Collaborative Learning of Gesture Recognition and 3D Hand Pose Estimation with Multi-Order Feature Analysis

Siyuan Yang[1,2], Jun Liu[3⋆], Shijian Lu[4], Meng Hwa Er[2], and Alex C. Kot[2]

[1] Rapid-Rich Object Search (ROSE) Lab, Interdisciplinary Graduate Programme,
Nanyang Technological University, Singapore
[2] School of Electrical and Electronic Engineering, Nanyang Technological University,
Singapore
[3] Information Systems Technology and Design Pillar, Singapore University of
Technology and Design, Singapore
[4] School of Computer Science and Engineering, Nanyang Technological University,
Singapore
siyuan005@e.ntu.edu.sg, jun_liu@sutd.edu.sg, {shijian.Lu, emher,
eackot}@ntu.edu.sg

Abstract. Gesture recognition and 3D hand pose estimation are two highly correlated tasks, yet they are often handled separately. In this paper, we present a novel collaborative learning network for joint gesture recognition and 3D hand pose estimation. The proposed network exploits joint-aware features that are crucial for both tasks, with which gesture recognition and 3D hand pose estimation boost each other to learn highly discriminative features. In addition, a novel multi-order multi-stream feature analysis method is introduced which learns posture and multi-order motion information from the intermediate feature maps of videos effectively and efficiently. Due to the exploitation of joint-aware features in common, the proposed technique is capable of learning gesture recognition and 3D hand pose estimation even when only gesture or pose labels are available, and this enables weakly supervised network learning with much reduced data labeling efforts. Extensive experiments show that our proposed method achieves superior gesture recognition and 3D hand pose estimation performance as compared with the state-of-the-art.

Keywords: Gesture Recognition · 3D Hand Pose Estimation · Multi-Order Multi-Stream Feature Analysis · Slow-Fast Feature Analysis · Multi-Scale Relation

1 Introduction

Gesture recognition and 3D hand pose estimation are both challenging and fast-growing research topics which have received contiguous attention recently due to their wide range of applications in human-computer interaction, robotics,

⋆ Corresponding author.

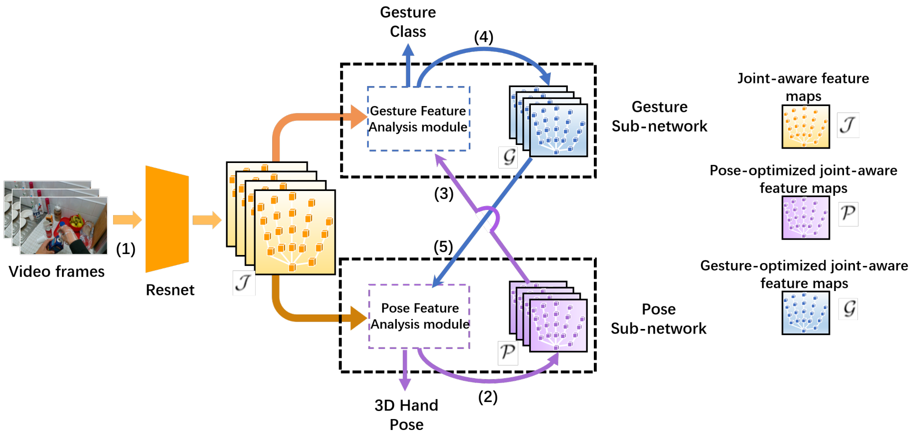

Fig. 1. Overview of our proposed network architecture for gesture recognition and 3D hand pose estimation from videos. The input is video frames, and the output is predicted gesture class of the video and 3D hand joint locations of each video frame. The process flow of our network can be divided into 5 stages: (1) Generating $\mathcal{J}$. (2) Generating $\mathcal{P}$ and Predicting 3D Hand Pose. (3) Aggregating input to Gesture Sub-Network. (4) Generating $\mathcal{G}$ and Recognizing Gesture Class. (5) Aggregating input to Pose Sub-Network. (As shown by the (1) - (5) in this Figure). Stage (2) to (5) are operated in an iterative way (details are introduced in Sec. 3.1).

virtual reality, augmented reality, etc. The two tasks are closely correlated as they both leverage heavily on joint-aware features, i.e. features related to the hand joints [19, 40]. On the other hand, the two tasks are often tackled separately by dedicated systems [1, 4, 9, 23, 24] . Though some recent efforts [13, 22, 29] attempt to handle the two tasks at one go, it does not consider to iteratively gain benefits from mutual learning of them.

In this paper, we propose to perform gesture recognition and 3D hand pose estimation mutually. We design a novel collaborative learning strategy to exploit joint-aware features that are crucial for both tasks, with which gesture recognition and 3D hand pose estimation can learn to boost each other progressively, as illustrated in Fig. 1.

Inspired by the successes [28, 34] that use motion information for human activity recognition in videos, we exploit motion information for better gesture recognition by focusing more on joint-aware features. Specifically, we distinguish slowly and fast-moving hand joints and exploit such motion information in the intermediate network layers to learn enhanced and enriched joint-aware features. Beyond that, we propose a multi-order multi-stream feature analysis module that exploits more discriminative and representative joint motion information according to the intermediate joint-aware features.

Additionally, annotating 3D hand poses is often very laborious and time-consuming. To address this issue, we propose a weakly supervised 3D pose estimation technique that can learn accurate 3D pose estimation models from the gesture labels which are widely available in many video data. We observe that the weakly supervised learning improve the 3D pose estimation significantly when only a few samples with 3D pose annotations are included, largely because the exploited joint-aware features that are useful for both gesture recognition and 3D hand pose estimation tasks. At the other end, the weakly supervised learning can also learn accurate gesture estimation models from hand image sequences with 3D pose annotations with similar reasons.

