HopFIR: Hop-wise GraphFormer with Intragroup Joint Refinement for 3D Human Pose Estimation Supplementary Material

In this supplementary material, we provide additional details and ablation studies that were not included in the main manuscript due to space constraints. Section 1 introduces more ablation experiments to investigate the effectiveness of the HA layer. Section 2 provides more details about the structure of our proposed HG module. Section 3 shows intuitive descriptions for k-hop. Section 4 visualizes the k-hop attention weight matrices and intragroup attention weight matrices. Section 5 presents additional qualitative results for comparison.

1. Effectiveness of the HA Layer

1.1. Comparison of Grouping the k-hop Joints

In section 4.3 of the main manuscript, we demonstrate how the HopFIR architecture can be used without the human body prior by using a random graph instead of the skeleton graph. We constructed three random graphs based on the number of edges in the human skeleton graph to comprehensively explore the effectiveness of grouping k-hop joints. These graphs were comprised of 15 edges, 30 edges, and all nodes connected. Due to the maximum number of edges corresponding to the fully connected graph and its specificity in GCN, we randomly generate fully connected graphs with varying edge weights. Notably, we explored up to three hops, consistent with the approach taken for the skeleton graph. For the fully connected graph, we generate three fully connected graphs to represent its three hops.

As shown in Table 1, the performance improves as the number of edges of the random graph increases, but remains inferior to that based on the human skeleton graph. This is because as the number of edges in a random graph increases, the number of possible combinations also increases. Therefore, each node can aggregate more information during the feature updating process. However, the lack of human body prior restricts the discovery of human joint synergies. Nevertheless, grouping the joints by k-hop neighborhood is more effective than using self-attention between nodes or learning nonlinear mappings for each node.

	MLP	Transformer	RG(15)	RG(30)	RG(Max)	Skeleton
MPJPE	36.59	36.21	35.39	34.85	34.68	**32.67**
P-MPJPE	29.21	27.77	28.80	27.89	27.94	**26.20**

Table 1. Quantitative comparison of HopFIR on random graphs. RG denotes random graph, and the elements in parentheses denote the number of edges in the respective random graph, where Max signifies the fully connected graph. MLP and Transformer refer to replacing the HA layer in HopFIR with the corresponding modules.

Method	Channels	Params	MPJPE	P-MPJPE
SemGCN [3]	128	0.27M	42.14	33.53
SemGCN + HA(HSS)	128	0.49M	**38.41**	**30.56**
SemGCN + HA(SSS)	128	0.49M	41.30	33.07
SemGCN + HA(SHH)	128	0.49M	38.81	31.05
SemGCN + MLP	128	0.57M	42.97	33.87
SemGCN + Transformer	128	0.86M	43.78	34.87
SemGCN [3] w/ Non-local [2]	128	0.43M	40.78	31.46
SemGCN w/ Non-local +HA(HSS)	128	0.66M	38.03	30.50
SemGCN w/ Non-local +HA(SSS)	128	0.66M	**37.75**	30.17
SemGCN w/ Non-local +HA(SHH)	128	0.66M	37.94	**29.71**
SemGCN w/ Non-local + MLP	128	0.73M	39.36	30.93
SemGCN w/ Non-local + Transformer	128	1.03M	43.86	34.62
Modulated GCN [4]	128	0.29M	38.25	30.06
Modulated GCN +HA(HSS)+W	128	0.96M	36.54	29.09
Modulated GCN +HA(SSS)+W	128	0.96M	**36.14**	**29.02**
Modulated GCN +HA(SHH)+W	128	0.96M	37.38	30.02
Modulated GCN + MLP	128	1.03M	39.12	30.93
Modulated GCN + Transformer	128	1.03M	39.22	29.33

Table 2. Comparison of performance changes for various methods upon adding MLP or Transformer with a relatively larger set of parameters compared to adding HA Layer.

1.2. Performance Comparison of the HA Layer

We provide a performance comparison of various methods added with MLP or Transformer in Table 2. The results indicate that the HA layer is more effective in exploring the correlation between the node and k-hop group compared to intra-node self-attention or non-linear mapping of individual nodes.

2. HG Module

The HG module is a variant of HGF designed for the output layer, which structure is shown in Fig. 1. In the HG module, the first FC layer maps the feature channel to the final output channel, but with a size of 3. This dimension is too small for a multi-head self-attention mechanism. Con-

sequently, we have opted to remove the HA layer and employ HopGCN for information aggregation.

