Learning to estimate 3D hand pose from depth image

Lunke Fei, Xin Wang, Qi Zhu, Shuping Zhao, Jie Wen, Jian Zhu, Xiaomin Hu

***Abstract*—Accurate 3D hand pose estimation from single depth images is crucial for applications such as human-computer interaction, yet remains challenging due to the inherent depth-to-3D ambiguity and self-occlusion. In this paper, we propose an attention-based graph convolutional network (AttGCN) for 3D hand pose estimation that learns fine grained hand-joint features in a coarse-to-fine manner. The core innovation of our approach is the synergistic integration of attention driven graph convolutions within a hierarchical architecture. Specifically, we first extract multiscale shallow features from hand depth images via a ResNet backbone, and then enhance joint-specific representation via cross-channel interaction guided by a 2D heatmap loss. After that, we employ a multi layer vanilla GCN to capture latent intra finger and inter finger dependencies, producing coarse 3D joint positions under a 3D information loss. Finally, we refine the 3D hand-joint position representation using a semantic graph convolution with attention embeddings, which incorporates explicit hand skeleton priors to yield precise and fine grained 3D coordinates. This progressive design effectively mitigates self-occlusion and models cross-joint dependencies. Extensive experimental results on three widely used datasets (MSRA, NYU, and ICVL) clearly demonstrate the effectiveness of the proposed method.**

***Index Terms*—Hand pose estimation, hand depth images, attention mechanism, graph convolutional network.**

I. Introduction

3D hand pose estimation refers to locating the 3D positions of hand joints from 2D hand images or video frames, which has found widespread potential applications in real-world scenarios, such as human-computer interaction [1], virtual reality, and augmented reality [2]. To date, extensive 3D hand pose estimation methods have been proposed in the literature, which mainly consist of RGB image-based [3] [4] and depth image-based [5] [6] [7] methods. While RGB hand images can be easily captured with fewer imaging requirements, they often suffer from depth ambiguity due to the lack of depth information, making it difficult to determine the hand-joint depth positions in 3D space. With the emergence of RGB-D cameras, hand depth images can also be acquired inexpensively. More importantly, 2D depth images have a 2.5D nature because they convey spatial distance information of the scene, thus providing more spatial relationships of hand joints. For this reason, recent efforts have been increasingly devoted to estimating 3D hand poses from depth images [5] [6] [7], which is also the focus of this paper.

Existing studies of 3D hand pose estimation from depth images can be roughly classified into regression-based, detection-based, and hybrid categories. More specifically, regression-based methods [8] [9] [6] typically directly regress the 3D coordinates or joint angles from hand images through fully connected networks for hand pose estimation tasks. For example, Ren et al. [9] proposed a stacked regression network with a differentiable reparameterization module to capture spatial structures for real-time 3D hand pose estimation. Lu et al. [6] adopted a segmentation-guided module to infer full hand segmentation and utilized progressive multi-path regression to integrate hierarchical features, thereby alleviating self-occlusion and hand-object interaction occlusions for fine-grained hand pose and shape estimation.

Rather than directly regressing joint coordinates, detection-based methods [10] [11] [12] [7] first obtain dense probability maps (e.g., heatmaps of the joints) to locate keypoints and then associate them with the corresponding joints. For instance, Rezaei et al. [10] decomposed the 3D hand pose estimation task into estimating 2D joint locations in depth image space (UV) using two complementary attention maps at both prediction and feature levels, and they further proposed an appearance-based data augmentation method named PixDropout to improve 3D hand pose estimation accuracy. In addition, Zhou et al. [7] generated hand depth images with precise annotations by mapping a Gaussian distributed variable into depth-specific noise patterns and transforming noise-free synthetic depth images into realistic-looking depth images for 3D hand pose estimation with only a small fraction of annotated real images.

To leverage the advantages of both the detection and regression methods, hybrid methods [5] [13] [14] aim to fuse hand-joint regression and keypoint detection schemes for 3D hand pose estimation. For example, Huang et al. [13] proposed an adaptive weighting regression (AWR) method for 3D hand pose estimation by aggregating different parts of the dense representation through discrete integration under the guidance of joint supervision. Fang et al. [5] introduced a joint graph reasoning-based pixel-to-offset prediction network that unifies dense pixel-wise offset prediction to model complex

This work was supported in part by the National Key R&D Program of China under Grant 2023YFF0905603, in part by the National Natural Science Foundation of China under Grant 62576112, and in part by the Guangdong Basic and Applied Basic Research Foundation under Grant 2026A1515012129. *(Corresponding author: Jian Zhu)*

Lunke Fei, Xin Wang, Shuping Zhao, Jian Zhu and Xiaomin Hu are with the School of Computer Science and Technology, Guangdong University of Technology, Guangzhou, China. (e-mail: flksxm@126.com; WX749265809@163.com; yb77458@um.edu.mo; dr.zhuj@gmail.com; xmhu@gdut.edu.cn)

Qi Zhu is with the College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, China. (e-mail: zhuqi@nuaa.edu.cn)

Jie Wen is with the School of Computer Science and Technology, Harbin Institute of Technology (Shenzhen), Shenzhen, China. (e-mail: jiewen_pr@126.com)

dependencies among joints with direct joint regression for end-to-end training. Despite these advances, accurately estimating occluded hand-joint positions from depth images remains an open problem due to the heavy self-occlusion and high self-flexibility of the hand.

In this paper, we propose a coarse-to-fine hand-joint position feature learning network, named the attention-based graph convolutional network (AttGCN), for 3D hand pose estimation. First, we extensively extract multiscale shallow features from the input hand depth images via a ResNet50 backbone. Then, we perform cross-channel feature interaction within multiscale shallow feature maps, and apply a 2D joint-specific heatmap loss to enhance the 2D joint-specific hand features. After that, we introduce multiple cascaded vanilla graph convolutional networks to capture the latent long-range dependencies among the hand joints. Finally, we refine the 3D positions of the interacting hand joints via a semantic graph convolution with embedding self-attention and cross-attention. We conduct extensive experiments on the challenging NYU, MSRA, and ICVL benchmarks to demonstrate the effectiveness of the proposed AttGCN method.

The main contributions of this paper can be summarized as follows:

1) We propose a coarse-to-fine hand-joint feature learning method, which first extensively exploits the coarse joint-highlighted hand features, and then specifically refines the joint-specific features. With the proposed method, the fine-grained 3D hand-joint positions can be effectively characterized from hand depth images for 3D hand pose estimation.
2) We develop three-layer vanilla graph convolutional networks to capture the underlying interacting hand-joint relationship, and further employ eight-layer semantic graph convolutions that embed self- and cross-attention to adaptively adjust the 3D positions of hand-joints with close-distance associations such that more 3D hand structure information can be obtained, and hand-joint positions with self-occlusion can be well discriminated during 3D hand pose estimation.
3) We conduct extensive comparative experiments on three challenging benchmarks, and the experimental results show that the proposed method is superior to most state-of-the-art methods for 3D hand pose estimation.

The remainder of the paper is organized as follows. Section II briefly reviews three related topics. Section III elaborates our proposed AttGCN for 3D hand pose estimation. Section IV analyzes the experimental results. Lastly, Section V offers the conclusion of this paper.

II. Related Work

In this section, we briefly review three related topics, including hand depth images, graph neural network, and attention mechanisms in 3D hand pose estimation.

A. Hand Depth Images

In general, RGB hand images are usually acquired using standard CCD cameras, which describe the hand's color information with three-channel RGB values at each pixel. In contrast, hand depth images need to be captured via a depth camera [15], which preserves the spatial distance information of the hand scene with a z-axis value (i.e., depth) at each pixel. Therefore, a hand depth image can be considered a type of 2D image, in which each pixel encodes a spatial distance value. In addition, RGB-D cameras can simultaneously capture both RGB and depth information of the hand. Consequently, RGB-D hand images convey both color measurements and depth information for 3D pose estimation [16], but they usually require expensive imaging devices. Therefore, hand depth images are more widely used for 3D hand pose estimation [17] [18] [19] [8] [5] due to their dense depth information and relatively lower image acquisition cost. In this study, we propose a coarse-to-fine feature learning network to estimate 3D hand-joint coordinates from hand depth images for 3D hand pose estimation.