The contributions of this work can be summarized from four aspects. *First*, we propose a novel collaborative learning network that leverage joint-aware features for both gesture recognition and 3D hand pose estimation simultaneously. To the best of our knowledge, this is the first network that exploits and optimizes the joint-aware features for both gesture recognition and 3D hand pose estimation. *Second*, it designs a multi-order feature analysis module that employs a novel slow-fast feature analysis scheme to learn joint-aware motion features which improves the gesture recognition greatly. *Third*, it designs a multi-scale relation module to learn hierarchical hand structure relations at multiple scales which enhances the performance of gesture recognition clearly. *Fourth*, we propose a weakly supervised learning scheme that is capable of leveraging hand pose (or gesture) annotations to learn powerful gesture recognition (or pose estimation) model. The weakly supervised learning greatly relieves the data annotation burden especially considering the very limited annotated 3D pose data and the wide availability of annotated hand gesture data.

2 Related Work

Gesture and action recognition. In the early stage, many gesture and action recognition methods were developed based on handcrafted features [14, 15, 32, 33]. With the advance of deep learning, Convolutional neural networks (CNNs) [7, 28, 30, 31, 34, 36, 38, 39] have been applied to gesture recognition and action recognition. Simonyan and Zisserman [28] proposed a two-stream architecture, where one stream operates on RGB frames, and the other on optical flow. Many works follow and extend their framework [7, 30, 36]. They all use the optical flow as the motion information. Wang *et al.* [34] built a new motion representation: RGB difference, which stacks the differences between consecutive frames, to save the time of optical flow extraction. The calculation process of optimal flow [28, 34] and RGB difference [34] are all pre-processed which is outside of the learning process.

Inspired by the above-mentioned works, in our work, we propose a new multi-order multi-stream feature analysis module, which is conducted at the intermediate features that capture more discriminative and representative motion information as compared to the original video data. Specifically, a slow-fast feature analysis module is added to consolidate the features of both the slowly and fast-moving joints at multiple orders which significantly enhances the gesture-aware features for more reliable gesture recognition.

3D Hand pose estimation. 3D hand pose estimation from RGB images has received much attention recently [2, 5, 6, 23, 26, 40]. However, only a few works [22, 29] focused on performing gesture recognition and 3D hand pose estimation from the RGB videos jointly. Tekin *et al.* [29] predicted hand pose and action categories first, and then use the predicted information to do the gesture recognition.

We propose to leverage the joint-aware features for mutual 3D pose estimation and gesture recognition. A novel collaborative learning method is proposed which iteratively boosts the performance of the two tasks by optimizing the joint-aware features which are crucial for both tasks. It also enables the weakly-supervised learning for 3D hand pose estimation.

Joint gesture/action recognition and 3D pose estimation. Gesture (or action) recognition and 3D pose estimation are highly related, thus many works performed gesture (or action) recognition based on the results of pose estimation. In the Skeleton-based gesture (or action) recognition [9, 16, 19, 17, 20, 24], joints' location (pose) information is used for recognizing the gesture (or action) categories. In the RGB-based action recognition, Liu *et al.* [21] also proposed to recognize human actions based on the pose estimation maps. Nie *et al.* [35] and Luvizon *et al.* [22] performed pose estimation and action recognition in a single network, yet they did not consider these two tasks mutually to optimize the performance of each other, i.e., they performed the two tasks either in a parallel way or in a sequential way.

Different from the aforementioned methods, we design a new collaborative learning method that boosts the learning of gesture recognition and 3D hand pose estimation in an *iterative* manner as shown in Fig. 1. To the best of our knowledge, our method is the first that learns gesture-aware and hand pose-aware information for boosting the two tasks progressively.

Weakly-Supervised learning on 3D hand pose estimation. In the past few years, several works focus on weakly-supervised learning in 3D pose estimation and 3D hand pose estimation areas, since it is hard to obtain the 3D pose annotations. Cai *et al.* [3, 4] proposed a weakly-supervised adaptation method by bridging the gap between fully annotated images and weakly-labelled images. Zhou *et al.* [37] transformed knowledge from 2D pose to 3D pose estimation network using re-projection constraint to 2D results. Chen *et al.* [8] used the multi-view 2D annotation as the weak supervision to learn a geometry-aware 3D representations.

All aforementioned methods still used 2D joint information as the weak supervision to generate 3D hand poses. Differently, we propose that the gesture label can also be used as the weak supervision for 3D hand pose estimation. Our experiments show that this weak-supervised learning method is efficient.

3 Methodology

We predict gesture categories and 3D hand joint locations directly from RGB image sequences as illustrated in Fig. 1. Specifically, the input is a sequence of RGB images centered on hand which is fed to a pre-trained ResNet [11] to learn joint-aware feature maps $\mathcal{J}$ (as shown in Fig. 1). The learned $\mathcal{J}$ are then fed to pose sub-network and gesture sub-network which learn collaboratively for more discriminative features. The whole network is trained in an end-to-end manner, more details to be presented in the following subsections.

3.1 Collaborative Learning for Gesture Recognition and 3D Hand Pose Estimation

Gesture recognition and 3D hand pose estimation are both related to the joint-level features. Joints' locations have been used for skeleton-based action recognition and gesture recognition, while gesture classes also contain potential hand posture information that is useful for hand pose estimation.

We propose a collaborative learning method that simultaneously learns the gesture features and 3D hand pose features mutually in an iterative way, as illustrated in Fig. 1. As described above, the pre-trained ResNet [11] is used to learn the joint-aware feature maps $\mathcal{J}$. Specifically, we equally divide the joint-aware feature maps $\mathcal{J}$ to N groups, where N is the number of hand joints, *i.e.* $\mathcal{J} = \{\mathcal{J}_i | i = 1, ..., N\}$, and $\mathcal{J}_i$ is the subset of feature maps representing the joint i ($i \in [1, N]$).

