Appendix for: Learning Spatial-Preserved Skeleton Representations for Few-Shot Action Recognition

Ning Ma[1,2,3], Hongyi Zhang[1,2,3] ⋆, Xuhui Li[1,2,3], Sheng Zhou[1,2,3] ⋆⋆, Zhen Zhang[4], Jun Wen[5], Haifeng Li[6], Jingjun Gu[1,2], and Jiajun Bu[1,2,3] ⋆⋆⋆

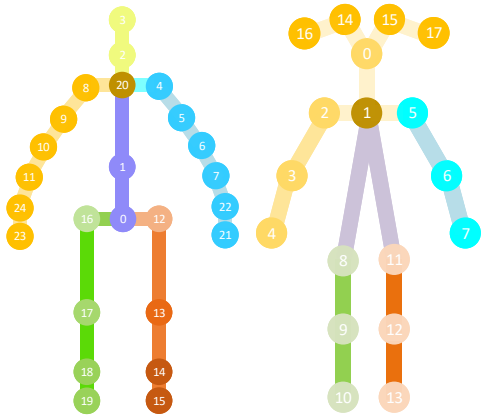
[1] Zhejiang Provincial Key Laboratory of Service Robot, College of Computer Science, Zhejiang University, Hangzhou, China
[2] Alibaba-Zhejiang University Joint Institute of Frontier Technologies, China
[3] Ningbo Research Institute, Zhejiang University, Ningbo, China
[4] Department of Computer Science, National University of Singapore, Singapore
[5] Department of Biomedical Informatics, Harvard Medical School, USA
[6] The Children's Hospital Zhejiang University School of Medicine, China
{ma_ning, zhy1998, 12021064, zhousheng_zju,junwen, 6199005, gjj, bjj}@zju.edu.cn, zhen@nus.edu.sg

A Dataset Partition

NTU RGB+D 120 [4] is a large-scale dataset with 3D joints annotations for human action recognition task. This dataset contains 113,945 skeleton sequences over 120 action classes captured from 106 distinct subjects and 32 different camera setups. Each skeleton graph contains N = 25 body joints as nodes (see Fig. 1a) in each frame, with their 3D locations in space as the initial features. Each frame of the action contains 1 to 2 subjects. For 2 subjects condition, to demonstrate the effectiveness of our core strategy, we just follow the original ST-GCN that encodes each skeleton independently and averages them. Our experiments use 120 action categories, including 80, 20 and 20 as training, validation and test categories. For each category, we randomly take 60 samples and 30 samples, denoted as two subsets "**NTU-S**" and "**NTU-T**", respectively. Here are the concrete train/validation/test classes:

1. **Train:** [0, 1, 2, 3, 5, 6, 7, 8, 10, 11, 13, 16, 17, 18, 19, 20, 21, 22, 23, 25, 27, 28, 29, 31, 33, 34, 36, 39, 40, 41, 46, 48, 49, 52, 53, 55, 58, 63, 64, 65, 66, 68, 70, 71, 72, 73, 74, 76, 77, 78, 79, 80, 81, 82, 84, 85, 87, 88, 89, 90, 91, 93, 94, 96, 98, 99, 101, 103, 104, 105, 106, 107, 108, 109, 110, 113, 114, 115, 117, 118]
2. **Validation:** [15, 35, 37, 38, 42, 47, 50, 51, 54, 56, 57, 59, 60, 61, 67, 69, 92, 95, 97, 116]
3. **Test:** [4, 9, 12, 14, 24, 26, 30, 32, 43, 44, 45, 62, 75, 83, 86, 100, 102, 111, 112, 119]

⋆ Equal Contribution With the First Author
⋆⋆ Corresponding Author
⋆⋆⋆ Corresponding Author



(a) NTU RGB+D 120 (b) Kinetics

Fig. 1: The used skeletons in our experiments.

Please see the website of NTU RGB+D 120 for more details[7].

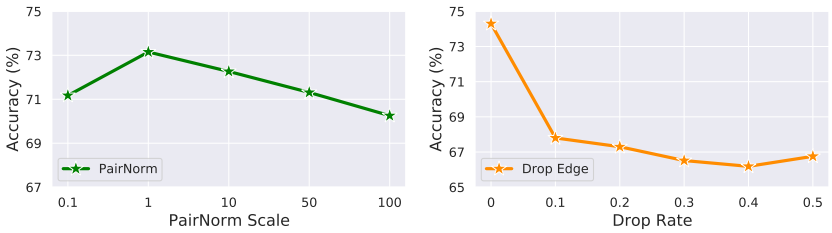
The dataset contains 260,232 videos over 400 classes, where each skeleton graph has 18 body joints (see Fig. 1b) after pose estimation, along with their 2D spatial coordinates and the prediction confidence score from OpenPose [1] as the initial joint features. At each frame, the number of skeletons is capped at 2 in each action skeleton, and skeletons with lower overall confidence scores are discarded. In our experiments, we only use the first 120 actions with 100 samples per class. The numbers of training/validation/test categories are 80/20/20. Here are the concrete train/validation/test classes:

1. **Train:** [1, 2, 5, 6, 8, 9, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 38, 39, 40, 41, 43, 47, 48, 49, 50, 51, 52, 58, 59, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 82, 83, 84, 85, 86, 87, 89, 90, 91, 93, 95, 97, 102, 105, 109, 110, 111, 113, 117, 118, 119]
2. **Validation:** [4, 7, 10, 36, 37, 42, 44, 45, 55, 56, 61, 88, 94, 98, 103, 104, 106, 108, 116, 120]
3. **Test:** [3, 22, 35, 46, 53, 54, 57, 60, 62, 80, 81, 92, 96, 99, 100, 101, 107, 112, 114, 115]
   The origin video dataset can be seen from the public link[8]. To download the skeleton data, please see this public link[9].

B More Discussion for Baselines

Why NGM [2] is a competitive baseline?

[7] https://rose1.ntu.edu.sg/dataset/actionRecognition/
[8] https://deepmind.com/research/open-source/kinetics
[9] https://github.com/open-mmlab/mmskeleton/blob/master/doc/SKELETON_DATA.md



(a) PairNorm (b) DropEdge

Fig. 2: (a). The test performances given different scale parameters in PairNorm [9] strategy. (b) .The test performances given different drop rates in Dropedge [5] strategy.

NGM (Neural Graph Matching) tries to solve few-shot action recognition with graph matching. Due to the lack of official implementation, we try our best to reproduce an enhanced version in our spatial-temporal alignment framework. NGM jointly learns a graph generator and a graph matching metric function in an end-to-end fashion to optimize the few-shot learning objective directly. In graph generator, they extract the spatial elements as graph nodes and learn adaptive edges to construct the scene graph from each frame. With graph matching metric function, they generate graph tensor following the calculation of node similarities. However, for our skeleton sequences, there have been skeleton graphs with pre-defined edges. Therefore, to compare with NGM, we mainly reproduce the Neural Graph Matching derived by edge weights within a graph, where these edge weights is replaced by our self-attention mechanism. Furthermore, our implementation is based on time alignment methods, which further improves the enhanced NGM.

How to select optimal hyper-parameter for PairNorm?

PairNorm [9] aims to tackle over-smoothing in graph neural networks (GNNs). By centering and rescaling node's representations, PairNorm uses a normalization after graph convolution layer to prevent all node embeddings from becoming too similar. In the official implementation[10], the scale of input is a hyper-parameter. We perform 5-way-1-shot task on NTU-T, with ST-GCN as backbone and DTW as time alignment method. From Fig. 2a, the optimal selection is around 1, which is used scaling value for all experiments.

How does Dropedge impact model's performance?

Both PairNorm and DropEdge are designed to alleviate the over-smoothing problem in large noisy graphs. **DropEdge** [5] randomly drops a few edges in input graphs to make the nodes aggregation from their neighbor diverse. Fig. 2b demonstrate the test performances with different drop rates. Only dropping 10% edges (24*0.1=2.4 edges) can damage the graph structure. Note that the experi-

[10] https://github.com/LingxiaoShawn/PairNorm

ments are performed on 5-way-1-shot task on NTU-T, with ST-GCN as backbone and DTW as time alignment method.

C More Discussion for Backbones

Can smaller backbones alleviate the over-smoothing problem?

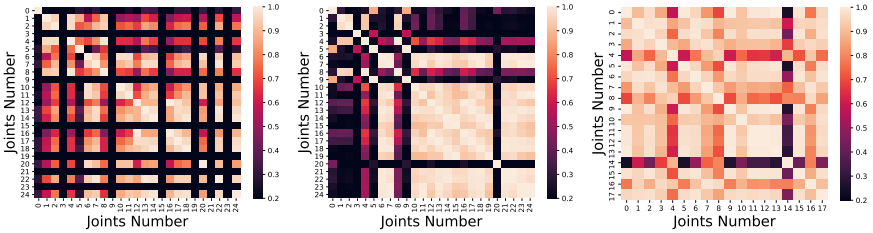
ST-GCN	(blocks=4)	(blocks=7)	(blocks=10)
seed 1	71.19	71.71	69.63
seed 2	70.58	71.78	73.15
seed 3	73.23	71.03	70.79
Mean	71.67	71.51	71.19

Table 1: Different performances with varying blocks of ST-GCN backbone. These experiments are performed with 5-way-1-shot task using ProtoNet baseline on NTU-T.