B. Graph Convolutional Network in 3D Hand Pose Estimation

Graph convolutional networks (GCNs) have been widely employed to model local and global relationships in graph-structured data, including applications such as traffic flow forecasting [20] and human pose estimation [21]. Given that hand joints, especially the intra-finger joints, naturally exhibit graph-like latent correlations, many existing studies have adopted GCNs to model these underlying relationships among hand joints. For example, Doosti et al. [22] proposed a graph-based U-Net with two adaptive GCNs to infer 3D joint locations from 2D key points. Ren et al. [11] introduced a cascaded hierarchical GCN-based network to model correlations across different hand regions for 3D hand pose estimation. Zhao et al. [23] designed a semantic graph convolution network (SemGConv) to exploit local and global node relationships for 3D human pose regression. In addition, Xu et al. [24] developed a GCN with a repeated encoder-decoder to incorporate multilevel features extracted from graph hourglass networks for 2D-to-3D human pose estimation. Unlike these approaches, in this study, we employ cascaded GCNs to model both short range and long range dependencies among hand joints, specifically targeting fine grained 3D joint position estimation. By explicitly distinguishing the complementary roles of shallow and deep graph convolutions, we achieve more accurate reconstruction of hand joints under challenging conditions such as self occlusion.

C. Attention Mechanisms of 3D Hand Pose Estimation

Imitating the human visual system, attention mechanisms embedded in the neural networks attempt to focus on important feature learning through a dynamic weighting process, which has benefited various visual tasks, such as image recognition, semantic segmentation, and object detection [25]. The hand pose in a hand image heavily depends on the joint regions, and attention mechanism can specifically model the pose features associated with interacting hand joints under self-occlusion. Therefore, attention mechanisms have been widely adopted for 3D hand pose estimation. For example, Fu et al. [26] proposed a dynamic fusion Transformer with cross-attention and temporal attention to predict hand poses in frames with

This article has been accepted for publication in IEEE Transactions on Multimedia. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMM.2026.3724754

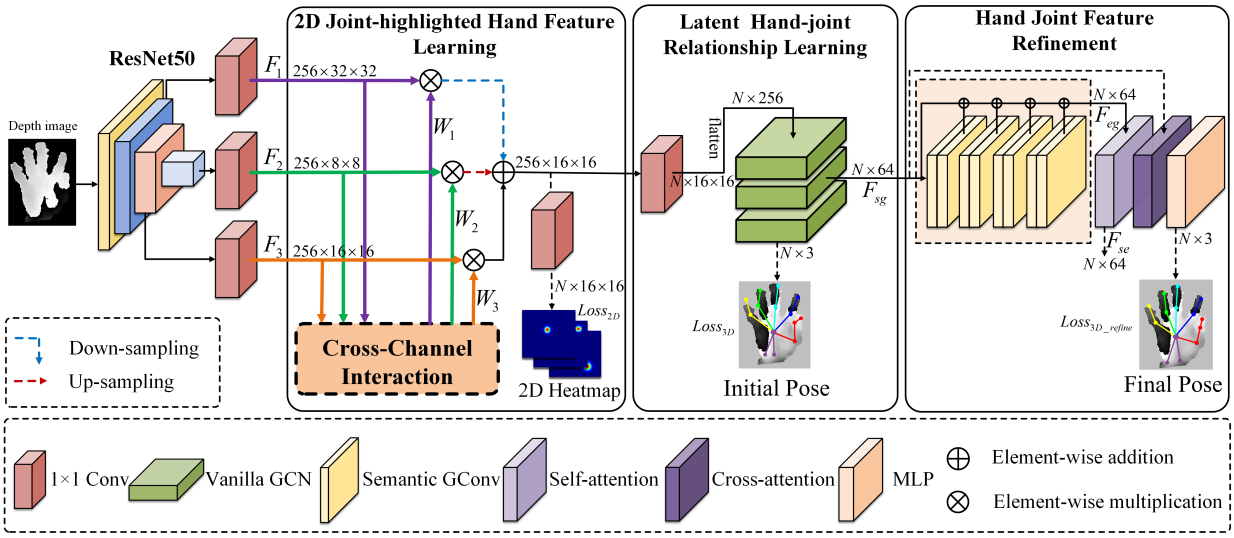

Fig. 1. The overall architecture of the proposed AttGCN for 3D hand pose estimation from hand depth images, which consists of a ResNet50-based shallow feature extraction backbone, and three coarse-to-fine and 2D to 3D hand-joint feature learning subnetworks.

heavy occlusion or blurring. Huang et al. [8] introduced a Transformer-based structured modeling framework with a multi-head attention mechanism to discover informative points for hand-joint localization. Wang et al. [27] presented a self-attention-based topology-aware network that specifically captures long-range dependencies and local topology connections between joints for 3D hand pose estimation. Zhao et al. [28] designed GraAttention by embedding a graph convolution layer after a multi-head self-attention block to model joint relationships for 3D pose estimation. In this study, we embed multiple attention mechanisms into a GCN to exploit short-range dependencies among joints to recover fine-grained 3D hand-joint positions.

III. AttGCN

In this section, we first present the overall architecture of our proposed AttGCN, and then elaborate the main components of the AttGCN.

A. AttGCN Architecture

Our proposed 3D hand pose estimation network, i.e., AttGCN, consists of a multiscale shallow feature extractor and three coarse-to-fine hand-joint feature learning subnetworks, as shown in Fig. 1. First, to extensively exploit the semantic information from hand depth images, we employ the ResNet50 [29] as the backbone to extract the multiscale shallow features of the hand depth images, which usually contain both hand local semantic and global structural information [29]. To be more specific, for a hand depth image with dimensions of $h \times w$, we extract the outputs of the last three layers of the ResNet50 with the down-sampling rates of ×8, ×16, and ×32 as the multiscale shallow features, and further map them into the same channel number (i.e., 256) via 1×1 convolutions. As a result, our backbone generates three-scale pyramid shallow feature maps with dimensions of $256 \times \frac{h}{32} \times \frac{w}{32}$, $256 \times \frac{h}{8} \times \frac{w}{8}$, and $256 \times \frac{h}{16} \times \frac{w}{16}$, referred to as F_1, F_2, F_3, respectively. Then, in the 2D joint-highlighted hand feature learning subnetwork, we exploit the multiscale features via cross-channel interaction by performing interactions between features at different scales, and generate three location weights (e.g., W_1, W_2, W_3) to enhance the pyramid hand shallow feature maps (i.e., F_1, F_2, F_3) via element-wise multiplication. Moreover, in the latent hand-joint relationship learning subnetwork, we employ multiple cascaded vanilla GCNs to capture the latent long-range dependencies among the hand joints. Finally, in the hand-joint feature refinement subnetwork, we refine the 3D hand-joint representation via a semantic graph convolution with embedding self- and cross-attention mechanisms for accurate 3D hand pose estimation. In the following, we elaborate the 2D joint-highlighted hand feature learning, latent hand-joint relationship learning, and hand-joint feature refinement subnetworks, respectively.

B. 2D Joint-highlighted Hand Feature Learning

The hand pose heavily depends on the spatial locations of hand joints. To capture the hand-joint-specific features, we perform cross-channel interaction within the intra-scale and the inter-scale hand shallow feature maps. Inspired by the fact that Feature Pyramid Network (FPN) [30] can effectively merge high-level semantic information with low-level detailed features through lateral connections, our proposed approach develops a pyramid structure to capture the fine-grained details of hand joints. Unlike the existing FPN that usually emphasizes most on the fusion of features at different scales, our proposed method primarily focuses on cross-channel interaction within the pyramid layers to enhance feature representations.

