Supplementary Materials
Learning Skeletal Graph Neural Networks for Hard 3D Pose Estimation

Ailing Zeng[1], Xiao Sun[2], Lei Yang[3], Nanxuan Zhao[1], Minhao Liu[1], Qiang Xu[1]
[1]The Chinese University of Hong Kong
[2]Microsoft Research Asia, [3]Sensetime Group Ltd.

This supplementary material presents more experimental details, including data pre-processing, implementation details, additional experimental results, and ablation studies.

1. 3D Human Pose Estimation

In this section, we demonstrate more detailed results on 3D human pose estimation. Sec. 1.1 gives more details on experiment settings. Second, Sec. 1.2 analyzes features of hard poses in this task. Third, Sec. 1.3 compares existing methods by the metric of PA-MPJPE. Finally, Sec. 1.4 shows the ablation study of only using a dynamic graph with HCSF module.

1.1. Dataset and Implementation Details

1.1.1 Dataset Pre-processing

We follow our baseline [2] to transform the 3D joint position under the camera coordinate system into the pixel coordinate system to remove the influence of pose scales for the single-view pose estimation. Following previous works [11, 2, 20], we normalize 2D input poses in the range of [-1, 1] according to the width and height of images. The furthest hop is 6 in our pre-defined topology. Meanwhile, we set the entry values of the adjacency matrix to be ones if two nodes are physically connected and zero if not.

1.1.2 Training Details

We build a six-layer network as the basic setting, including the first layer, two cascaded blocks, and the last layer. For a single-frame setting, each cascaded block consists of two HCSF layers followed by BN, LeakyReLU (alpha is 0.2), and dropout (random drop probability is 0.25). Besides, each block is wrapped with a residual connection, as shown in Fig.3 in the main paper. The channel size of each layer we report in the final result is 128. In the ablation study, we set all output channels as 64 for each node. The above framework is a common structure that is also used in those works [9, 11, 2, 21, 20]. For temporal settings, each cascaded block consists of one HCSF layer and one TCN layer. The fusion functions $\mathcal{F}_k$ and $\mathcal{F}_a$ are concatenation operators by default, which can also be addition, multiplication. L1 regression loss is used between the ground truth and outputs. Moreover, we train our model for 80 epochs using Adam [4] optimizer. The initial learning rate is set as 0.001, and the exponential decay rate is 0.95. The mini-batch size is 256. For data augmentation, we follow [11, 2, 21, 20] and use horizontal flip data augmentation at both training and test stages. Then, we evaluate our method with standard protocol following [2, 21, 20, 11].

1.2. Further Analysis on Model-Specific Hard Poses

We define *high-error poses* as hard poses in the 2D-3D pose regression task. After analyzing the error distribution of hard poses in recent works [9, 21, 2, 20], we could conclude they are model-specific. As shown in Fig. 1, we illustrate the comparison of the ($50\% \sim 5\%$) hardest poses from each method. For example, Fig. 1(a) shows the ($50\% \sim 5\%$) hardest poses from the fully connected network [9], and we compare the results with the other four methods under the same poses.

We can observe: (1) The hardest 10% poses of each method is different, indicating that hard poses are model-specific; (2) as the poses become increasingly difficult, the errors of all methods rise to some extent; (3) our method obtains the best results for the hardest poses of all the other four methods; (4) the error gap in Fig. 1(e) is smaller than Fig. 1(a∼d).

1.3. Comparison in PA-MPJPE

In Tab. 1, we compare our methods with other related works using the PA-MPJPE metric where available. We show the results from different 2D inputs, using detected poses or ground truth poses. Our approach achieves the new state-of-the-art with different inputs. Specifically, we surpass [20] from 27.8mm to 24.8mm (relative 10.8% improvement) with 2D ground truth input. Moreover, we improve upon [6] from 41.2mm to 39.0mm (relative 5.3% improvement) with 2D keypoint detection input. Our method can also show the superior in this metric, indicating the effectiveness of this method.