Pose Sub-Network: Following the previous works [18, 37, 40], we first use a Pose Feature Analysis module to estimate the 2D heatmaps based on the intermediate features for generating the 3D hand pose. The Pose Feature Analysis module is composed by two parts: 2D hand pose estimation part and depth regression part, which is similar to [18, 37, 40]. For the **2D hand pose estimation part**, its input are the joint-aware feature maps $\mathcal{J}$ and its output are N heatmaps (denoted by $\mathcal{H}$). Each map $\mathcal{H}_i$ is a $H \times W$ matrix, representing a 2D probability distribution of each joint in the image.

Follow the deep regression module in [18, 37]. We aggregate the joint-aware feature maps $\mathcal{J}$ and the generated 2D heatmaps $\mathcal{H}$ with 1×1 convolution by a summation operation, the summed feature maps are input of the **deep regression module**. Here the 1×1 convolution is used to map the generated 2D heatmaps $\mathcal{H}$ and the joint-aware feature maps $\mathcal{J}$ to the same size. The deep regression module contains a sequence of convolutional layers with pooling and a fully connected layer in order to regress the depth values $D = \{D_i | i = 1, ..., N\}$, where D_i denotes the depth value of the i_{th} joint.

Since the output of pose sub-network is the input of the gesture sub-network, and pose sub-network and gesture sub-network operate iteratively (as shown in Fig. 1), we set the input and output of pose sub-network the same size. To keep the size constant, we first duplicate the depth values to the same size of the heatmaps, and concatenate them with 2D heatmaps. For each joint, its depth value is a scalar, while heatmaps size is $H \times W$. Thus, we duplicate depth value HW to match heatmaps size to facilitate feature concatenation. Secondly, the 1×1 convolution is used to map the concatenated feature maps and the joint-aware feature maps $\mathcal{J}$ to the same size to generate the output of pose sub-network, named pose-optimized joint-aware feature maps $\mathcal{P}$ (see Fig. 1).

Gesture Sub-Network: The input of Gesture Sub-Network is obtained by aggregating the joint-aware feature maps $\mathcal{J}$ and pose-optimized joint-aware feature maps $\mathcal{P}$ with 1×1 convolution followed by a summation. The resultant feature maps are fed to the Gesture Feature Analysis module to generate the gesture-optimized joint-aware feature maps $\mathcal{G}$ and gesture category y (see Fig 1). Where the Gesture Feature Analysis module contains a sequence of convolutional layers as well as temporal convolution (TCN) layers to get the temporal relation, TCN layers are used here to predict the gesture class y.

Collaborative learning method: As shown in Fig. 1, we design a collaborative learning strategy to perform gesture recognition and 3D hand pose estimation in an iterative way. Our proposed framework's learning processes can be described in the following stages:

(1) **Generating $\mathcal{J}$**: The pre-trained ResNet [11] is used to learn the joint-aware feature maps $\mathcal{J}$.

(2) **Generating $\mathcal{P}$ and Predicting 3D Hand Pose**: The learned feature maps $\mathcal{J}$ are fed to Pose Feature Analysis module (shown in Fig. 1) to generate 3D hand poses (2D Heatmaps $\mathcal{H}$ and depth values D), and also the pose-optimized joint-aware feature maps $\mathcal{P}$.

(3) **Aggregating input to Gesture Sub-Network**: The 1×1 convolution is used to generate intermediate feature maps by aggregating the joint-aware feature maps $\mathcal{J}$ and the pose-optimized joint-aware feature maps $\mathcal{P}$.

(4) **Generating $\mathcal{G}$ and Recognizing Gesture Class**: The intermediate feature maps are fed to Gesture Feature Analysis module as input to generate the gesture-optimized joint-aware feature maps $\mathcal{G}$ and to recognize gesture category y.

(5) **Aggregating input to Pose Sub-Network**: We aggregate the gesture-optimized joint-aware feature maps $\mathcal{G}$ and the joint-aware feature maps $\mathcal{J}$ with 1×1 convolution followed by a summation. The aggregated feature maps are fed to next iteration's Pose Sub-Network as input for further feature learning.

(6) Stage **2** to **5** repeat in an iterative way to perform gesture recognition and hand pose estimation collaboratively for further improving the performance.

3.2 Multi-Order Multi-Stream Feature Analysis

As discussed in Section 2, prior studies have shown that motion information such as optical flow [28, 34] is crucial in video-based recognition. As we aim to learn joint-aware features, we propose a multi-order multi-stream feature analysis module as shown in Fig. 2 to learn the motion information based on the joint-aware features. The proposed multi-order multi-stream module participates in the Gesture Feature Analysis module (see Fig. 1).

Since the pre-trained ResNet [11] and our pose sub-network operate at the image level, the corresponding feature maps belonging to hand joints in an image. We name the image-level features as **Zero-Order Features** (denote by Zo, which stand for pose information and static information), as shown in the top line of Fig. 2, the cubes in it are feature maps of the corresponding hand joints. Zero-Order features form $N \times C \times H \times W$ tensors, where N is the total number