Fig. 2 shows the basic idea of the cross-channel interaction module. First, we convert the three-scale pyramid hand shallow feature maps (i.e., F_1, F_2, F_3) into three $256 \times 1 \times 1$ -dimensional channel feature maps via global average pooling (GAP) [31], and concatenate them to form an integrated $712 \times 1 \times 1$-sized channel-wise feature vector. Then, we

This article has been accepted for publication in IEEE Transactions on Multimedia. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMM.2026.3724754

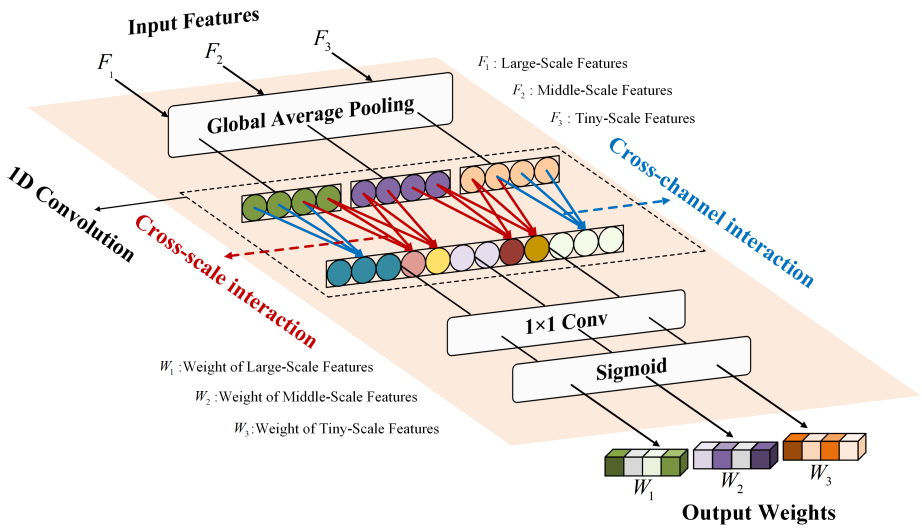

Fig. 2. The basic pipeline of the cross-channel hand feature interaction. We first convert the three-scale pyramid hand shallow feature maps into an integrated channel-wise feature vector via GAP. Then, we perform cross-channel interaction via 1D convolution to exploit the hand-joint-specific features.

perform a cross-channel feature interaction by convolving the integrated channel-wise feature vector with a fast 1D convolution template [32] with a kernel size of 3 (i.e., the coverage of cross-channel interaction). By doing so, the inter-scale and intra-scale hand feature associations at different channels can be established, and different scales of hand-joint features can also be associated. With the channel-interaction feature maps, we employ a 1×1 convolution layer followed by a sigmoid function to generate three $256 \times 1 \times 1$ location weights (e.g., W_1, W_2, W_3) to enhance the pyramid hand shallow feature maps (i.e., F_1, F_2, F_3) via element-wise multiplication. By doing so, multiscale hand-joint-specific feature maps at different locations can be adaptively modulated, enhancing the important features while suppressing less important ones.

Then, we employ up-sampling and down-sampling [29] to resize different scales of the highlighted feature maps to the same resolution (i.e., $256 \times 16 \times 16$) and aggregate them via element-wise addition to form a global joint-highlighted hand feature representation. After that, we apply Batch Normalization to stabilize the training process and meanwhile overcome gradient explosion. Following the Batch Normalization, we further employ the nonlinear ReLU activation function to capture the complex hand-joint-specific features.

Finally, to better model the 2D spatial information of hand-joints, we convert enhanced hand feature representation into the 2D confidence heatmap for each hand joint on the basis of the 2D Gaussian distribution of a joint located in a specific position [14]. After that, we calculate the joint-specific 2D heatmap loss (referred to as $Loss_{2D}$), as follows:

$$Loss_{2D} = \sum_{i=1}^{N} \|H_i - H_i^*\|^2, \tag{1}$$

where N denotes the total number of joints in a hand, and H_i^* and H_i represent the 2D ground-truth and predicted heatmaps of the i-th joint of the hand, respectively. By minimizing the hand-joint heatmap loss during training, we aim to highlight the hand-joint-specific features while preserving hand-joint 2D spatial information and suppressing the pose-irrelevant hand features.

C. Latent Hand-joint Relationship Learning

Notably, both the adjacent and nonadjacent hand joints of a hand are essentially associated. For example, the intra-finger joints are usually located along a curve, and the inter-finger joints of neighboring fingers are usually located within a close distance and in nearly parallel curves. The latent relationships among the associated hand joints usually play a decisive role in hand pose estimation. For this reason, having obtained the joint-highlighted hand feature representation, we further capture the latent interacting relationships among both intra- and inter-finger hand joints via three-layer vanilla GCNs, which can mitigate the over-smoothing problem commonly encountered in existing GCN [22]. To construct the connections among multiple hand joints, we convert the $256 \times 16 \times 16$-sized joint-highlighted hand feature map into N 256-sized vector-based descriptors of N hand joints, via a 1×1 convolution with a flattening operation. Based on the graph data, we construct an undirected graph: $G =< R, E >$, where R and E denote the node and edge sets of the graph, respectively. Each graph node in R represents a feature descriptor of a hand joint, and each graph edge $e_{i,j} \in E$ denotes a connection between the i-th and j-th hand-joints. With the propagation of the graph (i.e., G), we update the node features via the three-layer vanilla graph convolution, as follows:

$$X^{k+1} = \sigma(A^k X^k W^k), \tag{2}$$

where $X^k \in \mathbb{R}^{N \times F_k}$ represents the input graph node feature descriptors of N hand joints of a hand with the dimensions of F_k at the k-th (k=1, 2, 3) graph convolution layer and where $X^{k+1} \in \mathbb{R}^{N \times F_{k+1}}$ represents the output node feature descriptors of the k-th layer graph convolution. $A^k \in \mathbb{R}^{N \times N}$ denotes the adjacency matrix of the edge connections in E,

and the adjacency matrix is initialized to an identity matrix. $W^k \in \mathbb{R}^{F_k \times F_{k+1}}$ is the learnable weight matrix of the k-th layer graph convolution, and σ indicates the ReLU activate function. With the three-layer graph propagation, the N 256-sized hand-joint feature descriptors (e.g., $X^1 \in \mathbb{R}^{N \times 256}$) can be regressed into N three-dimensional (3D) coordinate vectors of the N hand-joints of a hand (e.g., $X^4 \in \mathbb{R}^{N \times 3}$).

To better model the 3D positions of interacting hand joints, we minimize the 3D coordinate information loss (referred to as $Loss_{3D}$) during GCN training as follows.

$$Loss_{3D} = \frac{1}{M} \sum_{m=1}^{M} (\left\| \tilde{J}_m^{3d} - J_m^{3d} \right\|^2), \tag{3}$$

where $\tilde{J}_m^{3d} \in \mathbb{R}^{N \times 3}$ and $J_m^{3d} \in \mathbb{R}^{N \times 3}$ denote the predicted and ground-truth 3D coordinate vectors of N hand-joints in the m-th training image, respectively. M denotes the total number of hand pose images in the training sample set. By doing this, the vanilla GCNs can better preserve the 3D information of the hand joints, such that the latent relationships among interacting hand joints can be modeled.

D. Hand-joint Feature Refinement

Hand-joints of a hand are usually distributed following the strict hand-skeleton structure. In another word, a hand-joint of a hand usually has only one to two neighboring hand-joints, and it usually has close connections with the neighboring hand-joints due to the small distance between them. Motivated by this, based on the prior knowledge of hand-skeleton structure, we further employ semantic graph convolution (SemGConv) [23] with two attention mechanisms (i.e., self-attention and cross-attention) to refine the 3D hand-joint positions. Specifically, we first construct an adjacency matrix $A_{dj} \in \{0,1\}^{N \times N}$ to define the physical hand-joint connections to represent the skeleton relationship of hand-joints, in which $a_{ij} \in A_{dj}$=1 denotes an edge connection between the i-th and j-th joints, and a_{ij}=0 otherwise. Then, we empirically employ eight layers of SemGConv to capture the underlying close-distance relationship between each node and its neighboring nodes under the guidance of hand skeleton structure and make the feature dimension consistent to 64, which can be formulated as follows.

$$X^{l+1} = \sigma(\rho_\alpha(M \odot A_{dj}) X^l B), \tag{4}$$

where M is the learned weighting matrix used to adaptively model the connection strength among hand-joints. $\odot$ denotes the element-wise multiplication, and $M \odot A_{dj}$ represents a weighting matrix of all N nodes. The SoftMax function ρ_α normalizes the connection weights (i.e., $M \odot A_{dj}$) into a probability distribution between 0 and 1 to dynamically adjust the importance of the graph node connections. B is a shared weighting matrix of all nodes. X^l represents the input graph node feature at the l-th (k=1, 2, …, 8) semantic graph convolution layer, and X^{l+1} represents the output node feature descriptors of the l-th layer semantic graph convolution. To overcome over-smoothing, we add one residual connection between two neighboring SemGConv layers [23]. In this way, we aim to capture the implicit connections among both neighboring and non-neighboring hand-joints during graph propagation processing.

