Spatial Temporal Graph Deconvolutional Network for Skeleton-Based Human Action Recognition

Wei Peng, Jingang Shi, and Guoying Zhao, *Senior Member, IEEE*

***Abstract*—Benefited from the powerful ability of spatial temporal Graph Convolutional Networks (ST-GCNs), skeleton-based human action recognition has gained promising success. However, the node interaction through message propagation does not always provide complementary information. Instead, it May even produce destructive noise and thus make learned representations indistinguishable. Inevitably, the graph representation would also become over-smoothing especially when multiple GCN layers are stacked. This paper proposes spatial-temporal graph deconvolutional networks (ST-GDNs), a novel and flexible graph deconvolution technique, to alleviate this issue. At its core, this method provides a better message aggregation by removing the embedding redundancy of the input graphs from either node-wise, frame-wise or element-wise at different network layers. Extensive experiments on three current most challenging benchmarks verify that ST-GDN consistently improves the performance and largely reduce the model size on these datasets.**

***Index Terms*—Graph neural network, skeleton-based action recognition, over-smoothing.**

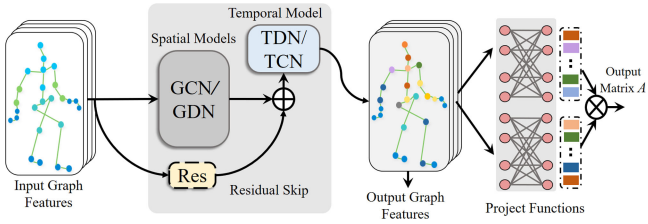


Fig. 1. **Illustration of the ST-GDNs block.** There are two core models in the block, spatial models including both GCNs and GDNs, and a temporal model, which can be either normal temporal filter (TCN) or temporal deconvolutional networks (TDN). Our ST-GDNs are different combinations of them. Here, $\oplus$ is the element-wise summation. $\otimes$ denotes matrix multiplication.

I. Introduction

RECENT years, skeleton-based action recognition has attracted great attention since the data is more compact and more robust to complex background, when compared to RGB inputs [1]–[6]. Formerly, deep neural models, including both convolutional neural networks (CNNs) [1], [7]–[10] and Recurrent Neural Networks (RNNs) [4], [11], [12], achieve promising results and become mainstream methods since they are able to automatically learn more distinguishable features from data. Nevertheless, just like for another irregular data, conventional neural networks like CNNs and RNNs are designed in Euclidean space thus the skeleton-based action recognition does not significantly benefit from the neural networks. Fortunately, by introducing GCNs into this task, remarkable improvements have been witnessed [13]–[19]. Yan *et al.* first proposed to use spatial-temporal GCN for this task [14] and it becomes one of the most common framework to skeleton-based action recognition. Derived from ST-GCN, Shi *et al.* proposed to add virtual typology to involve more semantic information [18]. Likewise, Peng *et al.* [19] turn to neural architecture search, NAS [20], and automatically construct ST-GCN module [19] for this task.

Nevertheless, as mentioned before, ST-GCNs capture and extract graph embeddings via a message passing paradigm, which makes the representation from different nodes indistinguishable to each other. Message propagation has the ability to enhance the interactions between nodes with correlation especially from topology structure. However, interaction with unrelated nodes May not get complementary information but noise which May even harm the original node representation. Especially from the high semantic level, node interactions based on topology connections May lead to very similar embeddings, which is unreasonable.

In this letter, we propose a novel graph neural architecture, referred as spatial temporal graph deconvolutional network (ST-GDN), to deal with the aforementioned issue. As shown in Fig. 1, the deconvolution operation provides a filter which can reshape and transform the graph features before the filtering. By changing the coordinates in a new feature space, the feature embeddings are standardized and unrelated to each other. With the deconvolutional operations, we remove the correlations and redundancy of the graph representations from either node-wise level, frame-wise level, or element-wise level. Finally, we evaluate our model on three current most challenging skeleton-based human action recognition tasks. Our contribution can be summarized as follows:

- We present a novel and flexible Graph Deconvolution Network (GDN), which is designed to address the graph over-smoothing problem and also can be easily plugged into variant graph neural networks.