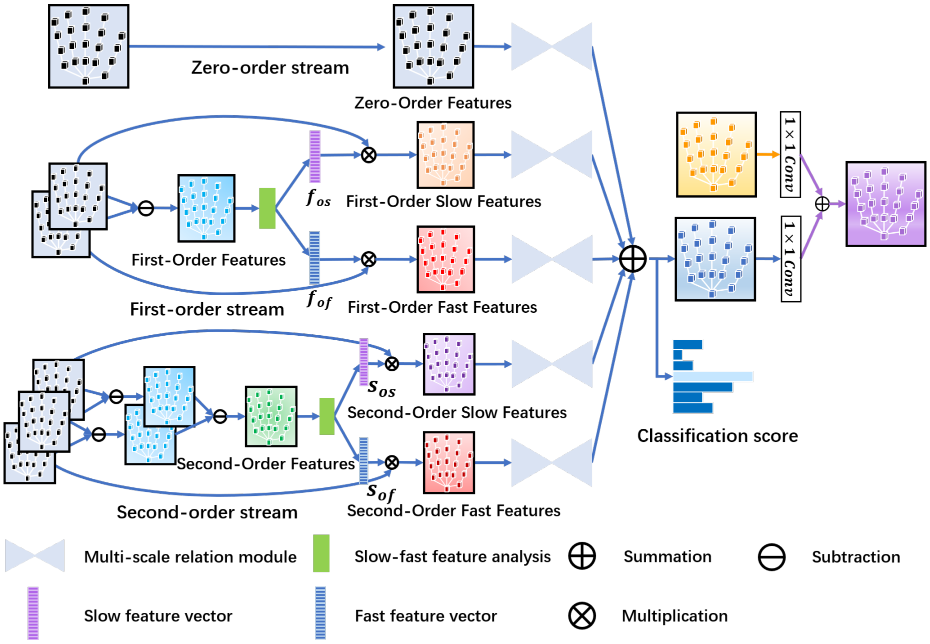

Fig. 2. Illustration of the multi-order multi-stream feature analysis module: With the zero-order features Zo as input, the multi-order multi-stream analysis generates motion information on the intermediate features including first-order slow & fast features and second-order slow & fast features. These four motion features, together with the zero-order features, are fed to five multi-scale relation modules (more details in Fig. 3), respectively, to generate gesture-optimized joint-aware feature maps $\mathcal{G}$ and gesture category y. The generated $\mathcal{G}$ are aggregated with joint-aware feature maps $\mathcal{J}$ and fed to the pose sub-network for pose feature learning. Our multi-order multi-stream feature analysis module participates in the Gesture Feature Analysis module, as shown in Fig. 1. (More description of Fig.2 are illustrated in supplementary material.)

of hand joints, C is the number of channels for each hand joint, H and W are the height and width of feature maps, respectively.

First-Order Features can be seen as velocity features. A temporal neighborhood pair of feature maps is constructed from the entire Zero-Order Features as follows:

$$\mathcal{U}_1 = \{\langle Zo_{t-1}, Zo_t \rangle : t \in T\} \tag{1}$$

$$Fo_t = Zo_t - Zo_{t-1} \tag{2}$$

where T is the length of input image sequences. First-order features of each joint are calculated by subtracting features of one frame from the previous frame. We use Zo_t minus Zo_{t-1} to get the first-order features (denote by Fo) as in Eq. 2.

Second-Order Features can be seen as the acceleration features. We construct a triplet subset for each frame's features:

$$\mathcal{U}_2 = \{\langle Zo_{t-1}, Zo_t, Zo_{t+1} \rangle : t \in T\} \tag{3}$$

$$So_t = (Zo_{t+1} - Zo_t) - (Zo_t - Zo_{t-1}) = (Fo_{t+1} - Fo_t) \tag{4}$$

Similar to the manner of getting first-order features, second-order features of each joint are calculated by subtracting features of current frame's first-order features from its previous frame's first-order features. We use Fo_{t+1} minus Fo_t to get the second-order features (So) by Eq. 4.

Slow-fast Feature Analysis: Slow and fast moving joints are both useful in gesture recognition. The features representing static tendency joints and motion tendency joints encode different levels of motion information. Instead of directly considering these motion features aggregately, we propose to explicitly learn these motion levels separately. Specifically, we design a slow-fast feature analysis method to explicitly distinguish these slow-moving and fast-moving joint features from First-Order Features Fo and Second-Order Features So. In this way, both static tendency joints and motion tendency joints can be exploited.

First-order features and second-order features tensors are of the shape $N \times C \times H \times W$ (the same as the zero-order ones). We first reshape these features to $N \times CHW$ matrices (where N is the number of hand joints), and then calculate the L_2 norm on each joint's first-order and second-order feature vector (with the shape of $1 \times CHW$) from the reshaped features matrices, respectively. There will be N L_2 norm results, denoted by Feature Difference ($FD = \{FD_i | i = 1, ...N\}$, a $N \times 1$ vector). Each FD_i is a value representing the motion magnitude of each corresponding joint. We adopt Gaussian distributions to obtain the feature maps of slow-moving and fast-moving joints. For slow motion analysis, we aim to enhance features from the *more static* joints, i.e., assign larger weights to joints that move more slowly. We use a Gaussian function (with FD_{min} as mean and $(FD_{max} - FD_{min})/3$ as standard deviation) to map FD values to weights (FD_{min}/FD_{max} denotes the min/max FD values). With this mapping, the weight of the joint with the min/max motion magnitude (FD_{min}/FD_{max}) will be close to 1/0. As there are N hand joints, we will obtain a $N \times 1$ *slow vector* that contains weights for the features of N joints. Similarly, we aim to enhance features from the *more dynamic* joints using the fast motion analysis module. We thus set FD_{max} and $(FD_{min} - FD_{max})/3$ as the mean and standard deviation. In this way, the joint that has min/max motion magnitude will have a weight around 0/1.

When the slow and fast motion analysis modules apply on the first-order and second-order features Fo and So, we obtain four $N \times 1$ vectors that contain weights of features of N joints as shown in Fig. 2: 1) First-order slow vector (f_{os}); 2) First-order fast vector (f_{of}); 3) Second-order slow vector (s_{os}); and 4) Second-order fast vector (s_{of}). All these four vectors are used to refine the zero-order features Zo which are first reshaped to an $N \times CHW$ matrix and then multiplied with these four vector separately. The embedding features are then reshaped back to $N \times C \times H \times W$ tensors, namely, first-order-slow features, first-order-fast features, second-order-slow features and second-order-fast features as shown in Fig. 2. These four features together with the zero-order features are fed to the multi-scale relation module (details to be discussed in the Sec. 3.3), respectively. Finally, the results of each stream are averaged to obtain the gesture-optimized joint-aware feature maps $\mathcal{G}$ and the gesture category y.