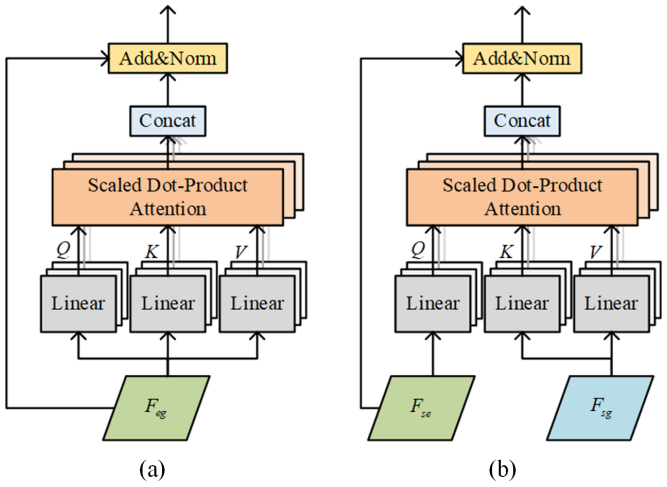

Fig. 3. The illustrations of (a) the multi-head self-attention and (b) the multi-head cross-attention.

It is well-known that attention mechanisms [32] can enable the model to divert more attention on the important regions by adaptively weighting features according to the importance of the input, which can provide benefits for long-distance hand-joint feature learning. Additionally, the cross-attention mechanism [33] enables the model to integrate important features from different sources, which can benefit the feature alignment of multiple hand joints. Motivated by this, we develop parallel four-head self-attention followed by parallel four-head cross-attention, as shown in Fig. 3, to refine the hand-joint feature dependencies. Specifically, we map the output of the eighth semantic graph convolution (referred to as F_{eg}) into Query, Key and Value (referred to as Q, K and V) via linear transformation [33], and compute the weight matrix of input feature (i.e., F_{eg}) via dot-production attention function [33] as follows:

$$Attention(Q, K, V) = Softmax(\frac{QK^T}{\sqrt{d_k}})V, \tag{5}$$

where QK^T and d_k represents the attention weights and channel dimension of the F_{eg}. After that, in parallel four-head cross-attention, we map the output of the self-attention layer (referred to as F_{se}) into query Q, and the output of the second vanilla graph convolution (referred to as F_{sg}) into the K and V, respectively, via linear transformation, and further obtain the refined global hand-joint feature representation via the cross-attention function. Similar to $Loss_{3D}$, we define a $Loss_{3D\_refine}$ loss by calculating the L2-norm difference between the predicted and the ground-truth 3D hand-joint coordinates for refined 3D hand-joint feature learning. As a result, the overall loss of the AttGCN can be written as follows:

$$Loss_{total} = Loss_{2D} + Loss_{3D} + Loss_{3D\_refine}. \tag{6}$$

Lastly, we map the refined hand-joint feature map into a 3D coordinate vector, and input it into multi-layer perception

(MLP) for 3D hand pose estimation.

IV. Experiments

In this section, we introduce three widely used datasets and implementation details of our model, and evaluate our proposed AttGCN on these datasets with different hand-skeleton structures and hand-joint numbers.

A. Datasets

The NYU database [34] contains 240,000 hand depth images, which were captured from one frontal view and two side views via a Microsoft Kinect sensor, and 80,000 images were acquired under each view. In this study, we use the frontal view for our experiment, which includes 72,000 training samples from one subject, and 8,000 testing samples collected from two subjects (one is the same as the subject of the training images) [11]. Each NYU image contains 36 hand-joint annotations. Following the evaluation protocol of [34], we select 14 annotated joints for hand pose estimation, including 2 points on each finger (except the thumb) and 3 points on the hand palm and the thumb.

The ICVL database [35] consists of 331,600 hand depth images captured from 10 distinct subjects, including 330,000 training and 1,600 testing images. Approximately 22,000 training images are in their original state [10], whereas the remaining training images are destroyed with random augmentation. For this reason, we employ 22,000 normal training images as the training set, and 1,600 testing images as the testing set for our experiment. For the ICVL database, 16 joint points are annotated in each ICVL image, including 3 finger joints and 1 palm joint point.

The MSRA database [36] contains 76,000 hand depth images captured from 9 different subjects. Each subject contains 17 hand gestures, and each hand gesture provides approximately 500 images. Following [36], we use the hand depth images of 8 subjects as the training set and the remaining images as the testing set. In each MSRA image, 21 labeled joint-points are annotated, including 4 points on each finger and 1 point on the palm.

Fig. 4 presents some representative hand images randomly sampled from the three complementary datasets. NYU provides large scale, controlled depth data with dense annotations, ICVL introduces substantial inter subject variability through multiple subjects and augmentation, while MSRA offers diverse gesture categories with richer joint annotations, enabling validation of adaptability to varied hand configurations. Together, these datasets enable a rigorous evaluation of the proposed method.

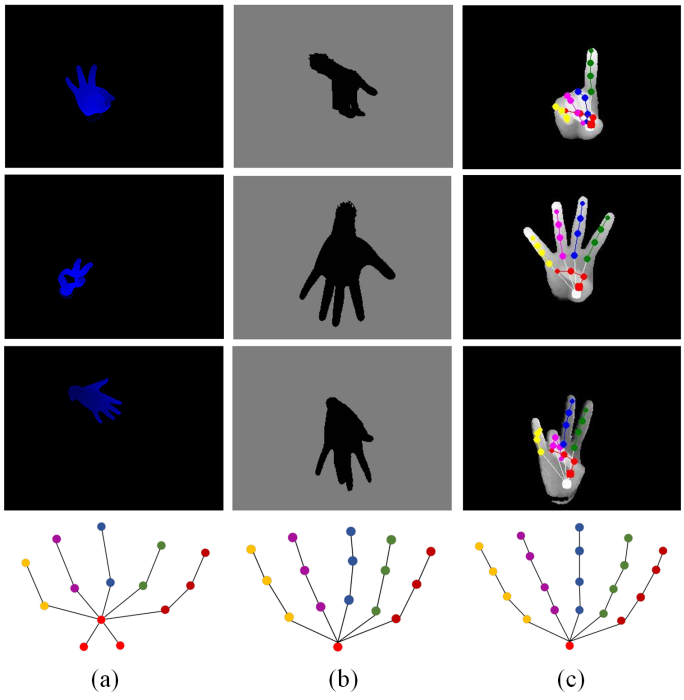

Fig. 4. The typical hand depth images randomly selected from the (a) NYU, (b) ICVL, and (c) MSRA databases, respectively, and the corresponding annotated joint-points of the images.

B. Implementation Details

In experiment, we implement the proposed network on the PyTorch framework with the AdamW optimizer, and train the network on the NVIDIA GeForce RTX 3080 GPU in an end-to-end manner. It is seen that the hand objects usually occupy a small part of the original hand depth images, as shown in Fig. 4. For this reason, we extract a cube for each hand depth image with a suitable size to crop the whole hand object with including as few non-hand areas as possible, and resize the cropped image into 256×256 pixels with normalized values ranging from -1 to 1 for 3D hand pose estimation. Moreover, we perform data augmentation operations, including in-plane rotation ([-180, 180] degrees), 3D scaling ([0.9, 1.1]), 3D translation ([-8, 8] mm) and PixDropout (the intensity controlling parameter is set to 0.15) [10], on each depth image to enrich the diversity of the training samples. Moreover, we empirically set different learning rates for training the proposed model on different databases, guided by the inherent characteristics of the samples from each dataset. Specifically, for the NYU dataset, we set the initial learning rate of the proposed model to 0.001 with a decay factor of 0.7 for every 20 epochs. For the ICVL dataset, the learning rate starts at 0.0005 with a reduced step of 0.7 for every 25 epochs, and for the MSRA dataset, we begin with a learning rate of 0.0001 with a decreasing factor of 0.2 for every 40 epochs. Uniformly, we train the proposed model for 70 epochs with a batch size of 32 for all three datasets.