Manuscript received November 10, 2020; revised December 24, 2020; accepted December 27, 2020. Date of publication January 6, 2021; date of current version February 4, 2021. This work was supported in part by the ICT2023 Project under Grant 328115, in part by the Academy of Finland for Project MiGA under Grant 316765, in part by the Infotech Oulu, and also with the National Natural Science Foundation of China under Grant 62002283. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Dezhong Peng. *(Corresponding author: Guoying Zhao)*

Wei Peng and Guoying Zhao are with the Faculty of Information Technology and Electrical Engineering, University of Oulu, 90570 Oulu, Finland (e-mail: ikerpeng@gmail.com; guoying.zhao@oulu.fi).

Jingang Shi is with the School of Software Engineering, Xi'an Jiaotong University Xi'an 710049, China (e-mail: jingang.shi@oulu.fi).

Digital Object Identifier 10.1109/LSP.2021.3049691

- We provide various ST-GCNs from different levels. By combining them at different layers, we present a brand-new graph neural network, named ST-GDN, which could largely reduce the model parameters as well as improve the feature representative ability.
- We utilize this model to deal with skeleton-based human action recognition tasks. The results on three current most challenging datasets show that we can get the best performance on any given evaluation metrics with an efficient fashion.

II. Proposed Approach

In this section, we will detail the theory of our approach and four various blocks, referring as ST-GDNs, including Node-wise ST-GDN (ST-GDN^2), frame-wise ST-GDN (ST-GDN-T), element-wise ST-GDN (ST-GDN-E), and a combination of GCN and GDN (ST-GDCN).

1) ST-GCNs: The basic GCN model here is designed based on the chebyshev polynomial approximation [21], [22], in which the matrix L has a complete set of orthonormal eigenvectors $U = [u_1, u_2, .., u_n]$, which are the Laplacian eigenvectors associated with non-negative eigenvalue $\{\lambda_l\}_{l=1}^n$ such that $L = U\Lambda U^T$ for $\Lambda = diag(\{\lambda_l\}_{l=1}^n)$. Then, taking the eigenvectors of the normalized Laplacian matrix as a set of bases, the graph convolution operator is defined by the Fourier transformation. Current GCNs approximate this convolution operations with $(K-1)$-th order polynomial expansion. Since our inputs are a sequence of skeletons, we introduce temporal filters to capture the dynamic information from the inputs. Inspired from (2+1)D convolution networks [23], we make the above mentioned GCN be followed by a temporal filter, thus the output representation should be

$$y = \Theta_t^\tau \left(U \left(\sum_{k=0}^{K-1} \theta_k \lambda^k \right) U^T x \right). \tag{1}$$

where Θ_τ^t is an 1D temporal convolutional filter with kernel size of τ, θ_k is the polynomial coefficient for the k-th order.

1) ST-GDNs: The basic architecture in the network is a spatial graph model(s) followed by a temporal model, as illustrated in Fig. 1. The spatial model can be either a GCN, a node-wise GDN, or their combinations. The temporal model can be a temporal convolutional filter (TCN) or a temporal deconvolutional network (TDN). Given a sequence of human skeletons or skeleton feature embeddings, the ST-GDNs output higher level representations (Output Graph Features in Fig. 1) and a dynamic graph embedding matrix (Matrix A in Fig. 1). Based on the representations, a dynamic graph embedding matrix is also provided for each GCN layer. We stack multiple ST-GDNs to learn the graph embedding.

