Ad-HGformer: An Adaptive HyperGraph Transformer for Skeletal Action Recognition -:Supplementary Material:-

Anonymous ECCV 2024 Submission

Paper ID #10985

We provide our git repository https://github.com/eccvanonymous/Ad-HGFormer to reproduce our findings. In this, we share our model file and will update other utility, supporting, and pre-train model files after the acceptance of the paper.

This supplementary material provides additional information about hypergraph convolution, extensive model analysis, comprehensive ablation study, and additional experimental results in sec. 1, 2, 3, and 4, respectively.

1 Hypergraph Convolution

Hypergraph: An undirected weighted graph $G = (V, E, W)$ is a structural representative of non-euclidean data points in terms of nodes (V) and associated edges (E) along with their weights (W). Each edge in G can only epitomize two nodes. The hypergraph $\mathcal{H} = (\mathcal{V}, \mathcal{E}, \mathcal{W})$ gives an excellent solution to achieve the semantic relation between more than two nodes. Here, $\mathcal{V}$ represents the nodes, same as in G, $\mathcal{E}$, and $\mathcal{W}$ denote the set of hyperedges and their corresponding weights, respectively. The hypergraph is represented by an incidence matrix $\mathbf{H} \in \{0,1\}^{|\mathcal{V}|\times|\mathcal{E}|}$

$$\mathbf{H}_{i,j} = \begin{cases} 1 & \text{if node } i \text{ belongs to hyperedge } j \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

A diagonal matrix $\mathbf{H_w} \in \mathbb{R}^{|\mathcal{E}|\times|\mathcal{E}|}$ is used to store the weights of hyperedges. The degree of a node $d_v(.)$ is the sum of the weights of all hyperedges associated with that node v, and the degree of a hyperedge $d_e(.)$ is the number of nodes contained in hyperedge e. $\mathbf{D_v}$ and $\mathbf{D_e}$ are diagonal degree matrices of nodes and hyperedges, respectively.

Hypergraph Convolution: A hypergraph convolution is defined as

$$\mathbf{Y_h} = \mathbf{D_v}^{-1/2}\mathbf{H}\mathbf{H_w}\mathbf{D_e}^{-1}\mathbf{H^T}\mathbf{D_v}^{-1/2}\mathbf{X_1}\theta \tag{2}$$

where $\mathbf{X_1} \in \mathbb{R}^{|\mathcal{V}|\times C_1}$ is the input node feature matrix and $\theta \in \mathbb{R}^{C_1\times C_2}$ is the convolutional filter on hypergraph. We use the notation $HGConv(X, \mathbf{H^n}, \mathbf{H_w^n})$ to represent hypergraph convolution on the input feature X.

2 Extensive Model Analysis

2.1 Temporal attention in STA-HT block

In the STA-HT block, temporal attention is applied to the hyperedge features H_f. Attention weights for different timestamps are calculated and multiplied

with H_f. We adopt a squeeze-excitation block to calculate the attention weights as in [4]. A linear layer squeezes the number of temporal dimensions with ReLU activation, and another linear layer excites the temporal dimension to bring it back to its original count with sigmoid activation. These weights are multiplied to residual hyperedge features H_f to get attentive features.

2.2 Model complexity

The tradeoff between computational complexity and model performance (accuracy in %) is illustrated in Table 1. Our model performs better with fewer parameters than CTR-GCN, Info-GCN, HD-GCN, and DST-CN. Although the proposed Ad-HGformer architecture needs more parameters than ST-GCN and Shift-GCN but outperforms the above two by a large margin in terms of flops and accuracies. The above ablation is analyzed on the NTU RGB+D 60 dataset.

Table 1: Performance analysis of Ad-HGformer based on model complexity.

