Neurocomputing 454 (2021) 45–53

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Rethinking the ST-GCNs for 3D skeleton-based human action recognition

Wei Peng [a], Jingang Shi [b], Tuomas Varanka [a], Guoying Zhao [a,*]

[a] CMVS, University of Oulu, Oulu, Finland
[b] School of Software Engineering, Xi'an Jiaotong University, Xi'an, China

ARTICLE INFO

Article history:
Received 8 August 2020
Revised 18 January 2021
Accepted 4 May 2021
Available online 6 May 2021
Communicated by Zidong Wang

Keywords:
Human action recognition
ST-GCNs
Dynamic graph modeling
Deep neural networks

ABSTRACT

The skeletal data has been an alternative for the human action recognition task as it provides more compact and distinct information compared to the traditional RGB input. However, unlike the RGB input, the skeleton data lies in a non-Euclidean space that traditional deep learning methods are not able to use their fullest potential. Fortunately, with the emerging trend of Geometric deep learning, the spatial-temporal graph convolutional network (ST-GCN) has been proposed to deal with the action recognition problem from skeleton data. ST-GCN and its variants fit well with skeleton-based action recognition and are becoming the mainstream frameworks for this task. However, the efficiency and the performance of the task are hindered by either fixing the skeleton joint correlations or providing a computational expensive strategy to construct a dynamic topology for the skeleton. We argue that many of these operations are either unnecessary or even harmful for the task. By theoretically and experimentally analysing the state-of-the-art ST-GCNs, we provide a simple but efficient strategy to capture the global graph correlations and thus efficiently model the representation of the input graph sequences. Moreover, the global graph strategy also reduces the graph sequence into the Euclidean space, thus a multi-scale temporal filter is introduced to efficiently capture the dynamic information. With the method, we are not only able to better extract the graph correlations with much fewer parameters (only 12.6% of the current best), but we also achieve a superior performance. Extensive experiments on current largest 3D datasets, NTU-RGB+D and NTU-RGB+D 120, demonstrate the ability of our network to perform efficient and lightweight priority on this task.

1. Introduction

Human action recognition is a valuable but challenging topic which has attracted substantial attention from different research areas in recent years, since it provides significant insights into many valuable fields like action surveillance, human behavior analysis, pedestrian tracking, and robotics. Recently, skeleton data has become the mainstream input for action recognition. Compared to the traditional RGB video data, the skeleton data is more compact, leading to reduced computational costs, yet better performance can be seen when the background is complicated. Besides, skeleton data, as a simple kind of non-Euclidean data, also provides an appropriate way to investigate and analyze structured data, *e.g.*, graph topology (Fig. 1).

Here, we focus on 3D human action recognition task. Just like in other computer vision areas, deep neural networks [1] have become the main tools for this task. Generally, most previous studies have used deep models in the following two ways. Firstly, rearranging the structured data such that the resulted tensor is adapted to the fundamental neural networks, like the traditional recurrent neural networks (RNN) and convolutional neural networks (CNN). Representative works include [2–6]. The motivations are straightforward. In this way, the already well known and well developed network architectures can be easily utilized. Nonetheless, though substantial improvements have been seen in action recognition, the capability of deep learning is still constrained as there is no natural notion of locality in the skeleton data. Secondly, elaborating a customized neural network to adapt to the structured data. One of the most successful representative works in skeleton-based action recognition is spatial-temporal graph convolutional network, *i.e.*, ST-GCN [7]. To exploit the underlying structure of skeleton data, this work introduces Graph Convolutional Networks (GCN) [8,9] to model the graph representation of each skeleton and a succeeding temporal filter to capture the dynamic temporal information. ST-GCN and its variants have already achieved many encouraging results [10,7,11–14] and have become one of the most frequently used frameworks for the task. Work in [13] further improves the ST-GCN based on the observation that a

* Corresponding author.