Over-smoothing leads to very similar representations for each node or even each feature element. Our ST-GDNs address this problem via filtering graph with representation standardization. By changing the coordinates in a new feature space, the feature embeddings are standardized and unrelated to each other, thus the issue is relieved. Here, we begin to introduce the proposed node-wise deconvolution firstly. It is very easy to generalize to frame-wise and element-wise models. Assume $X \in R^{N\times T\times F}$ is the N node graph representations for a T frames skeleton sequence and the feature dimension for each node is F. From Eq. (1) we know that, a convolutional filter will be introduced to capture the node embeddings for the tensor X. Assume that the kernel size of the convolutional filter is $k_1 \times k_2$. In practice, the filter multiples with an unfold tensor of X, in which there are much redundancy and unavoidable will lead to the over-smoothing issue. Suppose we unfold the embedding X to m feature blocks while each block contains $N \times k_1 \times k_2$ elements. The number of feature block m can be represented by

$$m = \prod_{i=\{1,2\}} \left(\frac{S_i + 2 \times p_i - d_i \times (k_i - 1) - 1}{t_i} + 1 \right). \tag{2}$$

The index i belongs to the set $\{1, 2\}$ since the GCN is a 2D operation. $S_i = \{T, F\}$ is the length of feature along each dimension. The parameters p_i, d_i, k_i and t_i are the padding value, dilation value, kernel size and stride value for the corresponding dimension, respectively. Thus, we can obtain the unfolded feature embedding $X \in R^{(N\times k_1\times k_2)\times m}$. Instead of directly filter this tensor, we preform the deconvolution on it. Therefore, once getting this unfolded embedding, we first calculate the mean node embedding $\mu \in R^{N\times k_1\times k_2}$ for the m feature blocks. Then, a covariance matrix can be written as

$$C = (X - \mu)^T (X - \mu) + \epsilon I. \tag{3}$$

To make it more stable, a very small value ϵ is added to the diagonal of the matrix. The graph deconvolution operation can be a GCN on the transformed input feature with the covariance matrix, that is:

$$Y = U \left(\sum_{k=0}^{K-1} \theta_k \lambda^k \right) U^T ((X - \mu)C^{-\frac{1}{2}}). \tag{4}$$

Here, the transformed feature representation $(X - \mu)C^{-\frac{1}{2}}$ has an identity matrix since its covariance is

$$((X - \mu)C^{-\frac{1}{2}})^T ((X - \mu)C^{-\frac{1}{2}}) = I. \tag{5}$$

By changing the coordinates in a new feature space, the feature embeddings are standardized and unrelated to each other. This operation can be considered as a deconvolution since it can negates the process of convolution. It means that for a delta kernel δ, the transformed feature $XC^{-\frac{1}{2}}$ will not be changed by using kernel $C^{\frac{1}{2}}\delta$. In this case, the deconvolution kernel is $C^{-\frac{1}{2}} \cdot vec(\delta)$, where $vec(\delta)$ is equal to slicing the middle row/column of $C^{-\frac{1}{2}}$ and reshaping it to the kernel size.

However, calculating the inverse square root of a matrix is still computationally expensive and unstable. Instead of directly computing it, like [24], we further use coupled Newton-Schulz iterations to reduce the cost. The iteration starts with $Y_0 = C, Z_0 = I$, and could be executed by

$$Y_{k+1} = \frac{1}{2} Y_k (3I - Z_k Y_k), Z_{k+1} = \frac{1}{2} (3I - Z_k Y_k) Z_k. \tag{6}$$

Once the iteration is done, the final result of Z_k will converge to $C^{-\frac{1}{2}}$. Then, the value of Z_k could be utilized to approximate the $C^{-\frac{1}{2}}$ as a trade-off between efficiency and accuracy.

Here, the deconvolutional operation in Eq. (4) can be further simplified by only considering the first-order polynomial approximation and setting the polynomial coefficients $\theta = \theta_0 = -\theta_1$. Then, the learnable θ is expected to make the approximation more robust and higher-order node connections can be

captured by stacking multiple layers. Thus, the output from single GDN layer can be represented as

$$Y = \theta(I - L)((X - \mu)C^{-\frac{1}{2}}). \tag{7}$$

To break the limitation for the higher level graph embedding, like [19], instead of providing a predefined correlation embedding matrix, we introduce self attention mechanism to automatically compute a dynamic embedding matrix based on the representation similarity:

$$A_{i,j} = \frac{e^{\phi(h(x_i)) \bigotimes \psi(h(x_j))}}{\sum_{j=1}^{n} e^{\phi(h(x_i)) \bigotimes \psi(h(x_j))}}. \tag{8}$$

Here, $A_{i,j}$ is the correlation between node i and node j. The two projection functions $\phi(\cdot)$ and $\psi(\cdot)$ are used to map features to another feature space, where the Gaussian similarity is used to measure the node correlation strength.