Models	Publication	Params (M)	Flops (G)	X-Sub/X-View (%)
ST-GCN [8]	AAAI-2018	3.08	16.32	81.5/88.3
Shift-GCN [2]	CVPR-2020	2.76	10.01	90.7/96.5
CTR-GCN [1]	ICCV-2021	5.84	7.88	92.4/96.8
Info-GCN [3]	ICCV-2022	6.28	6.72	92.7/96.9
HD-GCN [5]	ICCV-2023	6.72	6.40	93.0/97.0
DST-HCN [7]	ICME-2023	3.50	2.93	92.3/96.8
Hyperformer [2]	arXiv-2023	2.60	14.8	92.9/96.5
Ad-HGformer	**Proposed**	**3.20**	**15.4**	**93.5/97.3**

3 Comprehensive Ablation Study

We evaluate our model using different values of related hyperparameters to get the best performance. These hyperparameters are the number of channels in transformer layers (c), the number of transformer blocks (L), and the number of hyperedges (k). Fig. 1 shows the plots for model performance (accuracy in %) vs number of epochs for the NTU RGB+D 60 dataset in the X-sub setting. As shown in this figure, we obtained the best results for $c = 216$, $L = 10$, and $k = 5$. We also generate the t-SNE plots of the above hyperparameters taking the same corresponding values as shown in Fig. 2 to consolidate our ablation. The best separation of different classes is visually evident for 10 transformer blocks with channel count 216 and 5 hyperedges.

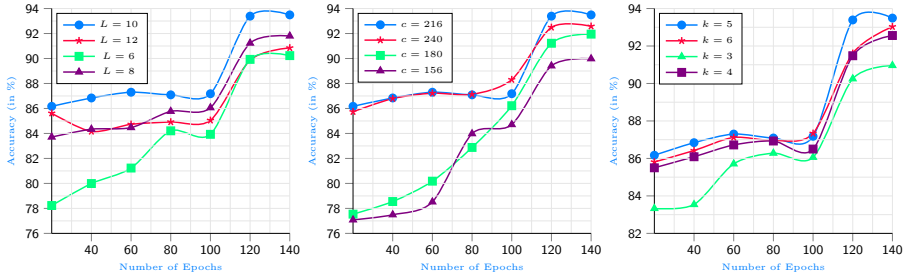

Fig. 1: Epoch-wise performance (accuracy in %) comparison of the proposed Ad-HGformer for **Left:** transformer block counts (L), **Middle:** transformer channel counts (c), **Right:** hyperedge count (k). [on NTU RGB+D 60(X-sub)]

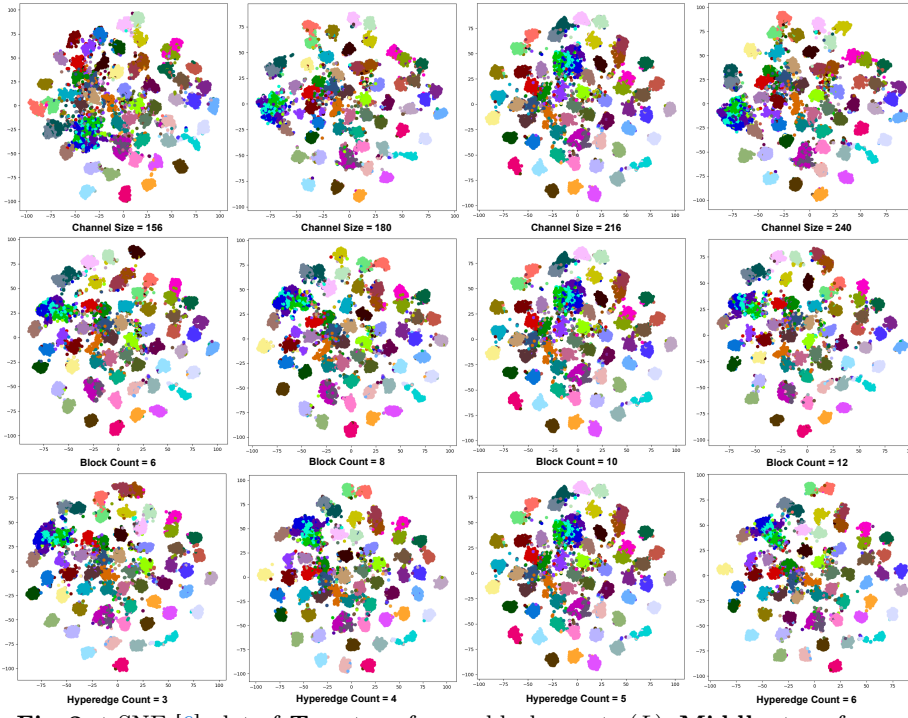

Fig. 2: t-SNE [6] plot of **Top:** transformer block counts (L), **Middle:** transformer channel counts (c), **Bottom:** hyperedge count (k). [on NTU RGB+D 60(X-sub)]