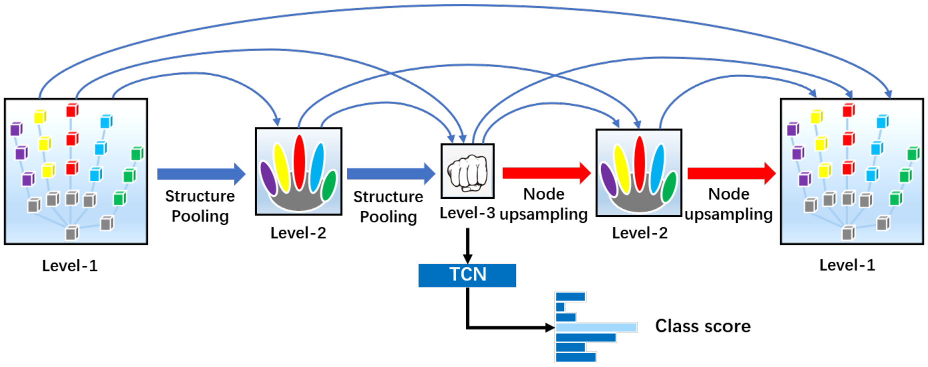

Fig. 3. Illustration of the multi-scale relation module: The multiple scale analysis process the feature maps from the slow-fast feature analysis at three different levels to generate relations at each level. It interact with the Gesture Sub-network by applying temporal convolution (TCN) on Level 3 (containing global information) to generate the classification scores. Node up-sampling is applied to keep the input and output of the same shape.

3.3 Multi-scale relation module

Considering different levels of semantic information contained in the hierarchical structure of hand, human hand can be defined with different scales. As shown in Fig. 3, we show three levels, where the level-1 is the local level consisting of the hand joints, and the level-2 is the middle level representing five fingers and palm. For the level-3, we see the hand globally as complete holistic information. Following the connection between contiguous scale, we use the structure pooling to perform feature aggregation across these three scales, and recognize gesture class y at the Level-3 using TCN, since it contains the global information.

Structure pooling means we use average pooling over the hand joints by following hierarchical physical structure of hand to perform step-wise feature aggregation. We first average features of the joints that belong to each finger or palm, in order to get features for the five fingers and palm (see Fig. 3), then average features of five fingers and palm to obtain the final global features representing full hand.

Additionally, we calculate a relation matrix for each level to better learn the features at each scale. Take the first level as the example; the whole feature maps size is $N \times C \times H \times W$. We first activate it through two embedding function ($1 \times 1 \times 1$ convolution). The two embedding features are rearranged and reshape to a $N \times CHW$ matrix and $CHW \times N$ matrix. They are then multiplied to obtain a $N \times N$ relation matrix. The values of the matrix mean the degree of relation between each pair of joints. The softmax function is used here to do the normalization. In this way, we can calculate relation matrices for each level and use them to refine the feature maps at each hand scale.

To maintain the input and output of this module in the same shape, we use the node up-sampling method: joints' features from the higher level are duplicated to the corresponding child joint in the lower level. In addition, the skipconnections (see thin blue arrows in Fig 3) are used over different spatial scales of hand to better learn multi-scale hand features and to preserve the original information. Our multi-scale network participates in each stream of multi-order multi-stream module (as shown in Fig. 2).

3.4 Weakly-Supervised Learning Strategy

Weakly-supervised 3D hand pose estimation using gesture labels: Annotating 3D poses is often laborious, and it's difficult to have a large amount of video samples with 3D pose annotations for training. In the supervised learning, the pose-optimized joint-aware feature maps $\mathcal{P}$ and the gesture-optimized joint-aware feature maps $\mathcal{G}$ are learned based on the joint-aware feature maps $\mathcal{J}$. We therefore propose a weakly-supervised learning method that use gesture labels as weak supervision for 3D hand pose estimation. We provide different ratios of training data with 3D pose annotations in training process.

Weakly-supervised gesture recognition using pose labels: When only a few videos have gesture labels, we can similarly use 3D hand pose annotations as weak supervision for gesture recognition. We provide different ratios of training data with gesture labels in training to make our method more applicable.

3.5 Training

We use the following losses in training. **2D Heatmaps loss.** $L_{2d} = \sum_{n=1}^{N} \|\mathcal{H}_n - \hat{\mathcal{H}_n}\|_2^2$, This loss measures the $L2$ distance between the predicted heatmaps $\mathcal{H}_n$ and the ground-truth heatmaps $\hat{\mathcal{H}_n}$. **Depth Regression loss.** $L_{3d} = \sum_{n=1}^{N} \|D_n - \hat{D_n}\|_2^2$, where D_n and $\hat{D_n}$ are the estimated and the ground truth depth values, respectively. L_{3d} is also based on the $L2$ distance. **Classification loss.** We use the standard categorical cross-entropy loss to supervise the gesture classification process, which is $L_c = CrossEntropy(y, \tilde{y})$, where y is the class predicted score and $\tilde{y}$ is the ground truth category.

Fully-Supervised training strategy. In our implementation, we first fine-tune the ResNet-50 to make it sensitive to human joint information. We then train the entire network in an end-to-end manner with the objective function:

$$L = \lambda_{2d} L_{2d} + \lambda_{3d} L_{3d} + \lambda_c L_c \tag{5}$$

Weakly-Supervised training strategy. Based on the Eq. 5, we set $\lambda_{2d} = 0$ and $\lambda_{3d} = 0$ when the samples do not have 3D pose annotations and we use gesture categories as weak supervision for 3D hand pose estimation. Similarly, we set $\lambda_c = 0$ for video sequences without gesture labels, where we use 3D pose annotations as weak supervision for gesture recognition.

4 Experiment

Implementation Details: We implement our method with the PyTorch framework, and optimize the objective function with the Adam optimizer with mini-batches of size 4. The learning rate starts from 10^{-4}, with a 10 times reduction when the loss is saturated. Following the same setting in [18, 37], the input image is resized to 256 × 256, and the heatmap resolution is set at 64 × 64. In the experiment, the parameters in the objective function are set as follows: $\lambda_{2d} = 1$, $\lambda_{3d} = 0.001$ and $\lambda_c = 0.001$. For the **weakly-supervised learning**, we choose the 15% to 40% samples as the weakly supervision samples and set $\lambda_{2d} = 0$ and $\lambda_{3d} = 0$ when the samples do not have 3D pose annotations (gesture categories are used as weak supervision for 3D hand pose estimation). Additionally, we set $\lambda_c = 0$ for video sequences without gesture labels, where 3D pose annotations are used as weak supervision for gesture recognition as described in in Sec. 3.4.