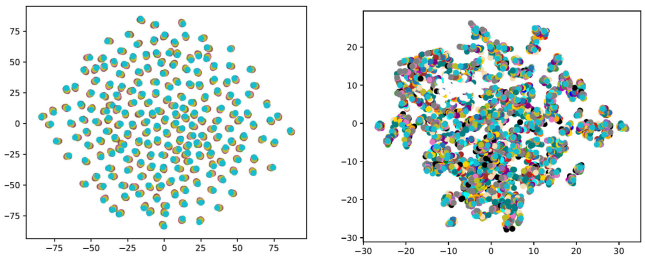
Based on the above structure, we construct four kinds of deconvolution models. First, the node-wise graph deconvolutional networks ST-GDN^2, which combines the mentioned graph model and a TCN together. To further benefit from the original GCN, we design a ST-GDCN, which replaces the graph model of ST-GDN^2 by a union of GCN and GDN. Both convolutional and deconvolutional feature embeddings are involved to promote the graph representation learning. Next, we extend deconvolution to frame-wise, in which we first rearrange the feature shape along the temporal dimension. So the feature representation $X \in R^{N \times T \times F}$ could be reshaped to $X \in R^{T \times N \times F}$. Then, the ST-GDN-T will be built by a normal GCN followed by a TDN, where the TDN is a temporal filter based on Eq. (7) and the matrix $(I - L)$ switches to an identity matrix. Finally, with above two models (ST-GDN^2 and ST-GDN-T), we combine GDN and TDN to build an element-wise ST-GDN, which is referred as ST-GDN-E.

III. Experiments

Here, we carry out comparative experiments on three skeleton datasets, i.e., NTU RGB+D [4], NTU RGB+D [25] and Kenitics-Skeleton [14], to evaluate our method.

A. Experiment Settings

Our model has seven graph neural layers. Like previous works [14], [18], [19], a residual skip connection is applied on graph convolutional block. The projection functions, as described in Eq. 8, are implemented by two channel-wise convolutional filters. The number of channels at each level are consistent with the current state-of-the-art methods [14], [18], [19] for fair comparison. The last output feature maps are averaged to a vector and then a fully connected layer is used for final class prediction. All the experiments are performed on PyTorch [26] and we train 50 epochs for our models with cross-entropy loss. A SGD with Nesterov momentum (0.9) is applied in the optimization algorithm. The weight decay is set to 0.0005. The learning rate is set as 0.1 and is decreased based on a cosine function. We execute five iterations for coupled Newton-Schulz.



(a) Feature with ST-GCN (b) Feature with ST-GDN^2

Fig. 2. Visualization of the features from ST-GCN and ST-GDN^2.

B. Visualization

We first visualize the features of each node at the last graph layer to observe whether this block could distinguish the embeddings. To this end, we compare ST-GCN network to the ST-GDN^2, which is designed from the node-wise, on NTU RGB+D dataset under the Cross-Subject (CS) evaluation.

After training these two networks for 50 epochs, we choose all the samples from the same class for evaluation. Since there are 25 nodes in each graph, we assign one different color to each node. We average the feature along the temporal dimension such that we get a 256-D representation for each node. Then we visualize it by using t-SNE [27]. Features from ST-GCN are shown in Fig. 2(a), while Fig. 2(b) presents for features obtained by ST-GDN^2. In Fig. 2(a), nodes from the same graph are nearly located at the same location since different nodes are with very similar feature representations. This is obviously an over-smoothing problem caused by GCN. On the contrary, as shown in Fig. 2(b), we can find very distinguishable representations for the nodes from ST-GDN^2, which proves that our method could alleviate this problem.