https://doi.org/10.1016/j.neucom.2021.05.004
0925-2312/

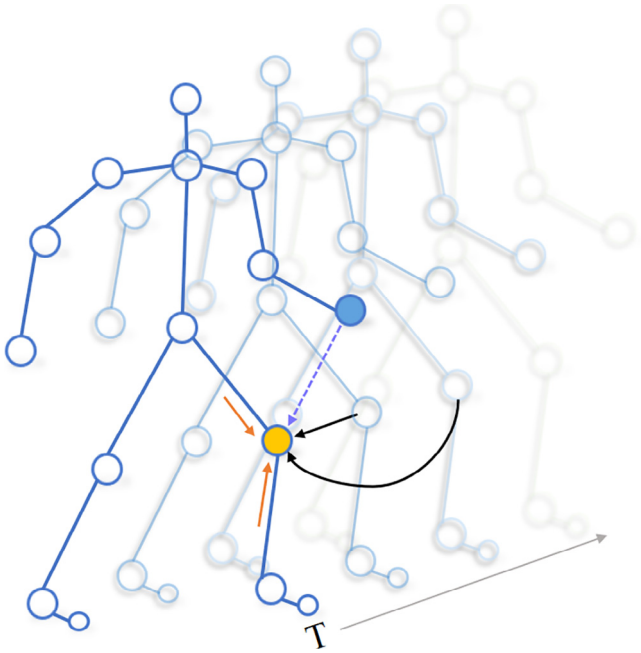


Fig. 1. Illustration of modeling skeleton with ST-GCNs. Like the yellow joint, every node in this skeleton will learn a feature representation via the message passing paradigm of ST-GCNs. At the same time temporal information(black arrow) is also learned by ST-GCNs. Different ST-GCNs provide different ways to involve the physical topology information(orange arrow) and semantic correlations(purple arrow). Thus, it is challenging to learn a distinguished node feature by message passing while at the same time keep its own characteristic feature. Here, we explore a simple but efficient way to extract graph representations.

pre-defined graph with fixed topology constraint ignores the implicit joint correlations. Therefore, by adding the fixed graph with an adaptive one based on the node similarity, the work of [13] involves multiple joint correlations into consideration. Peng et al. took this idea one step further by automatically providing a layer-wise dynamic matrix generation strategy with neural architecture searching [15]. In this work, they explored an automatic way to employ more correlation matrices, including higher-order connections and context-based dynamic matrices to improve the performance.

However, it spends so much computational resource to add these matrices. We doubt whether it is truly necessary to introduce such complicated strategies to learn the graph correlations. The following observations raise up our concern about these works. Firstly, the graph topology is not complicated since the graphs at different time steps share identical topology(ies), as the graphs are used to model the human body. Even there are hundreds of graphs in each sample, it is still reasonable and efficient to introduce this assumption. Besides, many of the ST-GCNs [15,13] in the mainstream follow this observation as well. Secondly, the GCNs are far from being efficient for small-scaled graph tasks. Generally, these methods will first find the neighbor to build a local perceptive field, then a neural network is constructed. However, for such a small graph, adaptively capturing the global graph information is much more efficient. Thirdly, the graph sequence can be treated as data lies in the Euclidean space from a global graph level. Each sample (graph sequence) lies in a non-Euclidean space only when observe from topology perspective. However, once the graph is processed from global level, it can be treated as a point. Thus the graph sequence will simply be a 1D vector lies in the Euclidean space. This will again let the network benefits from the well-developed neural network in Euclidean space.

Therefore, based on the previous observations, in this paper, we focus on reducing the manual efforts of building neural networks and the complexity of the architecture by presenting a brand new architecture for this task. This model, which is called Spatial-temporal Global Graph network, ST-GGN, totally discards the manual pre-defined topology constraint and is able to learn the global joint correlations automatically. In this work, a multi-head global graph representation learning strategy is provided to benefit from not only the graph correlations but also the efficiency of neural networks in Euclidean space. However, capturing global graph at each layer requires a very large filter with the same size as the number of nodes. We ingeniously avoid this by the proposed tensor rotation. Specifically, the dimension of the graph node is rotated to the channel dimension, thus a simple 1×1 filter can be used to capture the global information. Besides, we divide the temporal filters into several branches with different temporal perception fields such that the ST-GGN is able to capture richer temporal information, as well as further reduce the number of the model parameters. As a consequence, we provide a compact and efficient network for this task. To evaluate the proposed method, we perform comprehensive experiments on two large scale datasets, NTU RGB+D [2] and NTU RGB+D 120 [16]. Results show that our model is robust to subjects and view variations and achieves the state-of-the-art performance on both datasets. The contributions of this paper are presented as follows:

- We theoretically and experimentally analyze the most commonly used method, ST-GCN and its variants, for skeleton-based action recognition, and present a novel architecture for this task.
- Our method provides a simple but efficient way to capture global graph representation for small scale graph data, which does not require the pre-defined topology matrix and even make the method much more convenient. This method can be utilized as an alternative to the current GCNs.
- Extensive experiments are conducted on two current largest benchmarks. Comparison results not only show our superiority but also present its effectiveness since its model size is only 12.6% of the current best method, *i.e.*, NAS-GCN [15].

2. Related work

Skeleton-based Action Recognition. In human action recognition, skeleton data has increasingly attracted more attention due to its robustness against changes in body scales, viewpoints and backgrounds. Earlier approaches have focused on manually designing hand-crafted features and joint relationships, which are limited to lower-level information and the important semantic connectivity of the human body is ignored. To benefit from the great representation ability of deep learning, conventional methods tend to rearrange the skeleton data into grid-shaped structure and feed it directly into a classical RNN [2–4] or CNN [5,6] architectures. However, as mentioned in [17], the graph constructed by skeleton lies in a non-Euclidean space, which traditional CNNs and RNNs are not able to directly utilize since there is no natural locality information. Therefore, current works tend to use GCNs [7,10,44,45] since their operators are defined in the non-Euclidean space. Yan et al. and Li et al. are the first to use GCN for skeleton-based action recognition [7,10]. Shi et al. gave a two-stream GCN architecture, in which the joints and the second-order information (bones) are both used. With the help of Neural Architecture Search [18], Peng et al. provided an automatic way to design a GCN for this task [15]. With the layer-wise topology generation mechanism, it achieves the current best result. Our method can also be treated as variant of GCNs, but we explore a better way to capture the graph information with a more compact model.