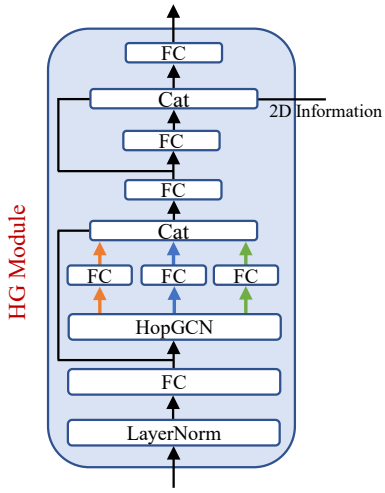


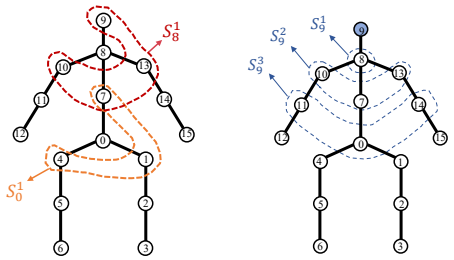
Figure 1. The details of the HG module. Arrows of different colors represent different hops.

3. Intuitive descriptions for k-hop

Fig. 2 shows intuitive descriptions for k-hop. (a) shows the 1-hops of joints 0 and 8. (b) shows the 3 hops (S_9^1, S_9^2, S_9^3) of joint 9.

4. Attention Weight Matrix

We visualize k-hop attention weight matrices in the HGF modules in Fig. 3 and self-attention weight matrices for the global skeleton graph and limb groups in Fig. 4. Three actions from the human3.6M [1], namely Discussion, Phoning, and Purchases, were randomly selected for analysis. In Fig. 3, one can see that each joint attends different groups in different actions. In Fig. 4, one can see that the peripheral joints in the MHSA layer in the IJR module do not pay attention to the other intragroup joints, whereas the MHSA layers within each group can attend to them. The results indicate that HopFIR can effectively aggregate peripheral joint information, and that peripheral joint representations can extract useful information from their associated limbs. Moreover, the IJR module promotes the HGF module to discover the latent synergies among joints, which is manifested in two ways: (1) IJR modules improve the performance of HGF which decreases the MPJPE to 32.67 mm and (2) IJR modules accelerate the learning process of HGF modules, *i.e.*, HopFIR networks without IJR modules need 38 epochs to converge to 35.30 mm and converge to 35.19 mm at the 57th epoch.



(a) 1-hops (b) 3-hops

Figure 2. Intuitive descriptions for k-hop. S_i^k denote the k-hop group of the joint i.

5. Qualitative Experiments

Fig. 5 presents additional visualization results obtained by HopFIR on the Human3.6M. In addition, we provide some qualitative results on wild images in Fig. 6. Despite being trained on the Human3.6M dataset, HopFIR is capable of achieving satisfactory results in unseen scenes.

References

[1] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3. 6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE transactions on pattern analysis and machine intelligence*, 36(7):1325–1339, 2013. 2, 4

[2] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7794–7803, 2018. 1

[3] Long Zhao, Xi Peng, Yu Tian, Mubbasir Kapadia, and Dimitris N Metaxas. Semantic graph convolutional networks for 3d human pose regression. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3425–3435, 2019. 1

[4] Zhiming Zou and Wei Tang. Modulated graph convolutional network for 3d human pose estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11477–11487, 2021. 1, 4