C. Ablation Experiments

Here, we evaluate the effectiveness of our method on the NTU RGB+D dataset under the CS evaluation. Here, current state-of-the-art 2S-AGCN [18] is utilized as the baseline. We also implement a seven-layer network, 2S-AGCN-7l, based on the block from 2s-AGCN. 2S-AGCN-7l is with the same architecture settings of ST-GDNs and is much smaller than 2S-AGCN. In this way, we want to know how well our model could perform when compared with GCN using the same setting. Finally, we combine all our ST-GDNs and expect to get a better graph network. Here, we empirically design our ST-GDN like this: for the first four layers, we put four ST-GDCN to capture richer representation of the input. For the fifth layer, we insert a node-wise deconvolution, ST-GDN^2. Inspired by [19], we set our ST-GDN with temporal-wise deconvolution and element-wise deconvolution at higher layers (layers six and seven) to enhance the importance of temporal information. In this way, we build a more powerful network (i.e., ST-GDN) for this task.

It can be seen from Table I that all our networks can achieve better results when compared to the baseline method 2S-AGCN. If we reduce 2S-AGCN to seven layers, which is the depth of our networks, our ST-GDN could even outperform it by 6.4% and 9.1% on joint and bone data, respectively. This shows the effectiveness of our method. Besides, results show that network benefits more from frame-level deconvolution (ST-GDN-T),

TABLE I
ABLATION EXPERIMENT

Methods	Joint(%)	Bone(%)
2S-AGCN [18]	86.6	85.8
2S-AGCN-7l	81.2	78.6
Ours(ST-GDN^2)	86.8	86.0
Ours(ST-GDCN)	86.7	86.1
Ours(ST-GDN-T)	87.2	87.0
Ours(ST-GDN-E)	86.8	86.5
Ours(ST-GDN)	87.6	87.7

TABLE II
COMPARISONS ON NTU RGB+D DATASET

Methods	CS(%)	CV(%)	Source
Dynamic Skeleton [28]	60.2	65.2	CVPR2015
P-LSTM [4]	62.9	70.3	CVPR2016
STA-LSTM [11]	73.4	81.2	AAAI2017
TCN [7]	74.3	83.1	CVPRW2017
VA-LSTM [12]	79.2	87.7	CVPR2017
Deep STGCK [13]	74.9	86.3	AAAI2018
ST-GCN [14]	81.5	88.3	AAAI2018
DPRL [29]	83.5	89.8	CVPR2018
SR-TSL [30]	84.8	92.4	ECCV2018
STGR-GCN [15]	86.9	92.3	AAAI2019
GR-GCN [16]	87.5	94.3	MM 2019
AS-GCN [17]	86.8	94.2	CVPR2019
2S-AGCN [18]	88.5	95.1	CVPR2019
NAS-GCN [19]	89.4	95.7	AAAI2020
Ours	**89.7**	**95.9**	

which is reasonable since over-smoothing caused by 300 frames could be more serious than that caused by 25 nodes. We could also find that directly adding deconvolution to all elements can not ensure the improvement when compared ST-GDN-E to ST-GDN-T. This also proves the architecture of ST-GDN is reasonable.

D. Comparison With the State-of-the-Art Methods

NTU RGB+D dataset Here, we compare with 14 state-of-the-art skeleton-based action recognition approaches under two evaluation metrics, i.e., CS and Cross-View (CV) metrics. All the comparison results are listed in Table II. In this task, like [18], [19], we build two stream networks and report the best result after performing the score-level fusion on joint and bone data. It can be seen from Table II that our model achieves the best performance in terms of either evaluation metrics. Specifically, our model gets the current best result 89.7% and 95.9% on CS and CV evaluations, respectively. Note that, our model even outperforms the NAS-based GCN method [19]. Besides, the model size of the proposed method also decreases by three times when compared with [19].