Following [34], each input video is divided into K segments and a short clip is randomly selected from each segment in training. On testing, each video is similarly divided into K segments and one frame is selected from each segment to make sure that temporal space between adjacent frames is equal to T/K. The final classification scores are computed by the average over all clips from each video, and the pose estimation is presented on image level.

Datasets: We perform extensive experiments on the large-scale and challenging dataset: First-Person Hand Action (FPHA) [10] for simultaneous gesture recognition and 3D hand pose estimation. To the best of our knowledge, this is the only publicly available dataset that provides labels of accurate 2D & 3D hand poses and gesture labels. The dataset consists of 1175 gesture videos with 45 gesture classes. The videos are performed by 6 actors under 3 different scenarios. A total of 105, 459 video frames are annotated with accurate hand pose and action classes. Both 2D and 3D annotations of the total 21 hand keypoints are provided for each frame. We follow the protocol in [10, 29] and use 600 video sequences for training and the remaining 575 video sequences for testing.

Evaluation Metrics: We adopt the widely used metrics for evaluation of gesture recognition and 3D hand pose estimation. For gesture recognition, we directly evaluate the accuracy of video classification. For 3D pose estimation, we use the percentage of correct keypoints (PCK) score that evaluates the pose estimation accuracy with different error thresholds.

4.1 Experimental Results

Gesture Recognition: Table 1 shows the comparison with state-of-the-art gesture recognition methods. It can be seen that our method outperforms the state-of-the-art by up to 3%, showing its effectiveness gesture recognition. Additionally, augmenting each of our proposed module (multi-scale relation, multi-order multi-stream and collaborative learning strategy) yield improved gesture recognition performance.

3D Hand Pose Estimation: We compare our method with prior works on FPHA as shown in the first graph in Fig. 4. Table 2 shows three 3D PCK results at three specific error threshold. It can be seen that our method outperforms the state-of-the-art with a large range between $0mm$ and $30mm$. Even though we use color images, our results are better than [10] that uses depth images which demonstrates the advantage of our proposed method.

Qualitative results on 3D Hand Pose Estimation: Fig. 5 illustrates 3D pose estimations by our method. We compare the ground truth 3D poses

Table 1. Comparisons to state-of-the-art gesture recognition methods: "Baseline" means 1-iteration network with no multi-order feature analysis and multi-scale relation.

Model	Input modality	Accuracy
Joule-depth [12]	Depth	60.17%
Novel View [27]	Depth	69.21%
HON4D [25]	Depth	70.61%
FPHA + LSTM[10]	Depth	72.06%
Two-stream-color [28]	Color	61.56%
Joule-color [12]	Color	66.78%
Two-stream-flow [28]	Color	69.91%
Two-stream-all [28]	Color	75.30%
[29] - HP	Color	62.54%
[29] - HP + AC	Color	74.20%
[29] - HP + AC + OC	Color	82.43%
Baseline	Color	72.17%
Baseline + multi-scale	Color	78.26%
Baseline + multi-scale + multi-order	Color	83.83%
Baseline + multi-scale + multi-order + 2-iterations	Color	85.22%

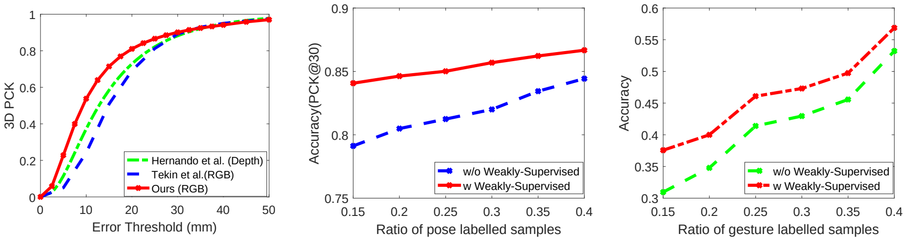

Fig. 4. Left: Comparing our method with [10] and [29] for 3D hand pose estimation with 3D PCK metric. **Middle:** Comparing our weakly supervised method with the baseline (with 3D PCK@30) when different amounts of pose labels are used. **Right:** Comparing our weakly supervised method with the baseline (with classification accuracy) when different amount of gesture labels are used.

(in blue-color structures) and the predicted 3D pose (in red-color structures) in the same 3D coordinate system. We also provide the predicted 2D poses in the original RGB image. As Fig. 5 shows, our method is capable of accurately predicting 3D poses of different orientations with different backgrounds.

4.2 Weakly-supervised Learning

Weakly-supervised results on 3D hand pose estimation: We present multiple experiments on our weakly-supervised method by providing different ratios (15% to 40%) of samples with pose labels (gesture labels are provided for all training samples) and compare with the baseline that does not use gesture labels. Fig. 4 (middle) shows 3D PCK@30 (percentage of correct keypoint when error threshold smaller than 30mm) results of the baseline and our weakly-supervised method. It can be seen that the 3D hand pose estimation is improved significantly for all labeled ratios when weak supervision is included. This validates that joint-aware features in the gesture can benefit 3D hand pose estimation.

