," *ACM Transactions on Interactive Intelligent Systems*, vol. 2, no. 1, pp. 1–28, 2012.

[2] H. Lin and T.-W. Chen, "Augmented reality with human body interaction based on monocular 3d pose estimation," in *International Conference on Advanced Concepts for Intelligent Vision Systems (ACIVS)*, 2010, pp. 321–331.

[3] H. Fan, T. Zhuo, X. Yu, Y. Yang, and M. S. Kankanhalli, "Understanding atomic hand-object interaction with human intention," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, pp. 275–285, 2022.

[4] C. Zimmermann and T. Brox, "Learning to estimate 3d hand pose from single rgb images," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2017, pp. 4903–4911.

[5] J. Romero, D. Tzionas, and M. J. Black, "Embodied hands: modeling and capturing hands and bodies together," *ACM Transactions on Graphics*, vol. 36, no. 6, p. 245, 2017.

[6] L. Ge, Y. Cai, J. Weng, and J. Yuan, "Hand pointnet: 3d hand pose estimation using point sets," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 8417–8426.

[7] A. Boukhayma, R. de Bem, and P. H. Torr, "3d hand shape and pose from images in the wild," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 10 843–10 852.

[8] G. Moon and K. M. Lee, "I2l-meshnet: Image-to-lixel prediction network for accurate 3d human pose and mesh estimation from a single RGB image," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020, pp. 752–768.

[9] M. Li, L. An, H. Zhang, L. Wu, F. Chen, T. Yu, and Y. Liu, "Interacting attention graph for single image two-hand reconstruction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 2751–2760.

[10] Z. Yu, S. Huang, F. Chen, T. P. Breckon, and J. Wang, "Acr: Attention collaboration-based regressor for arbitrary two-hand reconstruction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023, pp. 12 955–12 964.

[11] T. Luan, Y. Zhai, J. Meng, Z. Li, Z. Chen, Y. Xu, and J. Yuan, "High fidelity 3d hand shape reconstruction via scalable graph frequency decomposition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023, pp. 16 795–16 804.

[12] K. Karunratanakul, S. Prokudin, O. Hilliges, and S. Tang, "Harp: Personalized hand reconstruction from a monocular rgb video," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 12 802–12 813.

[13] H. Xu, T. Wang, X. Tang, and C.-W. Fu, "H2onet: Hand-occlusion-and-orientation-aware network for real-time 3d hand mesh reconstruction," in *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023, pp. 17 048–17 058.

[14] B. Zhang, Y. Wang, X. Deng, Y. Zhang, P. Tan, C. Ma, and H. Wang, "Interacting two-hand 3d pose and shape reconstruction from single color image," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 11 334–11 343.

[15] D. Kim, K. I. Kim, and S. Baek, "End-to-end detection and pose estimation of two interacting hands," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 11 169–11 178.

[16] S. Hampali, S. D. Sarkar, M. Rad, and V. Lepetit, "Keypoint transformer: Solving joint identification in challenging hands and object interactions for accurate 3d pose estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 11 080–11 090.

[17] Y. Cai, L. Ge, J. Cai, and J. Yuan, "Weakly-supervised 3d hand pose estimation from monocular rgb images," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 678–694.

[18] Y. Cai, L. Ge, J. Cai, N. Magnenat-Thalmann, and J. Yuan, "3d hand pose estimation using synthetic data and weakly labeled rgb images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, pp. 3739–3753, 2021.

[19] S. Li and D. Lee, "Point-to-pose voting based hand pose estimation using residual permutation equivariant layer," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 11 919–11 928.

[20] D. Xu, D. Anguelov, and A. Jain, "Pointfusion: Deep sensor fusion for 3d bounding box estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 244–253.

[21] L.-Z. Chen, Z. Lin, Z. Wang, Y.-L. Yang, and M.-M. Cheng, "Spatial information guided convolution for real-time rgbd semantic segmentation," *IEEE Transactions on Image Processing*, vol. 30, pp. 2313–2324, 2020.

[22] Z. Wu, S. Gobichettipalayam, B. Tamadazte, G. Allibert, D. P. Paudel, and C. Demonceaux, "Robust rgb-d fusion for saliency detection," in *International Conference on 3D Vision (3DV)*, 2022, pp. 403–413.

[23] H. Cheng, L. Yang, and Z. Liu, "Survey on 3d hand gesture recognition," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 26, pp. 1659–1673, 2016.

[24] E. Kazakos, C. Nikou, and I. A. Kakadiaris, "On the fusion of rgb and depth information for hand pose estimation," in *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, 2018, pp. 868–872.

[25] F. Mueller, D. Mehta, O. Sotnychenko, S. Sridhar, D. Casas, and C. Theobalt, "Real-time hand tracking under occlusion from an egocentric rgb-d sensor," in *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, 2017, pp. 1284–1293.

[26] X. Lin, Y. Zhou, K. Du, Y. Sun, X. Ma, and J. Lu, "Multi-level fusion net for hand pose estimation in hand-object interaction," *Signal Processing: Image Communication*, vol. 94, p. 116196, 2021.