Graph Convolutional Networks. Graph convolutional networks (GCNs), which generalize convolution to structured data,

has been successfully applied on many irregular data like social networks, and biological data. Generally, given a graph topology, previous works explored to propagate node features based on the spectral domain or the spatial domain. The spectral-domain methods [9,8] model the representation in the Fourier domain based on eigen-decomposition, which potentially leads to large computational costs. The spatial domain methods, also called Nodal-domain methods [17,19] directly implement operators on the graph node and its neighbors. However, it is difficult to model the global structure. To further improve the performance of GCN, attention mechanisms have also been introduced to GCN [19,20]. For instance, Velickovic et al. leveraged attention mechanism for graph node classification and achieved state-of-the-art performance [19]. Our ST-GGN can also be thought of as a variation to the spatial GCN, in which the multi-head global graph information is captured effortlessly by the proposed tensor rotation. Besides, our model is more compact and more efficient than other GCN methods.

3. Methodology

In this section, we give a detailed explanation of our spatial temporal global graph network, ST-GGN. To make the paper self-contained, we briefly review how to model a spatial graph with GCNs first. Then, a uniform formulation is given to describe the mainstream ST-GCNs for skeleton-based action recognition. Finally, we will introduce our method and provide the corresponding neural architecture.

3.1. GCN preliminaries

The input for skeleton-based action recognition is a graph sequence. Here, we define an undirected graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, A\}$ composed of $n = |\mathcal{V}|$ nodes, and $|\mathcal{E}|$ edges. The node connections are encoded in the adjacency matrix $A \in \mathcal{R}^{n\times n}$. The attribute of each node will be presented as follows. To make it simple, let $X \in \mathcal{R}^n$ be the input representation of $\mathcal{G}$ and $x_i \in X$ be the feature for its i-th node. In order to model the representation of $\mathcal{G}$, a Fourier transform is conducted on the graph so that the transformed signal, as in the Euclidean space, could then be dealt with formulation of fundamental operations such as filtering. To this end, a graph Laplacian L, of which the normalized definition is $L = I_n - D^{-1/2}AD^{-1/2}$ and $D_{ii} = \sum_j A_{ij}$, is used for Fourier transform. The procedure that a graph filtered by operator g_θ, parameterized by θ, can be formulated as

$$Y = g_\theta(L)X = Ug_\theta(\Lambda)U^TX, \tag{1}$$

where Y is the extracted graph feature. U is the Fourier basis which is a set of orthonormal eigenvectors for L so that $L = U\Lambda U^T$ with the Λ as its corresponding eigenvalues. However, this involves a matrix multiplication with the eigenvectors, which is computationally expensive [9]. Suggested by [21], a Chebyshev polynomial is a computational friendly strategy to approximate the filter.

$$Y = \sum_{r=0}^{K} \theta'_k T_k\left(\widehat{L}\right)X, \tag{2}$$

of which θ'_k denotes the Chebyshev coefficients. The Chebyshev polynomial $T_k\left(\widehat{L}\right)$ is recursively defined as

$$T_k\left(\widehat{L}\right) = 2\widehat{L}T_{k-1}\left(\widehat{L}\right) - T_{k-2}\left(\widehat{L}\right) \tag{3}$$

with $T_0 = 1$ and $T_1 = \widehat{L}$. Here $\widehat{L} = 2L/\lambda_{max} - I_n$ is normalized to [-1,1]. Following the work in [8], the parameters K and λ_{max} in Eq. (2) are set to 1 and 2, respectively. In this way, a first-order approximation of spectral graph convolutions is formed. Combining all,

$$Y = \theta'_0X + \theta'_1(L - I_n)X = \theta'_0X - \theta'_1\left(D^{-1/2}AD^{-1/2}\right)X. \tag{4}$$

Likewise, θ'_k can also be approximated with a unified parameter θ, which means $\theta = \theta_0 = -\theta_1$. All the approximation errors are expected to be adapted by the training process, then

$$Y = \theta\left(I_n + D^{-1/2}AD^{-1/2}\right)X. \tag{5}$$

The computational cost of Eq. (5) is much cheaper than Eq. (4). Thus, GCN built by Eq. (5) is widely utilized in the ST-GCNs. One can stack multiple GCN layers to get a high-level graph feature. In the aforementioned setting, the representation of the input x_i is simply treated as a 1D point. In practice, $X \in \mathcal{R}^{n\times C}$ is with multi-channels. In the following sections, we set $L = I_n + D^{-1/2}AD^{-1/2}$ with self-loop. Thus, Eq. (5) could be rewritten as:

$$Y = LX\theta. \tag{6}$$

3.2. Uniform formulation of the ST-GCNs

Though there are several variations of ST-GCNs for the skeleton-based action recognition, we can accordingly provide a uniform formulation for them. Here, we focus on three ST-GCN variants, including ST-GCN [7], 2S-AGCN [13], and NAS-GCN [15]. In fact, these works can be summarised as constructing different mathematical models to capture different joint correlations, including fixed topology correlations, underlying implicit node correlations, k-hop node correlations, and independent adaptive correlations. Although these methods are very different, the ideas can be summarized as a uniform formulation of constructing the correlation topologies L in Eq. (6) to help the message passing. That is

$$L = f(g_1(A), g_2(X)). \tag{7}$$

In Eq. (7), the function f provides different ways of combining multiple sets of topology information. Possible option can be summation with self-loops. g_1 and g_2 can be different ways to construct the topology matrices based on either adjacency matrix A or graph context X. The g_1 function is the operations on the pre-defined topology, which could simply utilize the topology information, involving with k-hop connections by Chebyshev polynomials, etc. The g_2 function automatically computes the topology matrices by the context information. Options can be Gaussian similarity function. Based on this uniform formulation, we give the analysis of the aforementioned three ST-GCNs.

3.2.1. ST-GCN

Spatial temporal Graph Convolutional Networks, ST-GCN [7], provides one of the most commonly used frameworks for skeleton-based action recognition. This method utilizes a simple way to construct the topology. Specifically,

$$L = A[0] + A[1] + A[2] \tag{8}$$

with f being the summation operation performed on the topology of the current node and the topologies of its two neighbors. Basically, they will also introduce self-loops with an identity matrix. We can also get that $g_1(A) = A$ and $g_2(X) = 0$. In this case, the authors manually build three different topologies. However, there are mainly two problems for these methods. First, they need extra labor to manually design these topologies. Second, different datasets have various encoding rules. Thus, an adaptive way is much preferred.

3.2.2. 2S-AGCN

Two-Stream Adaptive Graph Convolutional Networks, 2S-AGCN [13], is a variant of ST-GCN. Instead of only providing a fixed physical topology, they compute the semantic-level graph correlations at the same time. Besides, they also involve a learnable matrix with the size of A. Therefore, they totally introduce nine matrices, which means that

$$L = A + B + C, \tag{9}$$

where each matrix A, B, and C, consist of three sub matrices. Thus, f is also a summation function with self-loops. Then $g_1(A) = A + B_\theta(A)$, where $B_\theta(A)$ is a learnable matrix with the same size of A. And $g_2(X) = C$ for the entry $C_{i,j}$ of the C is

$$C_{i,j} = \frac{e^{\phi(h(x_i))\otimes\psi\left(h\left(x_j\right)\right)}}{\sum_{j=1}^{n} e^{\phi(h(x_i))\otimes\psi\left(h\left(x_j\right)\right)}}. \tag{10}$$

Here, $\otimes$ represents the matrix multiplication. $h(x_i)$ and $h(x_j)$ are the representation of node i and node j from X. Here, the authors also introduce two projection functions ϕ and ψ to map features to another feature space, where the Gaussian similarity can be used to measure the node correlation strength. In practice, the projection functions ϕ and ψ are implemented by the channel-wise filters. All of these will increase the computational complexity.

3.2.3. NAS-GCN

Peng et al. have provided an automatic way to build a GCN for this task, which is named as NAS-GCN [15]. This work not only provides an involvement of higher order connections, but also computes a layer-wise dynamic graph based on the semantic information. All of these matrices are determined by an automatic method, *i.e.,* NAS. Besides, they also involve a learnable matrix with the size of A. Here, we discuss the searched GCN architecture instead of the searching method. From the provided searched architecture, we know that they introduces different kinds of topology matrices for different layers. In total, there are four kinds, which means that

$$L = A^2 + A^4 + B + C. \tag{11}$$

This method involves more topology matrices compared to previous ST-GCNs. Here, f is also a summation function with self-loops. But $g_1(A) = A^2 + A^4$, which introduces k-hop connections by Chebyshev polynomials. Matrix B is the learnable matrix just like that in 2S-AGCN while $g_2(X) = C$, using different projection functions ϕ and ψ to compute the Gaussian similarity. These functions are implemented by spatial filters, temporal filters, and also spatial-temporal filters, which will again largely increase the computational complexity. In total, the method computes fifteen matrices for each layer.

3.2.4. Our global graph network

With the aforementioned information, we can observe that when GCNs involve more matrices to encode the topology information, hopefully they can get some benefits for the skeleton-based action recognition. However, we argue that it is unnecessary to introduce such complicated operations, *i.e.*, different g functions, for computing these matrices. Following the mainstream approaches of this field, all the graphs in each sample will share an identical set of topology matrices. Therefore, we have sufficient reasons to involve all the nodes in each graph to capture global graph information, since there are only 25 nodes in each graph. In this way, we do not need to manually design a physical topology graph. In addition, we do not need to utilize expensive projection functions to compute the Gaussian similarity.

