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

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

In this supplementary document, we will provide materials not included in the main paper due to space constraints. Firstly, Section 1 provides more detail of our proposed network structures. Next, Section 2 elaborates descriptions of Figure 2 in the main paper, and more details of our design choices for each component. Finally, Section 3 analyzes the abilities of our proposed multi-order module for capturing slow-fast motion.

1 Network Architecture

Figure 1 illustrates the iteration procedure of our proposed collaborative learning strategy. We show the framework with two iterations, and more pairs of Pose Sub-Network and Gesture Sub-Network can be stacked with the increase of iterations. Additionally, some details of Pose Sub-network and Gesture Sub-network are shown here for better understanding the whole framework. (More details of Gesture Sub-network can also be found in Figure 2 and 3 of the main paper.)

2 More details of Figure 2 in the main paper

The Gesture Subnet in Fig. 1 is a concise version of Figure 2 in the main paper. The operation flow of Figure 2 can be described as follow:

1) First, the Zero-Order Features are fed to the multi-order multi-stream feature analysis module as input. Then we use the Zero-Order Features of adjacent frames to calculate the First-Order Features (Velocity Features) and Second-Order Features (Acceleration Features). We then apply Slow-Fast Feature Analysis on these features, which extracts the information of both slowly-moving and

⋆ Corresponding author.

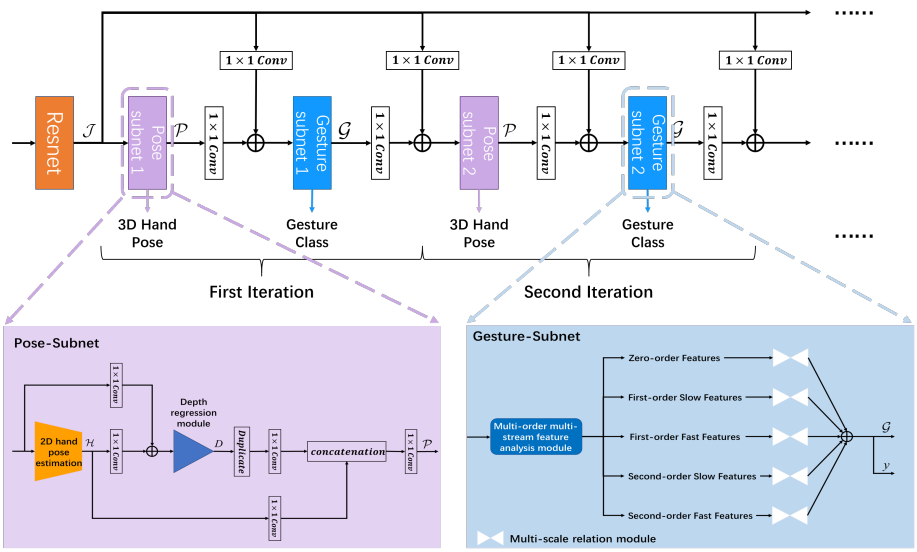

Fig. 1. Illustration of our proposed collaborative learning strategy and details of the Pose Sub-network and Gesture Sub-network. Here $\mathcal{J}$ is the joint-aware feature maps. $\mathcal{P}$ is the pose-optimized joint-aware feature maps. $\mathcal{G}$ is the gesture-optimized joint-aware feature maps. y is the predicted gesture class. $\mathcal{H}$ is the 2D Heatmaps. D is the Depth values. 3D positions can be obtained by combining the 2D Heatmaps $\mathcal{H}$ and Depth values D.

fast-moving joints. Thus we can achieve five sets of features, namely, Zero-Order Features, First-Order Slow Features, First-Order Fast Features, Second-Order Slow Features and Second-Order Fast Features. These five extracted features represent different orders of discriminative motion of the hand joints, respectively.

2) The obtained five sets of features are then fed to the multi-scale relation modules to capture multi-level semantic information about the hierarchical structure of hands (shown in Figure 3 in the main paper). Specifically, we feed these five feature sets to five independent multi-scale relation modules to learn hierarchical structure information at different motion magnitudes.

3) Finally, the results of the five streams that contain discriminative and rich motion and structure information are averaged to get the gesture-optimized joint-aware features $\mathcal{G}$.