C. 3D Hand Pose Estimation Results on the NYU

To evaluate our proposed method, we conduct 3D hand pose estimation experiments on the NYU database and compare the proposed method with state-of-the-art methods, including the dense regression network (DenseReg [17]), the region ensemble network (Ren-$4\times6\times6$ [37] and Ren-$9\times6\times6$ [38]), the pointwise regression network (Point-to-Point [39], V2V-PoseNet [40]), the anchor-to-joint regression network (A2J [18], A2J Transformer [41]), the adaptive weighting regression

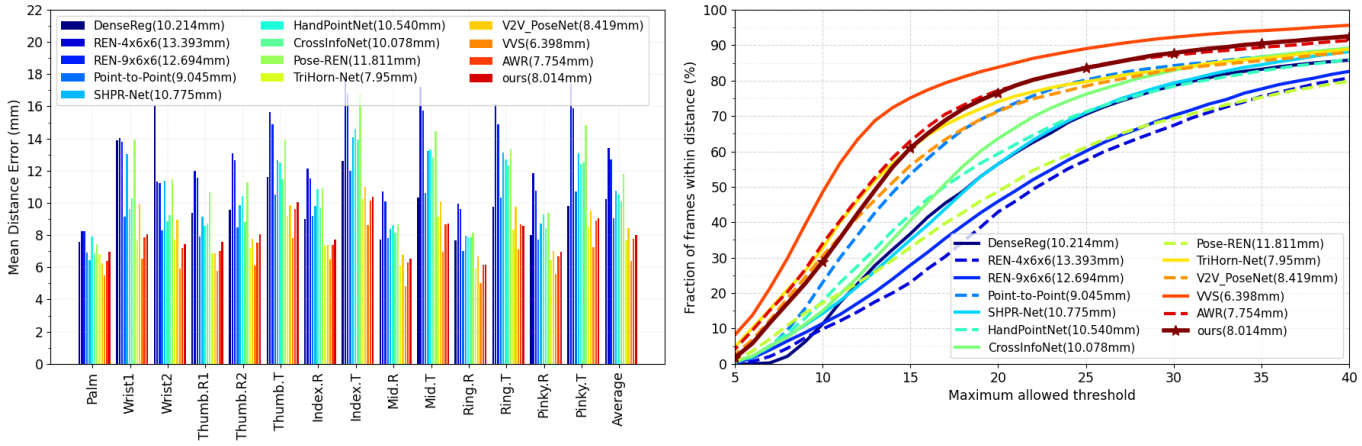

Fig. 5. 3D hand pose estimation results of (a) the mean distance errors, and (b) the fraction of successful frames with different thresholds, obtained via the different methods on the NYU database.

TABLE I
OVERALL MDES OBTAINED VIA DIFFERENT METHODS ON THE NYU DATABASE.

Methods	DenseReg [17]	REN-4x6x6 [37]	REN-9x6x6 [38]	Point-to-Point [39]	SHPR-Net [44]	CrossInfoNet [19]
MDEs	10.21	13.39	12.69	9.05	10.78	10.08
Methods	HandPointNet [43]	Pose-REN [42]	TriHorn-Net [10]	V2V-PoseNet [40]	Virtual View Selection [48]	AWR [13]
MDEs	10.54	11.81	7.95	8.41	6.40	7.75
Methods	MuTr [45]	A2J [18]	HandfoldingNet [47]	DRN [46]	A2J Transformer [41]	AttGCN (Ours)
MDEs	8.71	8.61	8.58	8.40	8.43	8.01

network (AWR [13]), the pose-guided hierarchical REN (Pose-Ren [42]), the regression-based point cloud network (HandPointNet [43]), the semantic hand pose regression network (SHPR-Net [44]), the multitask information-sharing network (CrossInfoNet [19]), the multi-stage transformer-based network (MuTr [45]) , the multi-branch network (TriHorn-Net [10]), the dynamic anchor estimation network (DRN [46]), the folding-based decoder network (HandFoldingNet [47]), and the multiview-based network (Virtual View Selection [48]). For evaluation, we calculate two widely used metrics: the mean distance error (MDE) [27] and the fraction of successful frames (FSF) [27]. MDE denotes the average Euclidean distance between the estimated and ground-truth coordinates of the hand joints. FSF is the percentage of test images for which the MDE is below a given threshold. In this experiment, we calculate 14 MDE values for 14 different hand joints of the NYU samples, including Palm, Wrist1, Wrist2, Thumb. R1, Thumb. R2, Thumb. T, Index. R, Index. T, Mid. R, Mid. T, Ring. R, Ring. T, Pinky. R, and Pinky. T ('R' and 'T' denote the root and tip of a finger, respectively). In addition, the average MDE over the 14 joints is also computed for overall comparison. Fig. 5 shows the 3D hand pose estimation results of the different methods on the NYU database, and Table 1 tabulates the detailed MDE results.

It is seen from Fig. 5(a) that our proposed method outperforms most comparison methods by achieving obviously lower MDEs for most hand-joint position estimations. Unlike most existing methods, such as Ren-$4 \times 6 \times 6$ [37], Ren-$9 \times 6 \times 6$ [38], and Pose-REN [42], which directly learn and preserve hand-joint spatial information via CNNs, other methods such as HandPointNet [43], Point-to-Point [39], and SHPR-Net [44] exploit point-cloud features to capture hand-joint spatial information without the guidance of the actual skeleton. In contrast, our proposed method extensively exploit pose-related hand-joint features through 2D cross-channel feature interactions, and further captures the underlying long- and short-distance relationships among the hand joints to refine the 3D spatial information with actual hand-skeleton guidance. As a result, fine-grained hand-joint positions are obtained, leading to small finger-wise MDEs. The proposed method achieves a slightly higher MDE than CrossInfoNet and HandPointNet for palm-joint position estimation. This may be because the palm-point is located only at the center of the hand and covers a relatively larger region than a finger, whereas our proposed AttGCN primarily considers the overall hand-joint coordinate loss during training, ignoring the small gap between the ground-truth and predicted palm-joint positions. In addition, Fig. 5(b) shows that our proposed method achieves a higher

This article has been accepted for publication in IEEE Transactions on Multimedia. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMM.2026.3724754

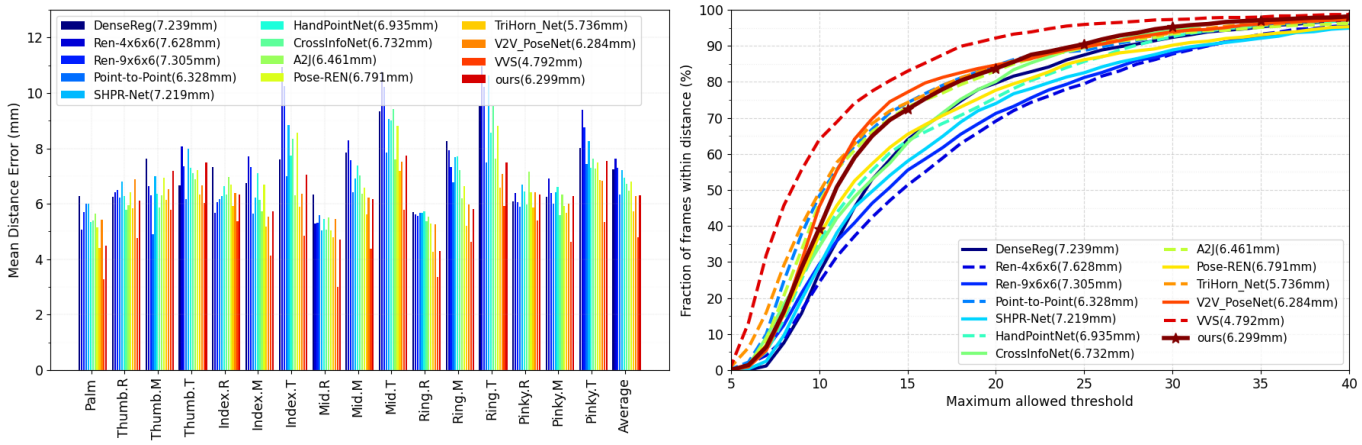

Fig. 6. The 3D hand pose estimation results of (a) the mean distance errors, and (b) the fraction of successful frames with different thresholds, achieved by different methods on the ICVL database.