Nevertheless, extracting node representations from 25 nodes is still not trivial. Employing a fully connected layer or introducing a filter with a kernel size of 25 is not a good option. Here, we provide an simple way to deal with this. We rotate a tensor, including input tensor and latent feature tensor, such that the node dimension is changed to the channel dimension. In this way, a simple 1×1 filter can be applied to capture the global graph information. Therefore, the function f in our method is also a summation function with self-loops. The $g_1(A) = 0$, since we do not need any pre-defined matrix. And $g_2(X) = \theta_R Rotation(X)$, which is used to capture the global graph information by using θ_R, a 1×1 filter after rotating the tensor. This simple operation avoids all the unnecessary and expensive projections. To involve more information, we add multi-heads into $g_2(X)$, just as shown in Fig. 2. After we extracting the global graph information for this module, we will rotate back the tensor, such that like the conventional network, the tensor will be learned by filters (here a convolutional filter with size of 1×1) in Euclidean space.

Another observation is that, in all of the previous ST-GCNs mentioned, the GCNs are succeeded with a temporal filter of the size 9×1. Here, we turn to a more sophisticated way to get more temporal information with fewer parameters. As illustrated in Fig. 2, the temporal filter in our model is called Multi-Scale Temporal filter, MST. The MST is inspired by work [23]. There are several branches in this filter. We uniformly group the inputs into each branch such that the model is able to capture richer temporal information with fewer parameters. For branches with the kernel size 3, we utilize different dilation settings such that they have different temporal windows to capture multi-scale information. Finally, all the outputs from different branches are concatenated together as the output from the temporal filter. Compared to the Multi-Scale Temporal convolutional filter (MS-TCN) in work [23], our MST is more compact since we remove the projection function before each branch (dilated filter). We will compare to this module in the next section.

4. Experiments

In this section, we carry out comparative experiments on two large-scale skeleton datasets, *i.e.*, NTU RGB+D [2] and NTU RGB+D 120 [16], to evaluate the performance of our model on the action recognition task. Firstly, we briefly describe the datasets and the corresponding experimental settings. Then, we provide an ablation study. Finally, we will give all the comparison results with the state-of-the-art methods and a corresponding analysis.

4.1. Datasets and metrics

NTU RGB+D. [2] is currently the most widely used indoor-captured action recognition dataset. There are four modalities, as shown in Fig. 3, in this dataset. Here, we focus on skeleton data, which contains a total of 56,880 3D skeleton video clips. The clips are captured from three cameras with different settings. All video clips contain a total of 60 human action classes including both single-actor and two-actor actions. The maximum frame number of a video clip is 300 and 25 joints coordinates are captured for each actor skeleton. The skeleton topology is illustrated at the second one of Fig. 3. Original work [2] suggests two benchmark evaluations, which are Cross-Subject (CS) and the Cross-View (CV) evaluations. In the CS evaluation, the dataset is divided into training and testing sets according to the subjects. The training set contains 40,320 videos from 20 subjects, and the rest 20 subjects with 16,560 video clips are used for testing. In the CV evaluation, dataset is divided by the camera ID number. The 37,920 videos captured from camera two and three are used in the training and the

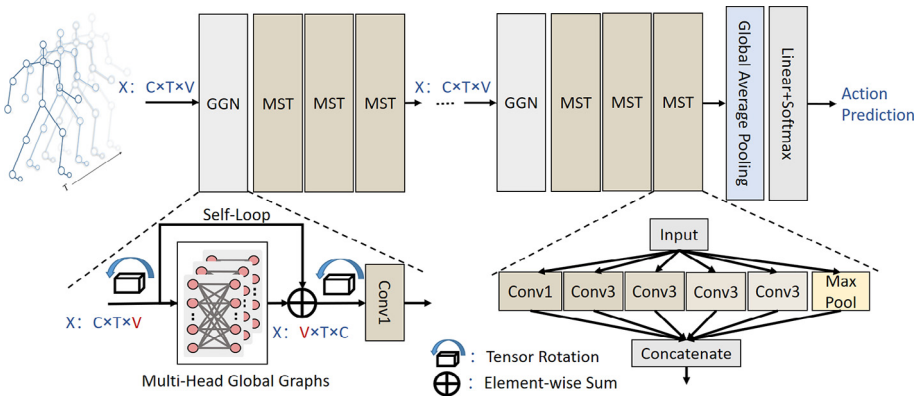


Fig. 2. Illustration of our framework. We stack several of our ST-GGN blocks to model the graph representation such that accurately providing an action prediction. For each of our ST-GGN block, there are mainly two parts, i.e, Global graph network, GGN, and the multi-scale temporal model, MST. Thus, the ST-GGN block is constructed by one GGN and three MSTs. In each GGN model, the core operation is the tensor rotation. By this way, we change the node dimension to the channel dimension, Then, the global graph information can be easily captured by utilizing a channel-wise filter. In order to involve more graph information, we introduce a multi-head strategy. Here, we keep all the V nodes. and a residual connection is employed to analogue to the self-loop in the spectral GCN. In the temporal filter, inspired by the Inception module [22], we adopt a multi-scale temporal filter via dividing the model into six branches, and each branch is assigned with convolutional filter $1 \times 1, 3 \times 1$ with different dilation sizes to capture multi-scale information, and also max pooling.