4 Additional Experimental Results

We observe the semantics of some action sequences in NTU RGB+D 60 datasets are very similar and often misclassified by the present state-of-the-art models. For example: (a) "writing[A12]" as "type on a keyboard[A30]" and vice-versa, (b)"reading[A11]" as "writing[A12]" and vice-versa, (c) "reading[A11]" as "play with the phone/tablet[A29]" and vice-versa, and (d) "writing[A12]" as "play with the phone/tablet[A29]" and vice-versa. We also recognize that using the proposed adaptive hyperedge decoder helps competently reduce the misclassifi-

cation of the above action sets. Fig. 3 validates the significance of the adaptive hyperedge decoder in the proposed Ad-HGformer.

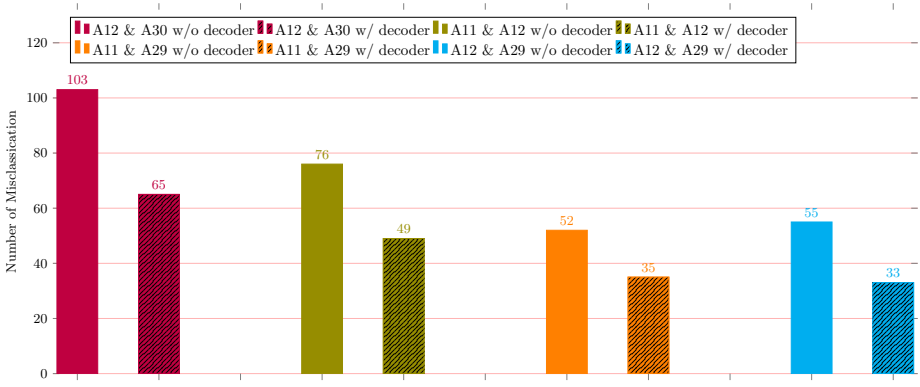

Fig. 3: Misclassification betweeen various ambiguous actions (Axx & Axx) of NTU RGB+D 60 dataset before and after implementing adaptive decoder. A11:reading, A12:writing, A29:play with the phone/tablet, A30:type on a keyboard.

References

1. Chen, Y., Zhang, Z., Yuan, C., Li, B., Deng, Y., Hu, W.: Channel-wise topology refinement graph convolution for skeleton-based action recognition. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 13359–13368 (2021) 2
2. Cheng, K., Zhang, Y., He, X., Chen, W., Cheng, J., Lu, H.: Skeleton-based action recognition with shift graph convolutional network. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 183–192 (2020) 2
3. Chi, H.g., Ha, M.H., Chi, S., Lee, S.W., Huang, Q., Ramani, K.: Infogcn: Representation learning for human skeleton-based action recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 20186–20196 (2022) 2
4. Hu, J., Shen, L., Albanie, S., Sun, G., Wu, E.: Squeeze-and-excitation networks (2019) 2
5. Lee, J., Lee, M., Lee, D., Lee, S.: Hierarchically decomposed graph convolutional networks for skeleton-based action recognition. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 10444–10453 (2023) 2
6. Van der Maaten, L., Hinton, G.: Visualizing data using t-sne. Journal of machine learning research **9**(11) (2008) 3
7. Wang, S., Zhang, Y., Qi, H., Zhao, M., Jiang, Y.: Dynamic spatial-temporal hypergraph convolutional network for skeleton-based action recognition. In: 2023 IEEE International Conference on Multimedia and Expo (ICME). pp. 2147–2152. IEEE (2023) 2
8. Yan, S., Xiong, Y., Lin, D.: Spatial temporal graph convolutional networks for skeleton-based action recognition. In: Proceedings of the AAAI conference on artificial intelligence. vol. 32 (2018) 2