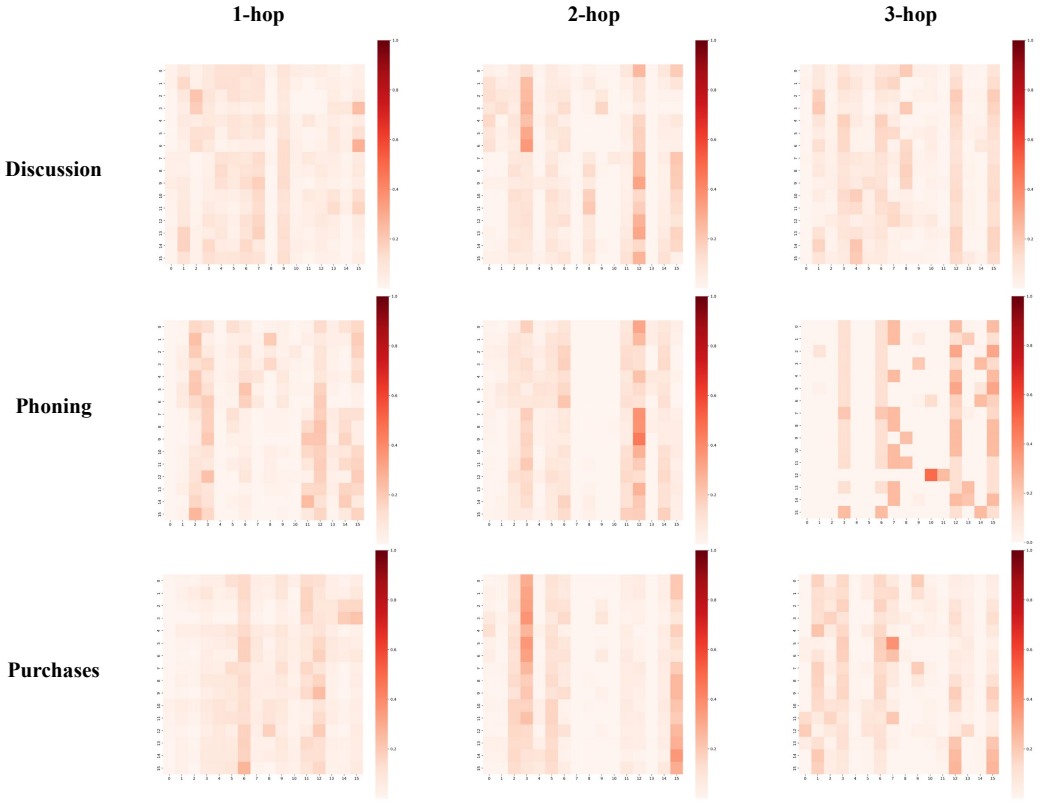


Figure 3. Visualization of k-hop attention weight matrix.

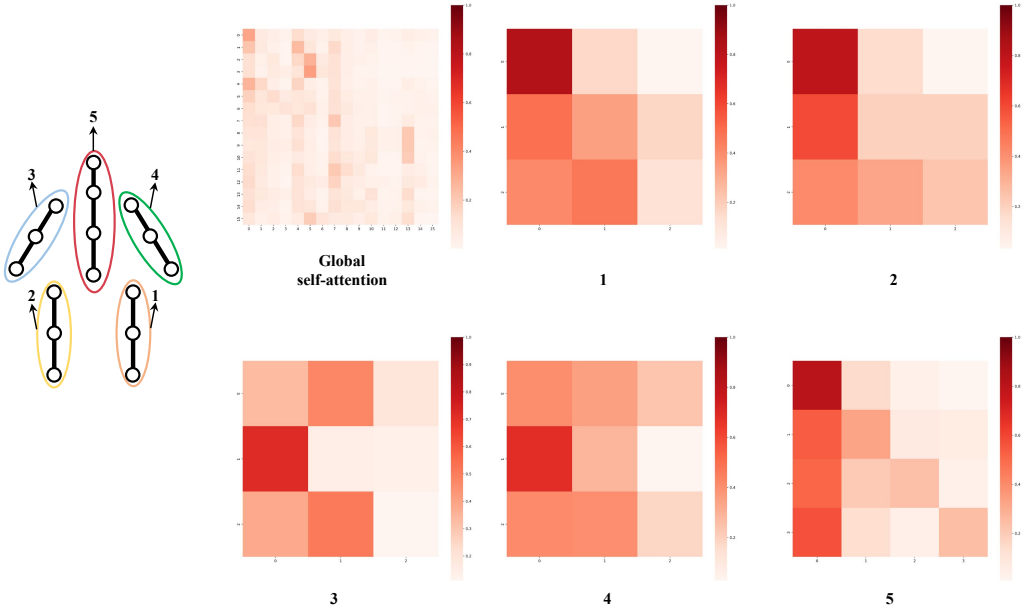


Figure 4. Visualization of the intragroup self-attention weight matrix.