Fig. 3. Illustration of the NTU datasets [2]**.** There are four modalities, including RGB, skeleton (shows in RGB + skeleton for better visualization), depth, and IR modalities of a sample frame. Here, we only focus on skeleton data.

18,960 videos from camera one are used for testing. We report the Top-1 accuracy on both of the two benchmarks. For inputs with more than one stream, *e.g.*, bones, a score-level fusion result is reported.

NTU RGB+D 120. [16] is the extended version of the NTU RGB+D dataset and it is much bigger than previous one. Besides, the new dataset is much more challenging since it involves more subjects and action categories. There are a total of 114,480 samples captured from people in a wide range of age distribution. The subjects increase from the previous 40 to 106 distinct subjects. Besides, the action categories also increase from 60 to 120. But the skeleton topology is still the same. According to [16], this dataset also suggests two evaluation metrics. These two evaluation protocols are Cross-Subject (CS) and the Cross-Setup (CST). The former one splits subjects in half to training and testing parts while the Cross-Setup metrics divides the samples based on the camera setup IDs. There are 32 different camera set-ups with different camera height and distance and for each set-up, they give a set-up ID. Thus, the dataset is divided into training part with even setup IDs and odd setup camera IDs for testing (16 setups). Consistent with the benchmark method, we report Top-1 accuracy on the two benchmarks with only one stream. One can definitely improve the performance by introducing two input streams.

Kinetics-Skeleton. is estimated from the action dataset Kinetics [24]. There are approximately 300 000 video clips, covering 400 kinds of human actions and being collected from YouTube. The original Kinetics-Skeleton dataset is provided by Yan et al. in their ST-GCN [7]. They employed the open source toolbox OpenPose [25] to estimate the 2D joints location (18 coordinates) of each frame. This dataset is very noisy. For frames which contain more than two persons, only the top-2 persons are selected based on the average joint confidence. The released data pad every clips to 300 frames. We test on this noisy 2D dataset to evaluate the robustness of our model. During comparison, both the Top-1 and Top-5 recognition accuracies are reported since this task is much harder due to its great variety.

The skeleton in Fig. 3 is the pre-defined skeleton typologies for NTU datasets. It shows the common way to encode the skeleton for the two benchmarks, though there are different encoding strategies with different joint numbers for the skeletons, like for the Kinetics-Skeleton dataset. Note that the positions of joints in the topology is different from that in the data tensor. which means the neighbors in the tensor may not be their real neighbors in the topology. We do not introduce the manually defined topology information into our model. Instead, we let the network, ST-GGN, automatically capture the correlations by their context.

4.2. Experiment settings

4.2.1. Data preparation

We keep the data preprocessing consistent with the previous best methods [13,15] for fair comparison. All the skeleton sequences are unified to 300 frames with two actors for each frame. For the single-actor data, we add the second body by padding the values with zeros. Thus, there are a total of 600 graphs with 15000 nodes in each input graph sequence. For works using two-stream networks [15,13], we also compute the second-order information to train the second architecture, and report the score-level fusion results just as they do.

4.2.2. Model settings

The architecture is shown in Fig. 2. Our ST-GGN has three blocks for three learning stages. Each GGN block is followed by three multi-scale temporal filters (MSTs). For each MST, we set four

different kinds of dilations, which means it can capture temporal information with the temporal windows from three (when dilation is one) to nine (when dilation is four). Stacking three MSTs will make the maximum of temporal windows size to 27. We also reduce the video frames by a factor of two at the first and the second stages. This reduces the input from 300 frames to 150 and then to 75 frames. At the same time, we double the channels once we reduce the temporal length. This is just like CNN increases the channel numbers when reducing the resolutions. In our ST-GGN, the feature maps are learned from the original three to 96, and then it been doubled to 192 at the second stages. Then it doubled again at the third stages. Thus, after the third stage, we get 384 feature maps which encodes the entire 25 nodes. A succeeding fully connected layer and a softmax are utilized for the final classification. Like previous works [7,13,15], the last 384 feature maps are averaged to a 384 dimensional vector and then is used for the final class prediction. There are two tensor rotations in each of our block. We add a batch normalization layer after the second rotation since the feature will be changed again after the tensor rotated back if we add it after the first one. There are nine and fifteen topology matrices in 2S-AGCN and NAS-GCN, respectively. In fact, multiple topology matrices provide multiple different ways to aggregate the node features. Similarly, our multi-head GCN also provided different ways to propagate the information. Inspired by this, we empirically set the number of multi-head in GGN to thirteen. Even though, comparing to the two methods, we do not increase many parameters, which will be shown in the ablation experiment part.

4.2.3. Training details

All the experiments are performed on the PyTorch [26] deep learning framework. We conduct 70 epochs for the NTU RGB+D dataset while 80 epochs for NTU RGB+D 120 dataset since the latter one is larger than the former. During the training process, the cross-entropy loss is utilized to optimize the networks. A stochastic gradient descent (SGD) with Nesterov momentum (0.9) is applied in the optimization algorithm for the networks. The weight decay is set to 0.0005. The learning rate is set as 0.05 and is decreased by a factor of 0.2 when the training phase reaches at the 30th epoch, 50th epoch, and 60th epoch.