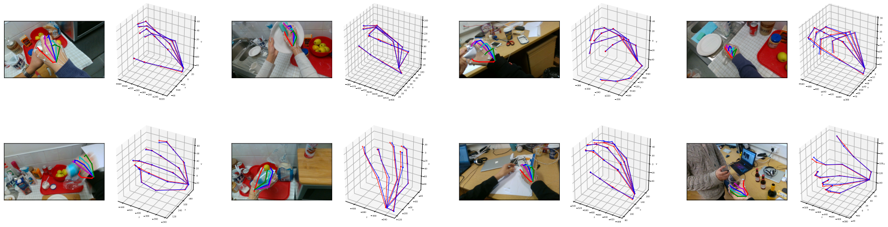

Fig. 5. Qualitative illustration of our proposed method: It shows the predicted 2D poses shown on the original image. It also compares the predicted 3D poses (the blue-color structures) and the Ground Truth 3D poses (the red-color structures).

Table 2. Comparisons on 3D pose estimation: Numbers are percentage of correct keypoint (PCK) over respective error threshold, more results available in Fig. 4 (left). Our results are based on the proposed 2-iterations multi-order structure.

Error Threshold(mm)	PCK@20	PCK@25	PCK@30
Hernando (Depth)[10]	72.13%	82.08%	87.87%
Tekin (RGB)[29]	69.17%	81.25%	89.17%
Ours (RGB)	81.03%	86.61%	90.11%

Weakly-supervised results on gesture recognition: We compare our weakly supervised method that uses pose labels as weak supervision for gesture recognition with the baseline which does not use pose labels. We conduct experiments by providing different ratios of training samples with gesture labels, while the pose labels of all samples are given. As Fig. 4 (right) shows, our weakly-supervised learning improves the gesture recognition significantly for all labeled ratios. This validates that joint-aware features in hand poses can improve the gesture recognition performance greatly.

4.3 Ablation Studies

Impact of number of network iterations: Table 3 shows the 3D PCK results and classification results of our method under different iterations of collaborative learning. It can be seen that our method improves with increasing iterations. This can be expected since hand pose estimation and gesture recognition learn in a collaborative manner and boost each other. Note that the improvement of 3D PCK and gesture recognition slows down with the increase of iterations. We use the two-iteration network in the experiment for the balance between accuracy and computational complexity. Note all these comparisons are based on the zero-order framework. We cannot evaluate multi-order network for the 3-itr, 4-itr, 5-itr due to our GPU's memory limitation.

Effect of the multi-order module: We analyze the advantage of our proposed multi-order module by implementing four variants as shown in Table 4 (part 1, 2, and 4). It can be seen that adding first-order and second-order slow-fast features leads to an accuracy improvement by 1.7% and 2.9%, respectively. Our multi-order module (Zero-order + First and Second order slow-fast) achieves the best accuracy at 85.22%, demonstrating its effectiveness.

Table 3. Evaluation of our proposed network on gesture recognition and pose estimation with respect to different iteration numbers.

Iteration (itr) number	1-itr	2-itr	3-itr	4-itr	5-itr
Pose estimation (PCK@30)	87.2%	89.3%	89.8%	89.9%	**89.9%**
Gesture recognition accuracy	78.3%	80.9%	81.7%	81.9%	**82.0%**

Table 4. Evaluation of our proposed gesture recognition network with different combinations of motion features of different orders and slow-fast patterns. (All experiments below are based on the 2-iteration network.)

	Network setting	Accuracy	Δ
1	Zero-order	80.87%	
2	Zero-order + First-order slow-fast	82.61%	1.74%
	Zero-order + Second-order slow-fast	83.80%	2.93%
3	Zero-order + First and Second order slow	82.96%	2.09%
	Zero-order + First and Second order fast	82.09%	1.22%
4	Zero-order + First and Second order slow-fast	85.22%	4.35%

Effect of the slow feature and fast feature: We also evaluate the impact of the slow-fast features and Table 4 (part 3) shows the results. It can be seen that the slow features and the fast features can improve the accuracy by 2.1% and 1.2%, respectively, and the best accuracy is obtain when both are included.

Effect of the multi-scale relation: We also assess the effectiveness of the our multi-scale relation module and Table 1 shows experimental results. As Table 1 shows, removing the multi-scale relation module leads to around 6% accuracy drop as compared with the "Baseline" and "Baseline + multi-scale", showing the benefit of the proposed multi-scale relation.

5 Conclusion

In this paper, we have presented a collaborative learning method for joint gesture recognition and 3D hand pose estimation. Our model learns in a collaborative way to recurrently exploit the joint-aware feature to progressively boost the performance of each task. We have developed a multi-order multi-stream model to learn motion information in the intermediate feature maps and designed a multi-scale relation module to extract semantic information at hierarchical hand structure. To learn our model in scenarios that lack labeled data, we leverage one fully-labeled task's annotations as weak supervision for the other very few labeled task. The proposed collaborative learning network achieves state-of-the-art performance for both gesture recognition and 3D hand pose estimation tasks.

Acknowledgement

The research was carried out at the Rapid-Rich Object Search (ROSE) Lab, Nanyang Technological University, Singapore. This research work was partially supported by SUTD projects PIE-SGP-Al-2020-02 and SRG-ISTD-2020-153.