[27] X. Sun, B. Wang, L. Huang, Q. Zhang, S. Zhu, and Y. Ma, "Crossfunet: Rgb and depth cross-fusion network for hand pose estimation," *Sensors*, vol. 21, no. 18, p. 6095, 2021.

[28] C. Wang, D. Xu, Y. Zhu, R. Martín-Martín, C. Lu, L. Fei-Fei, and S. Savarese, "Densefusion: 6d object pose estimation by iterative dense fusion," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 3338–3347.

[29] Y. Wang, X. Jiang, H. Fujita, Z. Fang, X. Qiu, and J. Chen, "Efn6d: an efficient rgb-d fusion network for 6d pose estimation," *Journal of Ambient Intelligence and Humanized Computing*, pp. 1–14, 2022.

[30] C. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," in *Advances in neural information processing systems (NeurIPS)*, 2017, pp. 5099–5108.

[31] T. Kwon, B. Tekin, J. Stühmer, F. Bogo, and M. Pollefeys, "H2o: Two hands manipulating objects for first person interaction recognition," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 10 118–10 128.

[32] Y. Wang, C. Peng, and Y. Liu, "Mask-pose cascaded cnn for 2d hand pose estimation from single color image," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, pp. 3258–3268, 2019.

[33] S. Guo, E. Rigall, L. Qi, X. Dong, H. Li, and J. Dong, "Graph-based cnns with self-supervised module for 3d hand pose estimation from monocular rgb," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, pp. 1514–1525, 2021.

[34] L. Ge, Z. Ren, and J. Yuan, "Point-to-point regression pointnet for 3d hand pose estimation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 475–491.

[35] M. Li, J. Wang, and N. Sang, "Latent distribution-based 3d hand pose estimation from monocular rgb images," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, pp. 4883–4894, 2021.

[36] G. Moon, T. Shiratori, and K. M. Lee, "Deephandmesh: A weakly-supervised deep encoder-decoder framework for high-fidelity hand mesh modeling," in *European Conference on Computer Vision (ECCV)*, 2020, pp. 440–455.

[37] J. Lee, M. Sung, H. T. Choi, and T.-K. Kim, "Im2hands: Learning attentive implicit representation of interacting two-hand shapes," in *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023, pp. 21 169–21 178.

[38] S. Guo, E. Rigall, Y. Ju, and J. Dong, "3d hand pose estimation from monocular rgb with feature interaction module," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, pp. 5293–5306, 2022.

[39] J. Ren, J. Zhu, and J. Zhang, "End-to-end weakly-supervised single-stage multiple 3d hand mesh reconstruction from a single rgb image," *Computer Vision and Image Understanding*, vol. 232, p. 103706, 2023.

[40] K. Lin, L. Wang, and Z. Liu, "Mesh graphormer," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 12 919–12 928.

[41] D. Kulon, R. A. Güler, I. Kokkinos, M. M. Bronstein, and S. Zafeiriou, "Weakly-supervised mesh-convolutional hand reconstruction in the wild," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 4989–4999.

[42] A. Spurr, A. Dahiya, X. Zhang, X. Wang, and O. Hilliges, "Self-supervised 3d hand pose estimation from monocular rgb via contrastive learning," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 11 210–11 219.

[43] Z. Tu, Z. Huang, Y. Chen, D. Kang, L. Bao, B. Yang, and J. Yuan, "Consistent 3d hand reconstruction in video via self-supervised learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, pp. 9469–9485, 2022.

[44] X. Zhang, Q. Li, H. Mo, W. Zhang, and W. Zheng, "End-to-end hand mesh recovery from a monocular rgb image," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 2354–2364.

[45] S. Baek, K. I. Kim, and T.-K. Kim, "Pushing the envelope for rgb-based dense 3d hand pose estimation via neural rendering," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 1067–1076.

[46] X. Chen, Y. Liu, C. Ma, J. Chang, H. Wang, T. Chen, X. Guo, P. Wan, and W. Zheng, "Camera-space hand mesh recovery via semantic aggregation and adaptive 2d-1d registration," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 13 269–13 278.

[47] C. Zimmermann, D. Ceylan, J. Yang, B. Russell, M. J. Argus, and T. Brox, "Freihand: A dataset for markerless capture of hand pose and shape from single rgb images," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 813–822.

[48] S. Hampali, M. Rad, M. Oberweger, and V. Lepetit, "Honnotate: A method for 3d annotation of hand and object poses," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 3193–3203.

[49] P. Panteleris, I. Oikonomidis, and A. A. Argyros, "Using a single rgb frame for real time 3d hand pose estimation in the wild," in *Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2018, pp. 436–445.

[50] Y. Rong, T. Shiratori, and H. Joo, "Frankmocap: A monocular 3d whole-body pose estimation system via regression and integration," in *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, 2021, pp. 1749–1759.