NTU RGB+D 120 dataset We compare with 14 skeleton-based action recognition approaches under CS and Cross-Setup (CST) evaluation metrics. Here, we report the best result on joint data. All the comparison results are listed in Table III. We can see from Table III that our model outperforms other compared approaches under both CS and CST metrics. When compare to the current best CNN-based method [31], GCN-based methods could get more than 10% improvements on average. That proves the graph convolutional networks are much suitable for this task. Comparison in the GCN-based methods could also show our superiority. For instance, when compared to the AS-GCN [17], which is the previous best model for this task, we can still get 3.4% and 3.1% improvements under the CST and CS evaluation metrics, respectively.

TABLE III
COMPARISONS ON NTU RGB+D 120 DATASET

Methods	CS (%)	CST (%)	Source
Dynamic Skeleton [28]	50.8	54.7	CVPR2015
P-LSTM [4]	25.5	26.3	CVPR2016
Spatio-Temporal LSTM [32]	55.7	57.9	ECCV2016
Internal Feature Fusion [33]	58.2	60.9	TIP2017
GCA-LSTM [34]	58.3	59.2	CVPR2017
MT Learning Network [35]	58.4	57.9	CVPR2017
Skeleton Visualization [9]	60.3	63.2	PR2017
2S Attention LSTM [36]	61.2	63.3	TIP2017
Soft RNN [37]	36.3	44.9	TPAMI2018
MT-CNN-RotClips [31]	62.2	61.8	TIP2018
Pose Evolution Map [38]	64.6	66.9	CVPR2018
ST-GCN [14]	72.4	71.3	AAAI2018
FSNet [39]	59.9	62.4	TPAMI2019
AS-GCN [17]	77.7	78.9	CVPR2019
Ours	**80.8**	**82.3**	

TABLE IV
COMPARISONS ON KINETICS-SKELETON DATASET

Methods	Top-1(%)	Top-5(%)	Source
Feature [40]	14.9	25.8	CVPR2015
P-LSTM [4]	16.4	35.3	CVPR2016
TCN [7]	20.3	40.0	CVPRW2017
ST-GCN [14]	30.7	52.8	AAAI2018
AS-GCN [17]	34.8	56.5	CVPR2019
2S-AGCN(Joint) [18]	35.1	57.1	CVPR2019
2S-AGCN(Bone) [18]	33.3	55.7	CVPR2019
2S-AGCN [18]	36.1	58.7	CVPR2019
NAS-GCN(Joint) [19]	35.5	57.9	AAAI2020
NAS-GCN(Bone) [19]	34.9	57.1	AAAI2020
NAS-GCN(Joint+Bone) [19]	37.1	60.1	AAAI2020
Ours(Joint)	**35.7**	**58.8**	
Ours(Bone)	**35.0**	**57.1**	
Ours(Joint+Bone)	**37.3**	**60.5**	

Kinetics-skeleton dataset We compare our method to seven different approaches. Like [18], [19], we report both top1 and top5 accuracy since this task is much challenging. All the comparison results are listed in Table IV.

It can be seen from Table IV that our model achieves the best performance on both of the metrics. Specifically, we get the best Top-1(37.3%) and Top-5(60.5%) performance on Kinetics-Skeleton dataset, which presents the score-level fusion results. For either using joint or bone data, we can always get the best results when compared with methods using the same data. Also, the size of the proposed model is smaller than the current best one [19] by three times.

IV. CONCLUSION

In this letter, we provide a novel and flexible spatial temporal graph deconvolutional network, ST-GDN, to address the graph over-smoothing issues in skeleton-based action recognition. The ST-GDN provides a new graph deconvolutional operation which not only performs a feature extraction but also provides a transformation of the graph representation such that it could be standardized. Based on this model, we build four different kinds of ST-GDNs and empirically insert them at the different levels of the networks. In this way, we construct our ST-GDN which could capture more powerful graph embeddings for the graph sequences. Compared to many state-of-the-art methods, the proposed model presents its efficiency and accuracy with corresponding metrics.