References

1. Abavisani, M., Joze, H.R.V., Patel, V.M.: Improving the performance of unimodal dynamic hand-gesture recognition with multimodal training. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2019)
2. Boukhayma, A., Bem, R.d., Torr, P.H.: 3d hand shape and pose from images in the wild. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 10843–10852 (2019)
3. Cai, Y., Ge, L., Cai, J., Magnenat-Thalmann, N., Yuan, J.: 3d hand pose estimation using synthetic data and weakly labeled rgb images. IEEE Transactions on Pattern Analysis and Machine Intelligence (2020)
4. Cai, Y., Ge, L., Cai, J., Yuan, J.: Weakly-supervised 3d hand pose estimation from monocular rgb images. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 666–682 (2018)
5. Cai, Y., Ge, L., Liu, J., Cai, J., Cham, T.J., Yuan, J., Thalmann, N.M.: Exploiting spatial-temporal relationships for 3d pose estimation via graph convolutional networks. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2272–2281 (2019)
6. Cai, Y., Huang, L., et al.: Learning progressive joint propagation for human motion predictionn. In: Proceedings of the European Conference on Computer Vision (ECCV) (2020)
7. Carreira, J., Zisserman, A.: Quo vadis, action recognition? a new model and the kinetics dataset. In: proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 6299–6308 (2017)
8. Chen, X., Lin, K.Y., Liu, W., Qian, C., Lin, L.: Weakly-supervised discovery of geometry-aware representation for 3d human pose estimation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 10895–10904 (2019)
9. De Smedt, Q., Wannous, H., Vandeborre, J.P.: Skeleton-based dynamic hand gesture recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. pp. 1–9 (2016)
10. Garcia-Hernando, G., Yuan, S., Baek, S., Kim, T.K.: First-person hand action benchmark with rgb-d videos and 3d hand pose annotations. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2018)
11. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
12. Hu, J.F., Zheng, W.S., Lai, J., Zhang, J.: Jointly learning heterogeneous features for rgb-d activity recognition. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2015)
13. Iqbal, U., Garbade, M., Gall, J.: Pose for action-action for pose. In: 2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017). pp. 438–445. IEEE (2017)
14. Klaser, A., Marszałek, M., Schmid, C.: A spatio-temporal descriptor based on 3d-gradients (2008)
15. Laptev, I.: On space-time interest points. International journal of computer vision **64**(2-3), 107–123 (2005)
16. Liu, J., Shahroudy, A., Xu, D., Kot, A.C., Wang, G.: Skeleton-based action recognition using spatio-temporal lstm network with trust gates. IEEE Transactions on Pattern Analysis and Machine Intelligence **40**(12), 3007–3021 (2018)

17. Liu, J., Wang, G., Duan, L., Abdiyeva, K., Kot, A.C.: Skeleton-based human action recognition with global context-aware attention lstm networks. IEEE Transactions on Image Processing **27**(4), 1586–1599 (2018)
18. Liu, J., Ding, H., Shahroudy, A., Duan, L.Y., Jiang, X., Wang, G., Chichung, A.K.: Feature boosting network for 3d pose estimation. IEEE transactions on pattern analysis and machine intelligence (2019)
19. Liu, J., Shahroudy, A., Xu, D., Wang, G.: Spatio-temporal lstm with trust gates for 3d human action recognition. In: European Conference on Computer Vision. pp. 816–833. Springer (2016)
20. Liu, J., Wang, G., Hu, P., Duan, L.Y., Kot, A.C.: Global context-aware attention lstm networks for 3d action recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1647–1656 (2017)
21. Liu, M., Yuan, J.: Recognizing human actions as the evolution of pose estimation maps. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2018)
22. Luvizon, D.C., Picard, D., Tabia, H.: 2d/3d pose estimation and action recognition using multitask deep learning. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2018)
23. Mueller, F., Bernard, F., Sotnychenko, O., Mehta, D., Sridhar, S., Casas, D., Theobalt, C.: Ganerated hands for real-time 3d hand tracking from monocular rgb. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 49–59 (2018)
24. Nguyen, X.S., Brun, L., Lezoray, O., Bougleux, S.: A neural network based on spd manifold learning for skeleton-based hand gesture recognition. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2019)
25. Oreifej, O., Liu, Z.: Hon4d: Histogram of oriented 4d normals for activity recognition from depth sequences. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2013)
26. Rad, M., Oberweger, M., Lepetit, V.: Domain transfer for 3d pose estimation from color images without manual annotations. In: Asian Conference on Computer Vision. pp. 69–84. Springer (2018)
27. Rahmani, H., Mian, A.: 3d action recognition from novel viewpoints. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2016)
28. Simonyan, K., Zisserman, A.: Two-stream convolutional networks for action recognition in videos. In: Advances in neural information processing systems. pp. 568–576 (2014)
29. Tekin, B., Bogo, F., Pollefeys, M.: H+o: Unified egocentric recognition of 3d hand-object poses and interactions. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2019)
30. Tran, D., Wang, H., Torresani, L., Ray, J., LeCun, Y., Paluri, M.: A closer look at spatiotemporal convolutions for action recognition. In: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition. pp. 6450–6459 (2018)
31. Tu, Z., Xie, W., Qin, Q., Poppe, R., Veltkamp, R.C., Li, B., Yuan, J.: Multi-stream cnn: Learning representations based on human-related regions for action recognition. Pattern Recognition **79**, 32–43 (2018)
32. Wang, H., Kläser, A., Schmid, C., Liu, C.L.: Dense trajectories and motion boundary descriptors for action recognition. International journal of computer vision **103**(1), 60–79 (2013)
33. Wang, H., Schmid, C.: Action recognition with improved trajectories. In: Proceedings of the IEEE international conference on computer vision. pp. 3551–3558 (2013)

34. Wang, L., Xiong, Y., Wang, Z., Qiao, Y., Lin, D., Tang, X., Van Gool, L.: Temporal segment networks: Towards good practices for deep action recognition. In: European conference on computer vision. pp. 20–36. Springer (2016)
35. Xiaohan Nie, B., Xiong, C., Zhu, S.C.: Joint action recognition and pose estimation from video. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1293–1301 (2015)
36. Xie, S., Sun, C., Huang, J., Tu, Z., Murphy, K.: Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 305–321 (2018)
37. Zhou, X., Huang, Q., Sun, X., Xue, X., Wei, Y.: Towards 3d human pose estimation in the wild: A weakly-supervised approach. In: The IEEE International Conference on Computer Vision (ICCV) (Oct 2017)
38. Zhu, H., Vial, R., Lu, S.: Tornado: A spatio-temporal convolutional regression network for video action proposal. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 5813–5821 (2017)
39. Zhu, H., Vial, R., Lu, S., Peng, X., Fu, H., Tian, Y., Cao, X.: Yotube: Searching action proposal via recurrent and static regression networks. IEEE Transactions on Image Processing **27**(6), 2609–2622 (2018)
40. Zimmermann, C., Brox, T.: Learning to estimate 3d hand pose from single rgb images. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 4903–4911 (2017)