Method	Direct	Discuss	Eat	Greet	Phone	Photo	Pose	Purcha.	Sit	SitD	Smoke	Wait	WalkD	Walk	WalkT	Avg.
Martinez et al. [9]	39.5	43.2	46.4	47.0	51.0	56.0	41.4	40.6	56.5	69.4	49.2	45.0	49.5	38.0	43.1	47.7
Fang et al. [3]	38.2	41.7	43.7	44.9	48.5	55.3	40.2	38.2	54.5	64.4	47.2	44.3	47.3	36.7	41.7	45.7
Park et al. [10]	38.3	42.5	41.5	43.3	47.5	53.0	39.3	37.1	54.1	64.3	46.0	42.0	44.8	34.7	38.7	45.0
Hossain et al. [14] §	35.7	39.3	44.6	43.0	47.2	54.0	38.3	37.5	51.6	61.3	46.5	41.4	47.3	34.2	39.4	44.1
Zou et al. [22] †	38.6	42.8	41.8	43.4	44.6	52.9	37.5	38.6	53.3	60.0	44.4	40.9	46.9	32.2	37.9	43.7
Liu et al. [7]†	38.4	41.1	40.6	42.8	43.5	51.6	39.5	37.6	49.7	58.1	43.2	39.2	45.2	32.8	38.1	42.8
Ci et al. [2]†	36.9	41.6	38.0	41.0	41.9	51.1	38.2	37.6	49.1	62.1	43.1	39.9	43.5	32.2	37.0	42.2
Liu et al. [6]†	35.9	40.0	38.0	41.5	42.5	51.4	37.8	36.0	48.6	56.6	41.8	38.3	42.7	31.7	36.2	41.2
Ours-HCSF†	34.3	37.6	37.5	38.6	39.5	44.2	38.3	35.5	48.5	55.6	41.4	38.7	42.3	30.8	32.2	39.7
Ours-HCSF w/A†	**33.9**	**37.2**	**36.8**	**38.1**	**38.7**	**43.5**	**37.8**	**35.0**	**47.2**	**53.8**	**40.7**	**38.3**	**41.8**	**30.1**	**31.4**	**39.0**
Zeng et al. [20]§	24.3	28.1	24.3	28.1	27.4	29.8	28.3	25.6	27.8	34.5	27.5	27.7	31.8	25.7	25.6	27.8
Ours-HCSF† §	20.9	27.3	22.4	25.3	24.4	29.7	24.9	23.0	27.2	32.6	25.8	25.6	26.4	20.4	21.7	25.2
Ours-HCSF w/A† §	**20.7**	**26.9**	**22.1**	**24.8**	**24.0**	**29.1**	**24.5**	**22.7**	**26.8**	**32.1**	**25.3**	**25.2**	**26.0**	**20.2**	**21.5**	**24.8**

Table 1: Comparison results regarding PA-MPJPE after *rigid transformation* from the ground truth. We highlight the graph-based methods by †. § donates the use of 2D ground truth poses as input. Best results in bold.

Method	LCN	a	b	c	d	e	f	g
$\mathbf{A}_k$	ori	Only $\mathbf{M}_k$ (ori)	Only $\mathbf{M}_k$ $(dense)$	Only $\mathbf{M}_k$ $(rand)$	Only $\mathbf{O}_k$	$\mathbf{M}_k + \mathbf{O}_k$	Eq.8	Eq.8 w/T
MPJPE(mm)	35.7	34.8	35.5	41.2	46.1	34.3	34.0	**33.5**

Table 2: Comparison on the effects of dynamic graph learning $\mathbf{A}$ in a *Non-hierarchy strategy*. ori is the static graph with physical connections, shown in LCN [2]. *Baseline* takes $\mathbf{A}_k$ as ori. *Only* $\mathbf{M}_k$ $(\cdot)$ denotes applying $\mathbf{M}_k$ with different initialization. *Only* $\mathbf{O}_k$ keeps the dynamic offset in Eq.8. $\mathbf{M}_k + \mathbf{O}_k$ equals to set $\alpha = 1$ in Eq.8. *w/T* represents the temporal-aware scheme defined in Sec.3.3.

1.4. Ablation Study on Dynamic Graph

This work has two main contributions: Hierarchical Channel-Squeezing Fusion (HCSF) and temporal-aware dynamic graph learning. We further explore how temporal-aware dynamic graph alone influences the regression results. The 2D inputs are 2D ground truth to explore the upper bound of our method to avoid some irrelevant noises from detected 2D poses.