REFERENCES

[1] Y. Xu, J. Cheng, L. Wang, H. Xia, F. Liu, and D. Tao, "Ensemble one-dimensional convolution neural networks for skeleton-based action recognition," *IEEE Signal Process. Lett.*, vol. 25, no. 7, pp. 1044–1048, Jul. 2018.

[2] W. Peng, J. Shi, Z. Xia, and G. Zhao, "Mix dimension in poincaré geometry for 3D skeleton-based action recognition," in *Proc. 28th ACM Int. Conf. Multimedia*, 2020, pp. 1432–1440.

[3] X. Liu, H. Shi, X. Hong, H. Chen, D. Tao, and G. Zhao, "3D skeletal gesture recognition via hidden states exploration," *IEEE Trans. Image Process.*, vol. 29, pp. 4583–4597, 2020.

[4] A. Shahroudy, J. Liu, T.-T. Ng, and G. Wang, "NTU RGB+D: A large scale dataset for 3D human activity analysis," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 1010–1019.

[5] H. Tang, S. Bai, P. H. Torr, and N. Sebe, "Bipartite graph reasoning GANs for person image generation," in *Proc. 7th Brit. Mach. Vis. Conf.*, 2020, *arXiv:2008.04381*.

[6] W. Peng, X. Hong, and G. Zhao, "Video action recognition via neural architecture searching," in *Proc. IEEE Int. Conf. Image Process*., 2019, pp. 11–15.

[7] T. S. Kim and A. Reiter, "Interpretable 3D human action analysis with temporal convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.,* 2017, pp. 1623–1631.

[8] C. Li, Y. Hou, P. Wang, and W. Li, "Joint distance maps based action recognition with convolutional neural networks," *IEEE Signal Process. Lett.*, vol. 24, no. 5, pp. 624–628, May 2017.

[9] M. Liu, H. Liu, and C. Chen, "Enhanced skeleton visualization for view invariant human action recognition," *Pattern Recognit.*, vol. 68, pp. 346–362, 2017.

[10] Q. Ke, S. An, M. Bennamoun, F. Sohel, and F. Boussaid, "SkeletonNet: Mining deep part features for 3-D action recognition," *IEEE Signal Process. Lett.*, vol. 24, no. 6, pp. 731–735, Jun. 2017.

[11] S. Song, C. Lan, J. Xing, W. Zeng, and J. Liu, "An end-to-end spatio-temporal attention model for human action recognition from skeleton data," in *Proc. Thirty-First AAAI*, 2017, pp. 4263–4270.

[12] P. Zhang, C. Lan, J. Xing, W. Zeng, J. Xue, and N. Zheng, "View adaptive recurrent neural networks for high performance human action recognition from skeleton data," in *Proc. IEEE ICCV*, 2017, pp. 2117–2126.

[13] C. Li, Z. Cui, W. Zheng, C. Xu, and J. Yang, "Spatio-temporal graph convolution for skeleton based action recognition," in *Proc. Thirty-Second AAAI*, 2018.

[14] S. Yan, Y. Xiong, and D. Lin, "Spatial temporal graph convolutional networks for skeleton-based action recognition," in *Proc. Thirty-Second AAAI*, 2018, pp. 2122–2130.

[15] B. Li, X. Li, Z. Zhang, and F. Wu, "Spatio-temporal graph routing for skeleton-based action recognition," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, 2019, pp. 8561–8568.

[16] X. Gao, W. Hu, J. Tang, J. Liu, and Z. Guo, "Optimized skeleton-based action recognition via sparsified graph regression," in *Proc. ACM Int. Conf. Multimedia*, 2019, pp. 601–610.

[17] M. Li, S. Chen, X. Chen, Y. Zhang, Y. Wang, and Q. Tian, "Actional-structural graph convolutional networks for skeleton-based action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 3595–3603.

[18] L. Shi, Y. Zhang, J. Cheng, and H. Lu, "Two-stream adaptive graph convolutional networks for skeleton-based action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 12026–12035.