Methods	NTU-T	NTU-S	Kinetics
LSTM	$76.8_{\pm 0.9}$	$80.3_{\pm 0.9}$	$47.3_{\pm 0.4}$
ST-GCN	$82.4_{\pm 0.2}$	$84.3_{\pm 0.3}$	$46.8_{\pm 0.4}$
2s-AGCN	$81.9_{\pm 0.1}$	$84.2_{\pm 0.1}$	$50.5_{\pm 0.2}$
MS-G3D	$82.3_{\pm 0.3}$	$85.3_{\pm 0.1}$	$50.0_{\pm 0.3}$

Table 2: The 5-way-5-shot performances using different backbones with ProtoNet only.

To alleviate the over-smoothing problem, a simple idea is reducing the layers of the encoder backbone to decrease the number of graph convolution. In the released model of ST-GCN [8], there are 10 spatial-temporal convolution blocks capturing the joint interactions (the longest distance among skeleton joints approximates 10). In Tab. 1, we can see the released backbone (10 blocks) has a slight performance dropping compared with fewer blocks. To explore the reason behind it, we further visualize the intra-skeleton similarity by heatmap (see Fig. 3). From Fig. 3, we can observe that the backbone with 4 blocks has the lowest over-smoothing due to limited graph convolution. With the increasing of blocks, the over-smoothing is magnified. However, the reduction of spatial convolution and temporal convolution could not capture enough joint interactions, which is why the smaller backbones only achieve slight improvement with the alleviated over-smoothing. Inspired by this, our strategy is to keep the full blocks of ST-GCN to capture the joint interactions and alleviate the over-smoothing in the meantime.



(a) 4 blocks ST-GCN (b) 7 blocks ST-GCN (c) 10 blocks ST-GCN

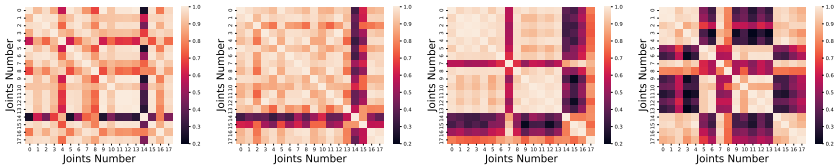
Fig. 3: The similarity heatmaps with varying layers of ST-GCN backbone. These experiments are performed with 5-way-1-shot task using Cosine baseline on NTU-T.

Furthermore, the small size backbone like LSTM [3] is inferior compared with ST-GCNs due to the lack of spatial encoding (see Tab. 2).

D Detailed Visualization Results

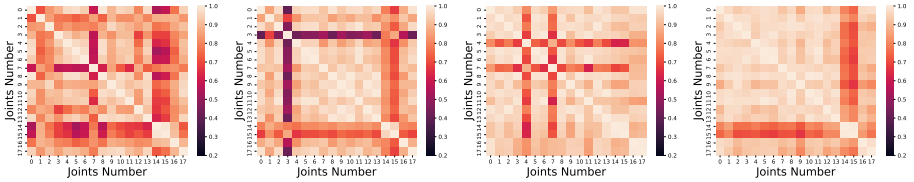
More visualization details for our spatial disentanglement strategy.

In this section, we provide more visualization results for different spatial recover strategies. We investigate "None" (Fig. 4), "DropEdge" (Fig. 6), "PairNorm" (Fig. 5) and "RankMax" (Ours, Fig. 7) and provide more detailed heatmaps. Note that all the spatial recover strategies are incorporated with DTW [6] + ProtoNet [7] on NTU-T dataset using 5-way-1-shot task.



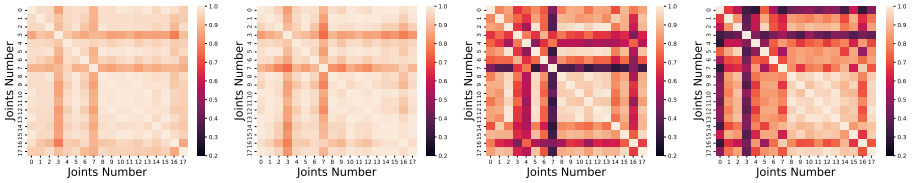
(a) (b) (c) (d)

Fig. 4: The intra-skeleton joints similarity heatmaps without any spatial recovery strategies. (a)&(b)&(c). The joints located in the head are disentangled rather than hand joints or feed joints, while these head joints usually denote less action information in the dataset NTU-RGB+D 120. Moreover, the over-smoothing problem is serious especially in the third figure. (d). Adjacent joints such as $\{0, 1, 2, 3, 4\}$ and $\{14, 15, 16, 17\}$ are over smoothed in graph convolution process.