Effects of dynamic graph learning. Dynamic graph learning shows different action-related connectivity with different inputs. It can be more flexible to extract specific-action patterns, especially for hard poses. We have demonstrated the influence on both HCSF and dynamic graph learning in the main paper. Accordingly, we study the effects of dynamic graph learning alone. We take the *Non-hierarchy strategy* LCN with the static graph aggregating with hop-2 as a baseline. Similar to the Tab.6 in the main paper, the Tab. 2a, 2b, 2c shows that $\mathbf{M}_k$ (ori), using the physical topology as an initial connections, is better than $\mathbf{M}_k$ $(dense)$ and $\mathbf{M}_k$ $(rand)$. The weighted graph $\mathbf{M}_k$ (ori) can also surpass the *same* weighted graph in LCN. Moreover, only learning graph structure from features increase the error from 35.7mm to 46.1, which is infeasible. After combining the weighed graph $\mathbf{M}_k$ (ori) with the dynamic offset $\mathbf{O}_k$, we can obtain 0.5mm improvement. Furthermore, considering a dynamic scale α to control the influence of the dynamic offsets, which is the formula in Eq.8, will be helpful. Last, we can observe that the temporal-aware scheme can boost the performance, decreasing the MPJPE from 34.0mm to 33.5mm.

Effects of the temporal scale. The uncertainty in single-frame poses will affect the regression results, making dynamic graph learning unstable and misleading. Hence, it is essential to introduce temporal consistency to make the process effective. We then explore how different settings in the temporal-aware scheme impact the performance. The temporal-aware schemes are different from the receptive fields. We fix S=1, L=2, d=1/8. The channel size of each layer is 128. And the frame of input is 9. From Tab. 3, we can find that using the 3×1 kernel size will be better than other settings. And using temporal information will consistently improve the single-frame results by $0.1 \sim 0.4$mm. Thus, we report our final results using the 3×1 kernel size.

F	(1,1)	(3,1)	(3,1) w/$st.$=2	(3,1) w/$di.$=2	(5,1)	(7,1)
HCSF	30.8	**30.4**	30.7	30.7	30.6	30.7

Table 3: The impact of settings F of temporal convolution in dynamic graph learning of 3D human pose estimation. $st.$ is an abbreviation for $stride$, and $di.$ is $dilation$.

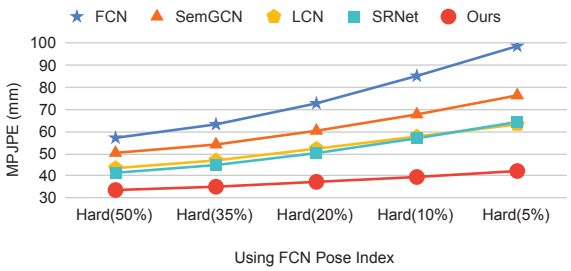
2. Skeleton-based Human Action Recognition

In this section, we present the experimental details, more results and ablation study of skeleton-based action recognition in Sec. 2.1, Sec. 2.2 and Sec. 2.3, respectively.

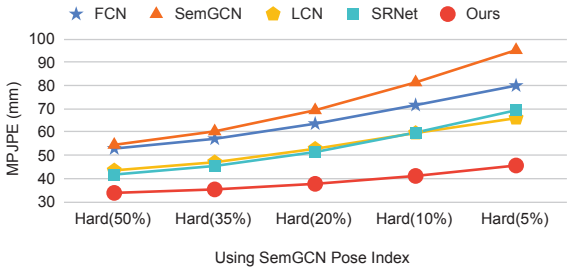
2.1. Dataset and Implementation Details

2.1.1 Data Description

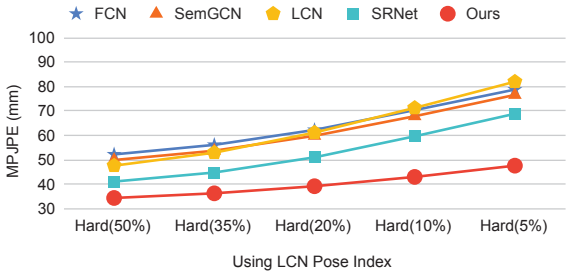
NTU RGB+D 60 [15] is one of the most widely used indoor RGB+Depth action recognition dataset with 60 actions. They include daily, mutual, and health-related actions. NTU RGB+D 60 has 40 subjects under three cameras. Following [16, 17, 19, 13, 18], we use skeleton sequences with 25 body joints captured by Kinect V.2 as inputs, and take two evaluation settings in NTU RGB+D 60: (1) Cross-Subject (X-Sub), where 20 subjects each for training and testing, respectively; (2) Cross-View (X-View),