TABLE II
OVERALL MDES OBTAINED VIA DIFFERENT METHODS ON THE ICVL DATABASE.

Methods	DenseReg [17]	REN-4x6x6 [37]	REN-9x6x6 [38]	Point-to-Point [39]	SHPR-Net [44]	HandPointNet [43]
MDEs	7.30	7.63	7.31	6.33	7.22	6.94
Methods	CrossInfoNet [19]	A2J [18]	Pose-REN [42]	TriHorn-Net [10]	V2V-PoseNet [40]	Virtual View Selection [48]
MDEs	6.73	6.46	6.79	6.08	6.28	4.79
Methods	AWR [13]	HandfoldingNet [47]	DRN [46]	AttGCN (Ours)		
MDEs	5.98	5.95	5.98	6.30		

FSF than state-of-the-art methods for the same distance error threshold. This is because AttGCN effectively constructs the 3D spatial structure of a hand with fine-grained hand-joint coordinates, which greatly reduces the effect of self-occlusion in the NYU hand images, thereby allowing smaller distance error thresholds to be applied to the samples.

Moreover, we tabulate the overall MDEs of all hand joints for the different methods in Table I. For a more comprehensive comparison, we also include the overall MDE results of representative methods reported in their respective publications, including 3DCNN [49], V2V-PoseNet [40], TriHorn-Net [10], DRN [46], HandFoldingNet [47] methods. It can be seen that our method is superior to most comparison methods with a competitive MDE result, demonstrating the promising effectiveness of the proposed method on 3D hand pose estimation. Notably, our method obtains a slightly higher MDE than Virtual View Selection [48], AWR [13], and Trihorn-Net [10], possibly because these methods employ specially designed depth information exploration strategies such as multiview fusion and dense prediction, leading to lower MDE results during hand pose estimation. It is also worth noting that our proposed method exhibits faster learning speed with fewer model parameters than most existing methods (see Table IV), demonstrating a better trade-off when both effectiveness and efficiency are the main concerns. .

D. 3D Hand Pose Estimation Results on the ICVL

In the ICVL database, 16 hand joints are annotated. We compare the proposed method with the nine comparison methods by calculating the MDEs for the 16 hand-joint (including Palm, Thumb. R, Thumb. M, Thumb. T, Index. R, Index. M, Index. T, Mid. R, Mid. M, Mid. T, Ring. R, Ring. M, Ring. T, Pinky. R, Pinky. M, and Pinky. T, where 'R', 'M', and 'T' denote the root, middle, and tip of a finger, respectively), as well as the FSFs. Fig. 6 shows the MDE and FSF results of the different methods.

We can see from Fig. 6 that our proposed method outperforms most state-of-the-art models by achieving significantly lower MDEs and higher FSFs for most thresholds. This is because our proposed AttGCN specifically captures the 2D spatial and 3D location information for each hand joint via 2D heatmap and 3D coordinate loss functions during coarse-to-fine pose feature learning. This significantly enhances the accuracy of hand-joint position estimation, leading to lower MDE for most joint points and a higher FSF. In addition, the proposed method shows a larger drop in FSF than the point-to-point method when the

TABLE III
OVERALL MDES OBTAINED VIA DIFFERENT METHODS ON THE MSRA DATABASE.

Methods	DenseReg [17]	3DCNN [49]	REN-9x6x6 [38]	Point-to-Point [39]	SHPR-Net [44]	HandPointNet [43]	CrossInfoNet [19]
MDEs	7.23	9.6	9.79	7.71	7.76	8.51	7.86
Methods	Pose-REN [42]	HandfoldingNet [47]	TriHorn-Net [10]	AWR [13]	V2V-PoseNet [40]	AttGCN (Ours)	
MDEs	8.65	7.34	7.65	7.15	7.49	7.47	

threshold is below 20 mm. This is possibly because the ICVL dataset is small-scale with limited hand pose variance, which may not provide sufficient information to train the multiple graph convolution layers of AttGCN, thereby limiting the generalizability of the proposed method. Moreover, Table II tabulates the overall MDEs of all the hand joints on the ICVL database reported by the different methods in the respective literature. Our proposed method achieves better or comparable estimation performance compared with most existing methods.

E. 3D Hand Pose Estimation Results on the MSRA

In the MSRA dataset, most methods cannot calculate the MDE for each hand joint of the MSRA samples due to a lack of ground-truth labels. Therefore, we report the overall hand-joint MDEs of the MSRA samples for different methods, as shown in Table III, where the results of the comparison methods such as DenseReg [17], 3DCNN [49], Ren-$9 \times 6 \times 6$ [38], HandPointNet [43], Point-to-Point [39], SHPR-Net [44], CrossInfoNet [19], Pose-REN [42], TriHorn-Net [10], AWR [13], V2V-PoseNet [40], and HandFoldingNet [47] , are provided in the literature. Our proposed method outperforms most regression methods, such as 3DCNN, REN-$9 \times 6 \times 6$, HandPointNet and Pose-REN. This is mainly because we introduce heatmap loss beyond the regression loss to capture 2D joint-specific information, which effectively preserves the 2D spatial locations of the hand-joint points, and improves overall hand pose estimation performance. In addition, our proposed method yields a higher MDE than 3D point cloud-based methods such as SHPR-Net and the point-to-point. The reason is that these 3D point cloud-based methods use 3D hand point clouds as inputs, which can better capture hand-joint positions and hand occlusions from different views. In contrast, our method uses only 2D depth images as inputs, significantly increasing the difficulty of estimating the 3D hand-joint positions for MSRA samples with various pose variations across different subjects. Notably, our approach is more lightweight than the 3D point cloud-based methods, demonstrating the promising efficiency of the proposed method for 3D hand pose estimation from depth images.

F. Analysis of Computational Efficiency and Running Speed

Following [47] [41], we evaluate the computational efficiency and learning speed of the proposed method on the NYU dataset in terms of parameters quantity (Params) [47], floating-point operations (FLOPs) [47], and frames per second (FPS) [41]. Of them, Params [47] denotes the parameter quantity of a network, FLOPs [47] represents the computational effort and algorithmic complexity, and FPS [41] indicates the running speed during model testing. Table IV tabulates the Params, FLOPs and speed obtained by different methods. It can be observed that our method usually obtains lower FLOPs and faster FPS than most state-of-the-art methods with comparable Params, demonstrating the competitive computational efficiency of the proposed method.

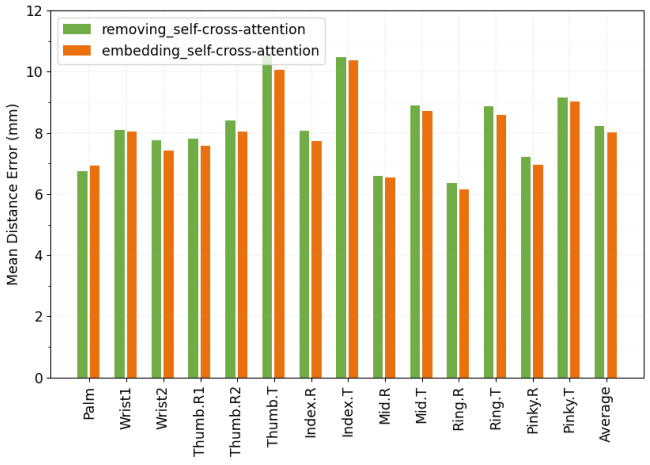

Fig. 7. Hand-joint-wise MDE of the proposed method with using and removing both the self- and cross-attention mechanisms, on the NYU dataset.