4.3. Ablation experiments

In this section, we will evaluate the effectiveness of our method on the NTU RGB+D dataset under the Cross Subject evaluation. Here, we compare with the three ST-GCNs [7,13,15], in terms of the model parameters (Params), and the accuracy (Acc.). There are mainly two parts in our ST-GGN, namely GGN and MST. We would like to investigate the contribution of each part. Thus, we replace our MST with identical temporal filters, which is a 2D convolutional filter with a size of 9×1. In this way, we prove the effectiveness of the temporal module. To verify the capability of the GGN, we replace it by the GCN module from 2S-AGCN [13] to test whether the performance will drop or increase.

The results are listed in Table 1. It can be seen that on the NTU RGB+D dataset under the Cross Subject evaluation, our method not only achieves the best result when compared with the mainstream ST-GCNs [7,13,15], as well largely reduces the model parameters. Specifically, we improve the accuracy by 6.7% when compared to ST-GCN. We also compared to the MS-TCN in work [23], since our MST module is inspired by this module. From Table 1, we can see that when compared to MS-TCN, our method achieves superior performance with a more compact model. Besides, the most encouraging result is that we can also outperform the current best method NAS-GCN by 0.8% with only 12.6% of its parameters, which sufficiently proves the effectiveness of our approach.

Table 1
Ablation Experiment. Here, the comparisons are preformed on NTU RGB+D under the CS evaluation.

Methods	Params(M)	Acc. (%)
ST-GCN [7]	3.141	81.5
2S-AGCN [13]	3.450	86.8
NAS-GCN [15]	6.567	87.4
Ours(GGN-T9)	1.982	84.6
Ours(2S-MST)	1.609	86.9
Ours(MS-TCN)	0.891	88.1
Ours	**0.829**	**88.2**

We also want to know the contribution of each part. Thus we replace the GGN and the MST with previous modules. Here, 2S-MST is built by replacing our GGN with GCN from 2S-AGCN. It can be seen from Table 1 that the accuracy drops about 1.3%, which proves the effectiveness of our GGN module. Here, we do not use the GCN from NAS-GCN since they provide different GCNs for different layers. To verify the effectiveness of the MST, we replace it with a temporal filter with a 9×1 kernel, which the temporal filter for the aforementioned three ST-GCNs, and build the GGN-T9 model. When compare it with our model, we can see the superiority since it drops the performance by 3.6% with two times of the parameters. All of these ablation studies prove the effectiveness of our method.

4.4. Comparison with the state-of-the-art methods

In this section, we will compare our ST-GGN with the State-of-the-art (SOTA) methods on two current most challenging 3D datasets, *i.e.*, NTU RGB+D [2], and the NTU RGB+D 120 [16].

NTU RGB+D dataset: We follow the original benchmark and evaluate on NTU RGB+D dataset under the CS and CV metrics. Here, we compare with 14 state-of-the-art skeleton-based action recognition approaches, including both handcrafted and deep learning methods. Specifically, the hand-crafted method is from [27]. Deep learning methods include reinforcement learning based method [28], CNN-based method [5], RNN-based methods [2–4,29,30] and current famous GCN-based methods [10,7,11,14,13,15]. All the comparison results are listed in Table 2. In this task, like [13,15], 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 2 that our model achieves the best performance in terms of both evaluation metrics. Specifically, our model gets the current best result 90.1% and 95.9% on CS and CV evaluations, respectively. Note that, when compared to the current best method, NAS-based GCN method [15], our model also outperforms it in accuracy while the model in [15] is about **eight times larger than our model**. All these results prove the effectiveness of our method.

NTU RGB+D 120 dataset: NTU RGB+D 120 dataset is more challenging than previous dataset since it involves more subjects and more action categories. On NTU RGB+D 120 dataset, we compare with 14 skeleton-based action recognition approaches under CS and CST evaluation metrics. The comparison methods include a hand-crafted method [27], conventional deep learning methods [2,31–34,6,35–39], and current commonly used GCN-based methods [7,14,40,41]. Here, like the original work [16], we report the best result on joint data. Definitely, one can further improve the current performance by using two stream architectures, just like for the NTU RGB+D dataset. All the comparison results are listed in Table 3.

We can see from Table 3 that our model outperforms most of the compared approaches under both CS and CST metrics. And it is able to get comparable results when compared to more

Table 2
Comparisons on NTU RGB+D with 14 state-of-the-art methods. Here, we report the results for both CS and CV metrics. It can be seen that our method can get the current best results under any given metrics.