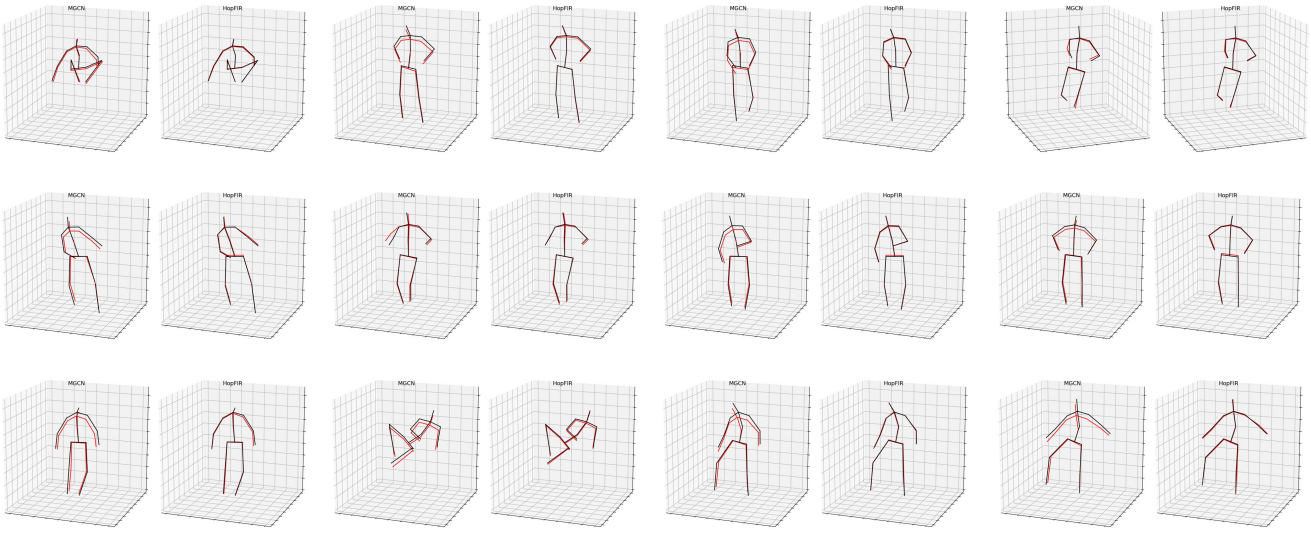


Figure 5. More qualitative visual results of our method and MGCN [4] on Human3.6M dataset [1]. The black lines are Ground Truth(GT) and the red lines are predictions by HopFIR and MGCN.

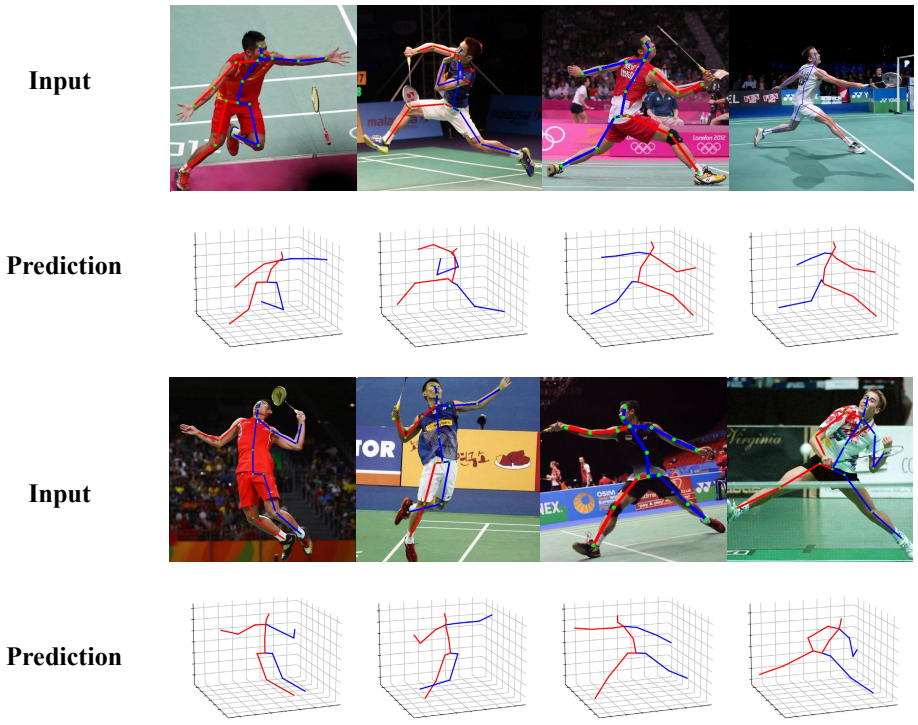


Figure 6. Visualization results of the HopFIR on in-the-wild images.