It can be seen that the proposed AttGCN achieves a favorable balance between accuracy and efficiency, making it well suited for diverse real world applications such as touchless human–computer interaction and sign language recognition. Its coarse-to-fine graph reasoning and occlusion aware design reliably handle highly articulated hand motions, while an inference speed of approximately 208 fps on a standard GPU fully supports real time deployment.

G. Ablation Analysis

Our proposed AttGCN consists of multiple coarse-to-fine hand pose feature learning subnetworks. In the 2D joint-highlighted hand feature learning subnetwork, we design a cross-channel interaction (CCI) module to capture the 2D joint-highlighted hand features. To evaluate the effectiveness

This article has been accepted for publication in IEEE Transactions on Multimedia. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMM.2026.3724754

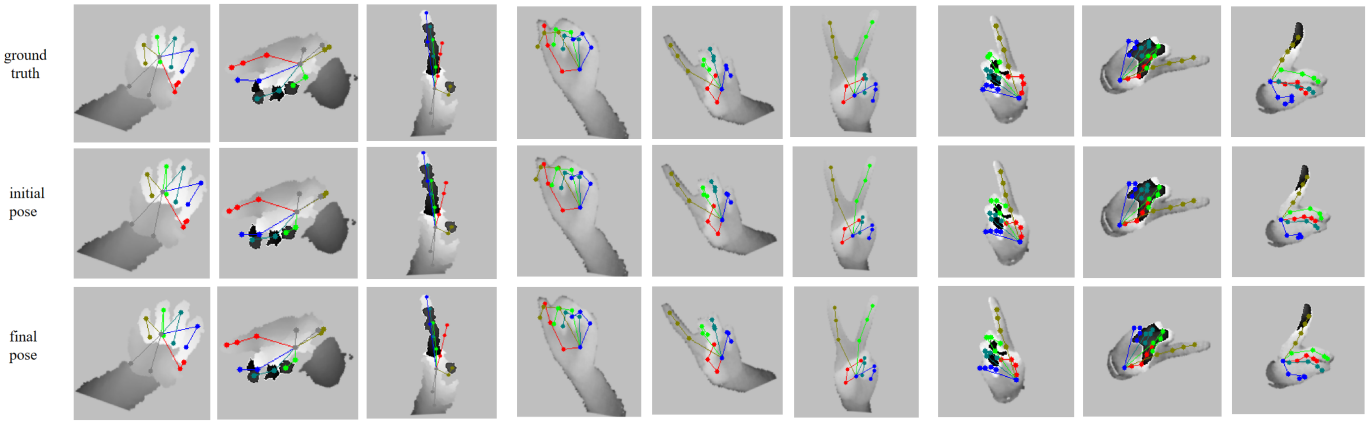

Fig. 8. The estimated 3D hand-joint positions on the (a) NYU, (b) ICVL and (c) MSRA samples with self-occlusions.

TABLE IV
COMPARISONS OF THE MODEL SIZE, TOTAL NUMBER OF FLOATING-POINT OPERATIONS, AND INFERENCE SPEED AMONG DIFFERENT METHODS.

Methods	Params (M)	FLOPs (G)	Speed (FPS)
HandFoldingNet [47]	**1.28**	-	84
A2J [18]	44.7	-	105.06
A2J Transformer [41]	42	-	24.81
Virtual View Selection [48]	-	-	6.40
AWR [13]	34	14.4	-
TriHorn-Net [10]	7.81	13.43	-
AttGCN (Ours)	26.06	**5.88**	**208.17**

TABLE V
THE MDES OF THE PROPOSED METHOD WHEN THE CCI AND ECA ARE EMBEDDED AND BOTH ARE REMOVED ON THE NYU DATASET.

ECA	CCI	MDE
×	×	8.201
√	×	8.428
×	√	**8.014**

of the CCI, we remove it from our proposed network, replace it with single-channel efficient channel attention (ECA) [32], and calculate its MDEs on the NYU dataset, as shown in Table V. The proposed method using the CCI outperforms both variants, showing the effectiveness of the CCI. This is mainly because the CCI module can effectively establish the associations among the inter- and intra-scale hand features so that more comprehensive hand-joint spatial information can be obtained.

In hand-joint relationship learning, we develop three-layer vanilla GCNs to capture the latent interacting relationships among both intra- and inter-finger hand-joints. To evaluate it, we use a three-layer MLP, SemGCN [23], Modulated GCN [50], ChebConv GCN [51], and PreaggGCN [52] to replace the vanilla GCN, respectively, and their estimation results are tabulated in Table VI. The proposed method with the embedding vanilla GCN achieves an obviously lower MDE than the other five replacement versions. This finding demonstrates that the vanilla GCN can better capture the latent hand-joint relationships with fewer constraints of adjacency matrix initialization and graph convolution operation constraints.

TABLE VI
THE MDES OF THE PROPOSED ATTGCN WITH EMBEDDING THE MLP AND VARIOUS GCNS IN THE LATENT HAND-JOINT RELATIONSHIP LEARNING SUB-NETWORK ON THE NYU DATASET.

Component Type	MDE
MLP	8.462
Semantic GCN	8.606
Modulated GCN	8.275
PreAggGCN	8.796
ChebGCN	8.637
Vanilla GCN	**8.014**

TABLE VII
THE PERFORMANCE OF MODELS THAT EMPLOY THE DIFFERENT ATTENTION MECHANISMS IN THE LATENT HAND-JOINT RELATIONSHIP LEARNING SUBNETWORK.

Self-Attention	Cross-Attention	MDE
×	×	8.217
√	×	8.256
×	√	8.181
√	√	**8.014**

In addition, we employ both self-attention and cross-attention to refine the final learned hand-joint features. Table VII tabulates the estimation results of the proposed method by respectively removing one and both of the two attention mechanisms from the refinement subnetwork. This clearly shows that, by embedding both the self- and cross-attention, the proposed method can significantly improve hand pose estimation. This is because the two attention mechanisms can specifically learn the fine-grained hand-joint positions with small-distance correlation, such that the accuracy of hand pose estimation can be improved.

Moreover, Fig. 7 depicts the comparative hand-joint-wise MDEs obtained by the proposed method when both attention mechanisms are used and removed from the NYU dataset.

By embedding both the self- and cross-attention mechanisms, the proposed AttGCN can focus more on correction learning among the finger joints, such that the finger-joint positions can be precisely located for more accurate hand pose estimation.

H. Visualization

Finally, Fig. 8 presents representative 3D hand pose estimation results of the proposed AttGCN on samples from the NYU, ICVL, and MSRA datasets. The results demonstrate that our method accurately estimates the 3D positions of hand joints even in depth images exhibiting severe self-occlusions. This robustness is primarily attributed to the use of multiple graph convolutional layers, which effectively capture the structural dependencies among hand joints and enable reliable reconstruction of the 3D hand skeleton.

V. Conclusion

In this paper, we propose a new coarse-to-fine 3D position feature learning network, namely AttGCN, for 3D hand pose estimation from depth images. Our proposed method consists of a cross-channel interaction module to enhance joint-specific hand features, and multiple graph convolutions with attention embeddings to generate a fine-grained 3D joint position-level representation while preserving the global relationships among hand joints, thereby achieving improved performance on hand pose estimation. Notably, our proposed method focuses primarily on single-hand pose estimation without considering temporal dynamics and interactions between hands (such as handshakes and object manipulation with hands), which makes it less effective for multi-hand pose estimation. This suggests an interesting direction for extending our proposed AttGCN to exploit more interactive features for two-hand interactive pose estimation under complex self-hand and cross-hand occlusions.

References

[1] Y. Hu, O. Tavakoli, M. J. Black, and D. Tzionas, "InterCap: Joint markerless 3D tracking of humans and objects in interaction," *International Journal of Computer Vision*, pp. 1–16, 2024.

[2] H. Laga, L. V. Gool, F. Boussaid, M. Bennamoun *et al.*, "A survey on deep learning techniques for stereo-based depth estimation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 4, pp. 1738–1764, 2020.