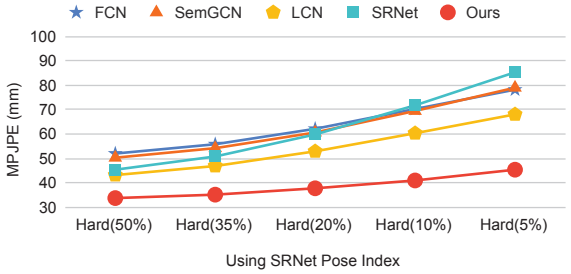
(a) The hard poses of FCN [9]



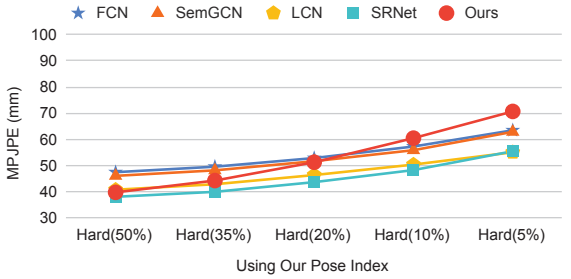
(b) The hard poses of SemGCN [21]



(c) The hard poses of LCN [2]



(d) The hard poses of SRNet [20]



(e) The hard poses of Ours

Figure 1: The comparison of the hard poses in terms of each method.

where 2 camera views for training and 1 camera view for testing. We perform the ablation study in Sec. 2.3 on the X-View setting.

NTU RGB+D 120 [5] collects 120 various actions by 106 distinct subjects and contains more than 114 thousand video samples and 8 million frames. We also follow some previous works [19, 8, 13, 12], using two evaluation settings: (1) Cross-Setup (X-Set), training on 16 camera setups and testing on other 16 camera setups; (2) Cross-Subject (X-Sub), half subjects for training and half for testing. We report the top-1 accuracy on both benchmarks.

2.1.2 Data Pre-processing

The procedure for both datasets follows [16, 17, 8]. Each video has a maximum of 300 frames, and if it is shorter than 300, we repeat some frames to make up for it. Since there are at most two people in both datasets, we pad the second body with zeros to keep the same shape of inputs when the second body does not appear.

2.1.3 Training Details

We build a ten-layer network, including nine cascaded blocks that consist of one HCSF layer followed by BN, ReLU, temporal convolution layer (TCN), BN and ReLU. Each temporal 1D convolution layer conducts 9×1 convolution on the feature maps. Each block is wrapped with a residual connection. The output dimension for each block are 64, 64, 64, 128, 128, 128, 256, 256 and 256. A global average pooling layer and a fully-connected layer are used to aggregate extracted features, and then, feed them into a softmax classifier to obtain the action class. The above framework is also a common setting as in [18, 16, 17, 19]. For multi-stream networks [17], we use four modalities, e.g., joints, bones and their motions, as inputs for each stream, and average their softmax scores to obtain the final prediction. Cross-entropy is used as the classification loss function to back-propagate gradients. We set the entry values in the adjacency matrix to be ones if two nodes are physically connected and zero if not.

For the training settings, we train our model for 60 epochs using the SGD optimizer with mini-batch size 64. The initial learning rate is 0.1 and it reduces by 10 times in both the 35_{th} and 45_{th} epoch, respectively. The weight decay is set as 0.0005. All data augmentation is the same as [16, 17].

2.2. Results of Single-Stream Framework

Due to space limitations, we only report the accuracy of the multi-stream framework [17] for the skeleton-based human action recognition task in the main paper. Specifically, the multi-stream network comprises four different modality inputs: the 3D skeleton joint position, the 3D skeleton bone vector, the motion of the 3D skeleton joint, and the motion

Method	NTU-RGB+D 60		NTU-RGB+D 120	
	X-Sub(%)	X-View(%)	X-Sub(%)	X-Set(%)
Joint	89.0	95.3	83.5	85.7
Bone	89.3	94.9	85.0	86.6
Joint-Motion	86.9	93.5	80.1	81.5
Bone-Motion	86.9	93.1	80.6	83.0
Multi-Stream	**91.6**	**96.7**	**87.5**	**89.2**

Table 4: Top-1 accuracy (%) is used as the evaluation metric. The best result in each K is in bold.