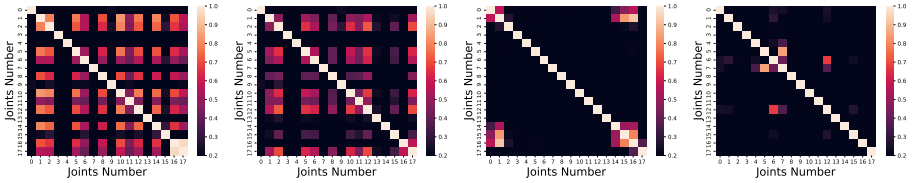
(a) (b) (c) (d)

Fig. 5: The intra-skeleton joints similarity heatmaps according to **PairNorm** [9] strategy. (a). Only the "hand" joint (number 7) is disentangled from others. (b). Only the "elbow" joint (number 3) is disentangled from others. (c) The left "hand" and right "hand" (numbers 4 and 7) are disentangled from others. (d). The unimportant joints located in the head (number 14, 15) are disentangled from others, which may harm the skeleton matching.



(a) (b) (c) (d)

Fig. 6: The intra-skeleton joints similarity heatmaps according to DropEdge [5] strategy. (a)&(b). Almost all joints have similar representations, which denotes that randomly dropping edges does not alleviate the over-smoothed graph convolution but rather may lead to the contrary due to destroyed skeleton structure. (c). The right "elbow", right "hand", left "hand" (number 3, 4 and 7, respectively) are disentangled from others, but the lower limb joints (number 8-13) are over smoothed. (d). The head joint (number 0) and left "elbow" (number 6) are disentangled from others, but the lower limb joints (number 8-13) are over smoothed.



(a) (b) (c) (d)

Fig. 7: The intra-skeleton joints similarity heatmaps according to **our RankMax** strategy. (a)&(b). The representative joints such as hands (number 4 and 7) and feet (number 13) are disentangled from other joints, providing disentangled spatial features for skeleton matching. (c)&(d). Almost all joints are disentangled given higher weight λ.

References

1. Cao, Z., Hidalgo, G., Simon, T., Wei, S.E., Sheikh, Y.: Openpose: realtime multi-person 2d pose estimation using part affinity fields. IEEE transactions on pattern analysis and machine intelligence **43**(1), 172–186 (2019)
2. Guo, M., Chou, E., Huang, D.A., Song, S., Yeung, S., Fei-Fei, L.: Neural graph matching networks for fewshot 3d action recognition. In: Proceedings of the European conference on computer vision (ECCV). pp. 653–669 (2018)
3. Hochreiter, S., Schmidhuber, J.: Long Short-Term Memory. Neural Computation **9**(8), 1735–1780 (11 1997). https://doi.org/10.1162/neco.1997.9.8.1735, https://doi.org/10.1162/neco.1997.9.8.1735
4. Liu, J., Shahroudy, A., Perez, M., Wang, G., Duan, L.Y., Kot, A.C.: Ntu rgb+d 120: A large-scale benchmark for 3d human activity understanding. IEEE Transactions on Pattern Analysis and Machine Intelligence **42**(10), 2684–2701 (2020)
5. Rong, Y., Huang, W., Xu, T., Huang, J.: Dropedge: Towards deep graph convolutional networks on node classification. In: International Conference on Learning Representations (2020), https://openreview.net/forum?id=Hkx1qkrKPr
6. Sakoe, H., Chiba, S.: Dynamic programming algorithm optimization for spoken word recognition. IEEE Transactions on Acoustics, Speech, and Signal Processing **26**(1), 43–49 (1978). https://doi.org/10.1109/TASSP.1978.1163055
7. Snell, J., Swersky, K., Zemel, R.S.: Prototypical networks for few-shot learning. arXiv preprint arXiv:1703.05175 (2017)
8. Yan, S., Xiong, Y., Lin, D.: Spatial temporal graph convolutional networks for skeleton-based action recognition. In: Thirty-second AAAI conference on artificial intelligence (2018)
9. Zhao, L., Akoglu, L.: Pairnorm: Tackling oversmoothing in gnns. In: International Conference on Learning Representations (2020), https://openreview.net/forum?id=rkecl1rtwB