[3] Y. Wang, H. Peng, L. Yang, J. Pan, T. Komura, W. Wang *et al.*, "Hierarchical temporal transformer for 3D hand pose estimation and action recognition from egocentric RGB videos," *Computing Research Repository*, pp. 21 243–21 253, 2023.

[4] G. Pavlakos, D. Shan, I. Radosavovic, A. Kanazawa, D. Forsyth, J. Malik *et al.*, "Reconstructing hands in 3D with transformers," *Computing Research Repository*, 2024.

[5] 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 *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020, pp. 120–137.

[6] J. Z, C. X, Y. G, L. C *et al.*, "Realistic depth image synthesis for 3D hand pose estimation," *IEEE Transactions on Multimedia*, vol. 26, pp. 5246–5256, 2024.

[7] H. Lu, S. Gou, and R. Li, "SPMHand: Segmentation-guided progressive multi-path 3d hand pose and shape estimation," *IEEE Transactions on Multimedia*, vol. 26, pp. 6822–6833, 2024.

[8] 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.

[9] P. Ren, H. Sun, Q. Qi, J. Wang, and W. Huang, "SRN: Stacked regression network for real-time 3D hand pose estimation," in *Proceedings of British Machine Vision Conference*, 2019, p. 112.

[10] M. Rezaei, R. Rastgoo, and V. Athitsos, "TriHorn-Net: A model for accurate depth-based 3D hand pose estimation," *Expert Systems with Applications*, vol. 223, p. 119922, 2023.

[11] P. Ren, H. Sun, J. Hao, Q. Qi, J. Wang, and J. Liao, "Pose-guided hierarchical graph reasoning for 3-D hand pose estimation from a single depth image," *IEEE Transactions on Cybernetics*, pp. 315–328, 2023.

[12] X. Jiang, H. Tao, J. N. Hwang, and Z. Fang, "A multiscale coarse-to-fine human pose estimation network with hard keypoint mining," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 54, no. 3, pp. 1730–1741, 2024.

[13] 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*, 2020, pp. 11 061–11 068.

[14] X. Zhang and F. Zhang, "Pixel-wise Regression: 3D hand pose estimation via spatial-form representation and differentiable decoder," *IEEE Transactions on Multimedia*, pp. 166–176, 2022.

[15] J. Han, L. Shao, D. Xu, and J. Shotton, "Enhanced computer vision with microsoft kinect sensor: A review," *IEEE Trans. Cybern.*, vol. 43, no. 5, pp. 1318–1334, 2013.

[16] A. Sinha, C. Choi, and K. Ramani, "DeepHand: Robust hand pose estimation by completing a matrix imputed with deep features," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 4150–4158.

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

[18] F. Xiong, B. Zhang, Y. Xiao, Z. Cao, T. Yu, J. T. Zhou, and J. Yuan, "A2J: Anchor-to-joint regression network for 3D articulated pose estimation from a single depth image," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2019, pp. 793–802.

[19] K. Du, X. Lin, Y. Sun, and X. Ma, "CrossInfoNet: Multi-task information sharing based hand pose estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 9896–9905.

[20] X. Chen *et al.*, "TSSRGCN: Temporal spectral spatial retrieval graph convolutional network for traffic flow forecasting," in *Proceedings of IEEE International Conference on Data Mining (ICDM)*, 2020, pp. 954–959.

[21] G. Liang, X. Lan, J. Wang, J. Wang, and N. Zheng, "A limb-based graphical model for human pose estimation," in *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2018, pp. 1080–1092.

[22] B. Doosti, S. Naha, M. Mirbagheri, and D. J. Crandall, "HOPE-Net: A graph-based model for hand-object pose estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 6608–6617.

[23] L. Zhao, X. Peng, Y. Tian, M. Kapadia, and D. N. Metaxas, "Semantic graph convolutional networks for 3D human pose regression," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 3425–3435.

[24] T. Xu and W. Takano, "Graph stacked hourglass networks for 3D human pose estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 16 105–16 114.

[25] G. Chen *et al.*, "A survey of the four pillars for small object detection: Multiscale representation, contextual information, super-resolution, and region proposal," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 52, no. 2, pp. 936–953, 2022.

[26] Q. Fu, X. Liu, R. Xu, J. Niebles, and K. M. Kitani, "Deformer: Dynamic fusion transformer for robust hand pose estimation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023, pp. 23 600–23 611.

[27] Y. Wang, L. Chen, J. Li, and X. Zhang, "HandGCNFormer: A novel topology-aware transformer network for 3D hand pose estimation," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2023, pp. 5675–5684.

[28] W. Zhao, W. Wang, and Y. Tian, "GraFormer: Graph-oriented transformer for 3D pose estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 20 438–20 447.

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

[30] T.-Y. Lin, P. Dollár, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, p. 936–944.

[31] J. Hu, L. Shen, S. Albanie, G. Sun, and E. Wu, "Squeeze-and-excitation networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 7132–7141.

[32] Q. Wang, B. Wu, P. Zhu, P. Li, W. Zuo, and Q. Hu, "ECA-Net: Efficient channel attention for deep convolutional neural networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 11 534–11 542.

[33] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Neural Information Processing Systems*, 2017.

[34] J. Tompson, M. Stein, Y. Lecun, and K. Perlin, "Real-time continuous pose recovery of human hands using convolutional networks," *ACM Transactions on Graphics*, vol. 33, no. 5, pp. 1–10, 2014.

[35] D. Tang, H. J. Chang, A. Tejani, and T.-K. Kim, "Latent Regression Forest: Structured estimation of 3D articulated hand posture," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014, pp. 3786–3793.

[36] X. Sun, Y. Wei, S. Liang, X. Tang, and J. Sun, "Cascaded hand pose regression," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 824–832.

[37] H. Guo, G. Wang, X. Chen, C. Zhang, F. Qiao, and H. Yang, "Region Ensemble Network: Improving convolutional network for hand pose estimation," in *IEEE International Conference on Image Processing (ICIP)*, 2017, pp. 4512–4516.

[38] G. Wang, X. Chen, H. Guo, and C. Zhang, "Region ensemble network: Towards good practices for deep 3D hand pose estimation," *Journal of Visual Communication and Image Representation*, vol. 55, pp. 404–414, 2018.

[39] 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.

[40] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 5079–5088.

[41] C. Jiang, Y. Xiao, C. Wu, M. Zhang, J. Zheng, Z. Cao, and J. Zhou, "A2J-Transformer: Anchor-to-joint transformer network for 3D interacting hand pose estimation from a single rgb image," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023, pp. 8846–8855.

[42] 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, 2020.

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

[44] 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.

[45] J. Kanis, I. Gruber, Z. Krňoul, M. Boháček, J. Straka, and M. Hrúz, "MuTr: Multi-stage transformer for hand pose estimation from full-scene depth image," in *Sensors*, 2023, p. 5509.

[46] H. Xing, J. Yang, and Y. Xiao, "Learning dynamic relationship between joints for 3D hand pose estimation from single depth map," *Journal of Visual Communication and Image Representation*, vol. 92, p. 103803, 2023.

[47] W. Cheng, J. Park, and J. Ko, "HandFoldingNet: A 3D hand pose estimation network using multiscale-feature guided folding of a 2D hand skeleton," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 11 260–11 269.

[48] J. Cheng, Y. Wan, D. Zuo, C. Ma, J. Gu, P. Tan, H. Wang, X. Deng, and Y. Zhang, "Efficient virtual view selection for 3d hand pose estimation," in *Proceedings of the Thirty-Sixth AAAI Conference on Artificial Intelligence*, 2022, p. 419–426.

[49] 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 Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 1991–2000.

[50] Z. Zou and W. Tang, "Modulated graph convolutional network for 3D human pose estimation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 11 477–11 487.

[51] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *Advances In Neural Information Processing Systems*, 2016, pp. 3844–3852.

[52] K. Liu, R. Ding, Z. Zou, L. Wang, and W. Tang, "A comprehensive study of weight sharing in graph networks for 3D human pose estimation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020, pp. 318–334.