Decay Rate d	1	1/2	1/4	1/8	1/16
Static-$\mathcal{G}$	93.9	94.5	94.6	**94.8**	94.5
Dynamic-$\mathcal{M}$	94.4	94.9	**95.1**	94.9	**95.1**
Dynamic-$\mathcal{A}$	94.6	95.0	95.2	**95.3**	**95.3**

Table 5: The impact of decay rate d under static matrix $\mathcal{G}$, dynamic graph from $\mathbf{M_k}$, and dynamic graph from $\mathbf{A_k}$ in Eq.8.

of the 3D skeleton bone. Here, we report the performance of *each modality input* in Tab. 4 for the ease of comparison with existing works.

2.3. Ablation Study

We investigate the proposed methods on the NTU RGB-D X-View setting with 3D joint positions as inputs.

Effects of hierarchical channel-squeezing fusion block. From Tab. 5, our method improves the accuracy of 0.7% steadily under all three graph settings, static graphs $\mathcal{G}_k$ and two dynamic graphs $\mathcal{M}_k$ and $\mathcal{A}_k$ in Eq.8. Basically, better results can be achieved when d=1/8. Moreover, we get the best results when using HCSF with dynamic graph $\mathcal{A}_k$, which validates the effectiveness of the proposed structure.

Furthermore, in Tab. 6, we demonstrate the performance of different methods concerning the number of hops. Since the skeleton topology in NTU-RGBD datasets is different from Human3.6M, it has more keypoints and further hops. The furthest hop is 13 in our pre-defined topology. We set S=5, L=7 and d=1/8. k-hop (k=1, 5, 7) means aggregating the neighbors within the distance k (1-hop with a static graph is ST-GCN [18]). Mixhop [1] means that it concatenates the k-hop (k=1, 5, 7) features as the output of a layer, and the output size of the k-hop feature is one-third of the final output. MS-Hop means that it averages the k-hop (k=1, 5, 7) features, and the output size of the k-hop feature is the same as the final output.

As illustrated in Tab. 6, though MixHop and MS-Hop show improvements on k-hop strategies, they have no distinction in handling distant and close neighbors, which over-mix the useful and noisy information. Our approaches outperform all other baselines, which indicates the effectiveness of the hierarchical channel-squeezing fusion strategy.

Additionally, we explore the effects of other hyper-parameters in the HCSF. We have the following observations. First, when using a dynamic graph $\mathcal{A}_k$ in Eq.8 and fixing the hyper-parameters squeezing ratio d and the output channel size C in a layer, we find little effects on the results that S and L has. The accuracy is stable around 95.1% ($\sim$ 0.2%). It indicates that the HCSF is robust to the noise in the graph. Second, as the number of hops increases, the performance first improves and then becomes stable. Since adding more hops leads to extra computations, to balance the computation efficiency and performance, our final setting for each layer is S=5, L=7, d=1/8, C of each layer is the same as [18, 16, 17]. Last, we also explore to automatically learn the relations between hops and dimensions with the guidance of channel attention. However, we find that the exponentially decaying in dimension consistently yields better results than the soft attention, which may be because the soft attention mechanism introduces more uncertainty and complexity.

Method	1-hop	5-hop	7-hop	MixHop	MS-Hop	Ours
Static $\mathcal{G}$	92.2	93.5	93.7	93.9	94.1	**94.8**
Dynamic-$\mathcal{M}$	93.4	94.1	94.1	94.5	94.6	**95.2**
Dynamic-$\mathcal{A}$	93.9	94.3	94.2	94.8	94.7	**95.3**

Table 6: Comparison on various multiple hop structures under static matrix $\mathcal{G}_K$, dynamic graph from $\mathcal{M}_k$, and a dynamic graph from $\mathcal{A}_k$. Top-1 accuracy is used as the evaluation metric.

F	(1,1)	(3,1)	(3,1) w/$st.$=2	(3,1) w/$di.$=2	(5,1)	(7,1)
HCSF	94.7	**95.3**	95.0	94.8	95.1	94.7

Table 7: The impact of settings of temporal convolution in dynamic graph learning of skeleton-based action recognition. $st.$ is an abbreviation for $stride$, and $di.$ is $dilation$.

Effects of the temporal-aware dynamic graph learning. The jitter and missing inputs will make dynamic graph learning unreliable, making it difficult to distinguish between similar actions, e.g., "eat a meal" and "brushing teeth." Such problems are serious in using single-frame features, but they can be improved by involving temporal information. From Tab. 7, we can observe that when using three frames into a temporal convolution, it can improve the single-frame setting by 0.6%. While the settings of temporal aggregation are important, the longer temporal contexts will also degrade the performance, and use three frames will be the optimal setting.