3 More evaluation for Slow-Fast Motion Analysis module

We provide some visualization results to analyze the abilities of our proposed slow-fast feature analysis module for capturing slow-fast motion, as shown in Fig. 2. We extract the fast and slow motion magnitude values from the first-order

fast/slow vector, showing in the left and right column of Fig. 2, respectively. Blue and red lines in Fig. 2 represent moving for hand joints, and the length of these lines stands for the moving distance. The blue lines represent the motion distance of fastest-moving joints, while red lines stand for slowest moving joints. Following [1], in testing, each video is divided into K segments. One frame is selected from each segment to make sure that temporal space between adjacent frames is equal to T/K. Each joint's moving distance is calculated by corresponding joint locations difference between selected adjacent frames, instead of the adjacent frames in RGB sequences. To make each sub-figure clear, we only present motion lines of the top-5 fastest-moving joints and top-5 slowest moving joints. The fast and slow magnitude values of these ten joints are shown around them in the left column and right column of Fig 2, respectively.

As shown in the left column of Fig. 2, values of blue lines are much larger than that of red lines. Additionally, the largest line's (fastest moving joint) value is 1, and all red lines (top-5 slowest moving joint) corresponding values are closed to 0. Similarly, in the right column of Fig. 2, the relations become opposite. It can be seen that fast-moving joints can have much larger fast-vector weight and the joints with slow motion will have a larger slow-vector weight, validating that our proposed slow-fast feature analysis module can distinguish slow-fast motion joints.

References

1. 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)

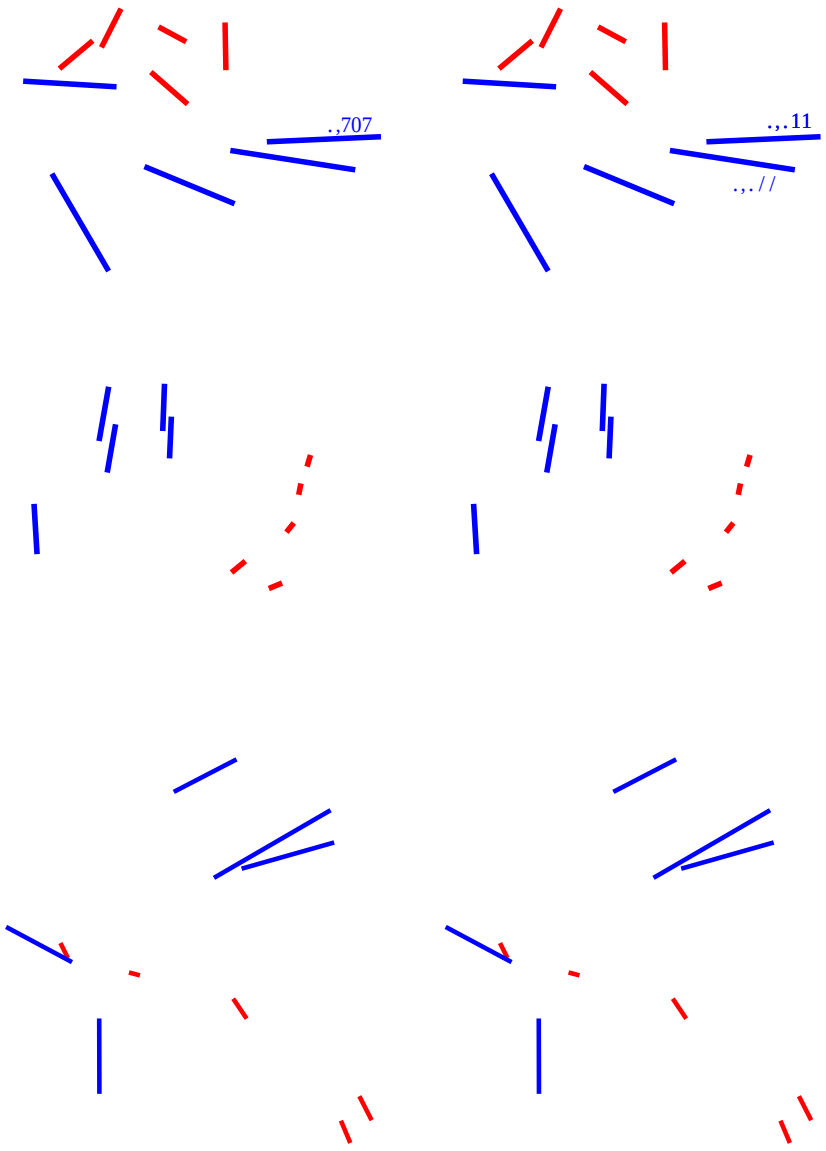

Fig. 2. Left: Fast-Vector weight for fast and slowly-moving joints. Right: Slow-Vector weight for fast and slowly-moving joints.