[51] G. Moon, S.-I. Yu, H. Wen, T. Shiratori, and K. M. Lee, "Interhand2.6m: A dataset and baseline for 3d interacting hand pose estimation from a single rgb image," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020, pp. 548–564.

[52] C. Wang, F. Zhu, and S. Wen, "Memahand: Exploiting mesh-mano interaction for single image two-hand reconstruction," in *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023, pp. 564–573.

[53] M. Oberweger, P. Wohlhart, and V. Lepetit, "Training a feedback loop for hand pose estimation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2015, pp. 3316–3324.

[54] C. Wan, T. Probst, L. V. Gool, and A. Yao, "Dense 3d regression for hand pose estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 5147–5156.

[55] X. Zhou, Q. Wan, W. Zhang, X. Xue, and Y. Wei, "Model-based deep hand pose estimation," in *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI)*, 2016, pp. 2421–2427.

[56] W. Huang, P. Ren, J. Wang, Q. Qi, and H. Sun, "Awr: Adaptive weighting regression for 3d hand pose estimation," in *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2020, pp. 11 061–11 068.

[57] L. Xu, C. Hu, J. Tao, J. Xue, and K. Mei, "Improve regression network on depth hand pose estimation with auxiliary variable," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, pp. 890–904, 2021.

[58] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 5679–5688.

[59] C. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 77–85.

[60] 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. 43 425–43 439, 2018.

[61] 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 6960–6969.

[62] L. Huang, J. Tan, J. Liu, and J. Yuan, "Hand-transformer: Non-autoregressive structured modeling for 3d hand pose estimation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020, pp. 17–33.

[63] J. Taylor, V. Tankovich, D. Tang, C. Keskin, D. Kim, P. L. Davidson, A. Kowdle, and S. Izadi, "Articulated distance fields for ultra-fast tracking of hands interacting," *ACM Transactions on Graphics*, vol. 36, pp. 1–12, 2017.

[64] F. Mueller, M. Davis, F. Bernard, O. Sotnychenko, M. Verschoor, M. A. Otaduy, D. Casas, and C. Theobalt, "Real-time pose and shape reconstruction of two interacting hands with a single depth camera," *ACM Transactions on Graphics*, vol. 38, no. 4, pp. 1–13, 2019.

[65] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[66] I. Oikonomidis, N. Kyriazis, and A. A. Argyros, "Tracking the articulated motion of two strongly interacting hands," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012, pp. 1862–1869.

[67] 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 *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016, pp. 294–310.
[68] D. Tzionas, L. Ballan, A. Srikantha, P. Aponte, M. Pollefeys, and J. Gall, "Capturing hands in action using discriminative salient points and physics simulation," *International Journal of Computer Vision*, vol. 118, pp. 172–193, 2015.
[69] S. Yuan, B. Stenger, and T.-K. Kim, "3d hand pose estimation from rgb using privileged learning with depth data," in *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, 2019, pp. 2866–2873.
[70] J. Hu, L. Shen, S. Albanie, G. Sun, and E. Wu, "Squeeze-and-excitation networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, pp. 2011–2023, 2017.
[71] X. Wang, K. Yu, C. Dong, and C. C. Loy, "Recovering realistic texture in image super-resolution by deep spatial feature transform," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 606–615.
[72] X. Zhou, D. Wang, and P. Krähenbühl, "Objects as points," in *arXiv preprint arXiv:1904.07850*, 2019.
[73] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *Advances in neural information processing systems (NeurIPS)*, 2016, pp. 3837–3845.
[74] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal loss for dense object detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 2, pp. 318–327, 2020.
[75] Y. Hasson, B. Tekin, F. Bogo, I. Laptev, M. Pollefeys, and C. Schmid, "Leveraging photometric consistency over time for sparsely supervised hand-object reconstruction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 571–580.
[76] B. Tekin, F. Bogo, and M. Pollefeys, "H+o: Unified egocentric recognition of 3d hand-object poses and interactions," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 4506–4515.
[77] Y. Liu, Y. Liu, C. Jiang, Z. Fu, K. Lyu, W. Wan, H. Shen, B.-H. Liang, H. Wang, and L. Yi, "Hoi4d: A 4d egocentric dataset for category-level human-object interaction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 20 981–20 990.
[78] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in neural information processing systems (NeurIPS)*, 2019, pp. 8024–8035.
[79] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollár, and R. Girshick, "Segment anything," *arXiv:2304.02643*, 2023.

Jinwei Ren is currently a PhD candidate in the College of Computer Science and Technology, Zhejiang University, Hangzhou, China. Before that, he received the bachelor degree from Chongqing University, China, in 2017. His research interests include SLAM and computer vision, with a focus on 3D reconstruction.

Jianke Zhu received the master's degree from University of Macau in Electrical and Electronics Engineering, and the PhD degree in computer science and engineering from The Chinese University of Hong Kong, Hong Kong in 2008. He held a post-doctoral position at the BIWI Computer Vision Laboratory, ETH Zurich, Switzerland. He is currently a Professor with the College of Computer Science, Zhejiang University, Hangzhou, China. His research interests include computer vision and multimedia information retrieval.