References

[1] Sami Abu-El-Haija, Bryan Perozzi, Amol Kapoor, Nazanin Alipourfard, Kristina Lerman, Hrayr Harutyunyan, Greg Ver Steeg, and Aram Galstyan. Mixhop: Higher-order graph convolutional architectures via sparsified neighborhood mixing. *arXiv preprint arXiv:1905.00067*, 2019.

[2] Hai Ci, Chunyu Wang, Xiaoxuan Ma, and Yizhou Wang. Optimizing network structure for 3d human pose estimation. In

Proceedings of the IEEE International Conference on Computer Vision, pages 2262–2271, 2019.

[3] Hao-Shu Fang, Yuanlu Xu, Wenguan Wang, Xiaobai Liu, and Song-Chun Zhu. Learning pose grammar to encode human body configuration for 3d pose estimation. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[4] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[5] Jun Liu, Amir Shahroudy, Mauricio Lisboa Perez, Gang Wang, Ling-Yu Duan, and Alex Kot Chichung. Ntu rgb+ d 120: A large-scale benchmark for 3d human activity understanding. *IEEE transactions on pattern analysis and machine intelligence*, 2019.

[6] Kenkun Liu, Rongqi Ding, Zhiming Zou, Le Wang, and Wei Tang. A comprehensive study of weight sharing in graph networks for 3d human pose estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020.

[7] Kenkun Liu, Zhiming Zou, and Wei Tang. Learning global pose features in graph convolutional networks for 3d human pose estimation. In *Proceedings of the Asian Conference on Computer Vision*, 2020.

[8] Ziyu Liu, Hongwen Zhang, Zhenghao Chen, Zhiyong Wang, and Wanli Ouyang. Disentangling and unifying graph convolutions for skeleton-based action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 143–152, 2020.

[9] Julieta Martinez, Rayat Hossain, Javier Romero, and James J Little. A simple yet effective baseline for 3d human pose estimation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2640–2649, 2017.

[10] Sungheon Park and Nojun Kwak. 3d human pose estimation with relational networks. *arXiv preprint arXiv:1805.08961*, 2018.

[11] Dario Pavllo, Christoph Feichtenhofer, David Grangier, and Michael Auli. 3d human pose estimation in video with temporal convolutions and semi-supervised training. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7753–7762, 2019.

[12] Wei Peng, Xiaopeng Hong, Haoyu Chen, and Guoying Zhao. Learning graph convolutional network for skeleton-based human action recognition by neural searching. In *AAAI*, pages 2669–2676, 2020.

[13] Wei Peng, Jingang Shi, Zhaoqiang Xia, and Guoying Zhao. Mix dimension in poincaré geometry for 3d skeleton-based action recognition. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 1432–1440, 2020.

[14] Mir Rayat Imtiaz Hossain and James J Little. Exploiting temporal information for 3d human pose estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 68–84, 2018.

[15] Amir Shahroudy, Jun Liu, Tian-Tsong Ng, and Gang Wang. Ntu rgb+ d: A large scale dataset for 3d human activity analysis. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1010–1019, 2016.

[16] Lei Shi, Yifan Zhang, Jian Cheng, and Hanqing Lu. Skeleton-based action recognition with multi-stream adaptive graph convolutional networks. *arXiv preprint arXiv:1912.06971*, 2019.

[17] Lei Shi, Yifan Zhang, Jian Cheng, and Hanqing Lu. Skeleton-based action recognition with multi-stream adaptive graph convolutional networks. *IEEE Transactions on Image Processing*, 29:9532–9545, 2020.

[18] Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. *arXiv preprint arXiv:1801.07455*, 2018.

[19] Fanfan Ye, Shiliang Pu, Qiaoyong Zhong, Chao Li, Di Xie, and Huiming Tang. Dynamic gcn: Context-enriched topology learning for skeleton-based action recognition. *arXiv preprint arXiv:2007.14690*, 2020.

[20] Ailing Zeng, Xiao Sun, Fuyang Huang, Minhao Liu, Qiang Xu, and Stephen Lin. Srnet: Improving generalization in 3d human pose estimation with a split-and-recombine approach. *arXiv preprint arXiv:2007.09389*, 2020.

[21] 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 Conference on Computer Vision and Pattern Recognition*, pages 3425–3435, 2019.

[22] Zhiming Zou, Kenkun Liu, Le Wang, and Wei Tang. High-order graph convolutional networks for 3d human pose estimation. BMVC, 2020.