[19] W. Peng, X. Hong, H. Chen, and G. Zhao, "Learning graph convolutional network for skeleton-based human action recognition by neural searching," *Thirty-Fourth AAAI Conf. Artif. Intell.*, 2020.

[20] B. Zoph and Q. V. Le, "Neural architecture search with reinforcement learning," in *Proc. Int. Conf. Learn. Representations*, 2017.

[21] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. Int. Conf. Learn. Representations*, 2017.

[22] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," *Proc. NeurIPS*, 2016, pp. 3844–3852.

[23] D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun, and M. Paluri, "A closer look at spatiotemporal convolutions for action recognition," in *Proc. IEEE Conf. Compu. Vis. Pattern Recognit.*, 2018, pp. 6450–6459.

[24] C. Ye *et al.*, "Network deconvolution," in *Proc. Int. Conf. Learn. Representations*, 2020.

[25] J. Liu, A. Shahroudy, M. L. Perez, G. Wang, L.-Y. Duan, and A. K. Chichung, "NTU RGB+D 120: A large-scale benchmark for 3D human activity understanding," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 10, pp. 2684–2701, Oct. 2019.

[26] A. Paszke *et al.*, "Pytorch: An imperative style, high-performance deep learning library," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 8024–8035.

[27] L. V. D. Maaten and G. Hinton, "Visualizing data using T-SNE," *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, 2008.

[28] J.-F. Hu, W.-S. Zheng, J. Lai, and J. Zhang, "Jointly learning heterogeneous features for RGB-D activity recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 5344–5352.

[29] Y. Tang, Y. Tian, J. Lu, P. Li, and J. Zhou, "Deep progressive reinforcement learning for skeleton-based action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 5323–5332.

[30] C. Si, Y. Jing, W. Wang, L. Wang, and T. Tan, "Skeleton-based action recognition with spatial reasoning and temporal stack learning," in *Proc. ECCV*, 2018, pp. 103–118.

[31] Q. Ke, M. Bennamoun, S. An, F. Sohel, and F. Boussaid, "Learning clip representations for skeleton-based 3D action recognition," *IEEE Trans. Image Process.*, vol. 27, no. 6, pp. 2842–2855, Jun. 2018.

[32] J. Liu, A. Shahroudy, D. Xu, and G. Wang, "Spatio-temporal LSTM with trust gates for 3D human action recognition," in *Proc. Eur. Conf. Comput. Vis*., 2016, pp. 816–833.

[33] J. Liu, A. Shahroudy, D. Xu, A. C. Kot, and G. Wang, "Skeleton-based action recognition using spatio-temporal LSTM network with trust gates," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 12, pp. 3007–3021, Dec. 2018.

[34] J. Liu, G. Wang, P. Hu, L.-Y. Duan, and A. C. Kot, "Global context-aware attention LSTM networks for 3D action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 1647–1656.

[35] Q. Ke, M. Bennamoun, S. An, F. Sohel, and F. Boussaid, "A new representation of skeleton sequences for 3D action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 3288–3297.

[36] J. Liu, G. Wang, L.-Y. Duan, K. Abdiyeva, and A. C. Kot, "Skeleton-based human action recognition with global context-aware attention LSTM networks," *IEEE Trans. Image Process.*, vol. 27, no. 4, pp. 1586–1599, Apr. 2018.

[37] J.-F. Hu, W.-S. Zheng, L. Ma, G. Wang, J.-H. Lai, and J. Zhang, "Early action prediction by soft regression," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 11, pp. 2568–2583, Nov. 2019.

[38] M. Liu and J. Yuan, "Recognizing human actions as the evolution of pose estimation maps," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 1159–1168.

[39] J. Liu, A. Shahroudy, G. Wang, L.-Y. Duan, and A. K. Chichung, "Skeleton-based online action prediction using scale selection network," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 6,pp. 1453–1467, Jun. 2020.

[40] B. Fernando, E. Gavves, J. M. Oramas, A. Ghodrati, and T. Tuytelaars, "Modeling video evolution for action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 5378–5387.