Methods	CS(%)	CV(%)	FLOPs (G)	Source
Dynamic Skeleton [27]	60.2	65.2	-	CVPR2015
P-LSTM [2]	62.9	70.3	-	CVPR2016
STA-LSTM [3]	73.4	81.2	-	AAAI2017
TCN [5]	74.3	83.1	-	CVPRW2017
VA-LSTM [4]	79.2	87.7	-	CVPR2017
Deep STGCK [10]	74.9	86.3	-	AAAI2018
ST-GCN [7]	81.5	88.3	16.356	AAAI2018
DPRL [28]	83.5	89.8	-	CVPR2018
SR-TSL [29]	84.8	92.4	-	ECCV2018
STGR-GCN [11]	86.9	92.3	-	AAAI2019
AS-GCN [14]	86.8	94.2	35.532	CVPR2019
2S-AGCN [13]	88.5	95.1	37.224	CVPR2019
2sAGC-LSTM [30]	89.2	95.0	54.4	CVPR 2019
NAS-GCN [15]	89.4	95.7	108.82	AAAI2020
Ours	**90.1**	**95.9**	5.46	

advanced methods, like shift-GCN [40] and MixD [41]. Specifically, when compared to the current best CNN-based method [37], GCN-based methods are able to get more than 10% improvements on average, which tells the reason why the graph convolutional networks are much welcomed for this task. The comparison among the GCN-based methods, ST-GCN [7] and AS-GCN [14], also show the outstanding of our approach. For instance, when compared to the AS-GCN [14], which is the previous best model for this task, we can both get 2.8% improvements under the CS and CST evaluation metrics. Since the new comparison methods, shift-GCN [40] and MixD [41], introduce more advanced techniques, like hyperbolic space [42] and shift GCN, the classification accuracy is further improved. However, our method still gets a comparable even though not the best result, with much lighter computational cost. Besides, just as mentioned before, the number of parameters in our model is much less than the other ones. All of these proves the effectiveness of our method.

Kenetics-skeleton dataset: Here, to test the robustness of our model, we evaluate on the Kenetics-skeleton dataset. We compared to eight state-of-the-art skeleton-based action recognition approaches, including one hand-crafted method [43], two conventional deep methods [2,5], and four GCN-based methods [7,14,13,15]. All the comparison results are listed in Table 4. Like [13,15], we report the best results after performing the score-level fusion on joints and bones.

This task is much challenging since the skeleton data is estimated by third-party source, OpenPose [25] and there are much more classes (400 classes). It can be seen from Table 4 that our model achieves the Top-1(33.1%) and Top-5(55.2%) performance on Kinetics-Skeleton dataset, which presents the score-level fusion from joint and bone. We can get a comparable, but not best result when compared to the current best method [15]. However, our model is only one eighth of the current best model, which proves the effectiveness of our method.

Table 3
Comparisons on NTU RGB+D 120 with 14 state-of-the-art methods. Here, we report the results under the CST and CS evaluation metrics.

Methods	CS (%)	CST (%)	Source
Dynamic Skeleton [27]	50.8	54.7	CVPR2015
P-LSTM [2]	25.5	26.3	CVPR2016
Spatio-Temporal LSTM [31]	55.7	57.9	ECCV2016
Internal Feature Fusion [32]	58.2	60.9	TIP2017
GCA-LSTM [33]	58.3	59.2	CVPR2017
MT Learning Network [34]	58.4	57.9	CVPR2017
Skeleton Visualization [6]	60.3	63.2	PR2017
2S Attention LSTM [35]	61.2	63.3	TIP2017
Soft RNN [36]	36.3	44.9	TPAMI2018
MT-CNN-RotClips [37]	62.2	61.8	TIP2018
Pose Evolution Map [38]	64.6	66.9	CVPR2018
ST-GCN [7]	72.4	71.3	AAAI2018
FSNet [39]	59.9	62.4	TPAMI2019
AS-GCN [14]	77.7	78.9	CVPR2019
MixD [41]	80.5	83.2	ACM MM2020
Shift-GCN [40]	**80.9**	**83.2**	CVPR2020
Ours	80.5	81.7	

Table 4
Performance comparison on Kenetics-skeleton with the third-party estimated joint inputs. Note that the provided skeletons are 2D noisy ones. Here, we compare to eight state-of-the-art methods and report both the Top-1 and Top-5 results.

Methods	Top-1(%)	Top-5(%)	Source
Feature [43]	14.9	25.8	CVPR2015
P-LSTM [2]	16.4	35.3	CVPR2016
TCN [5]	20.3	40.0	CVPRW2017
ST-GCN [7]	30.7	52.8	AAAI2018
AS-GCN [14]	34.8	56.5	CVPR2019
2S-AGCN [13]	35.1	57.1	CVPR2019
STGR-GCN [11]	33.6	56.1	AAAI2019
NAS-GCN [15]	**35.5**	**57.9**	AAAI2020
Ours	33.1	55.2	-

5. Conclusions

In this paper, we present an efficient and compact spatial temporal global graph network, ST-GGN, to deal with the skeleton-based action recognition task. The ST-GGN ingeniously provides a global graph information capturing strategy by tensor rotation. With this strategy, the learning of the graph embedding does not need any pre-defined topology matrix. Besides, the graph information can be easily extracted in the Euclidean space. This not only improves the efficiency of the learning procedure of the graph topology, but also largely reduces the model parameters. To capture more temporal dynamic information with less parameters, we build multi-branch temporal filters with different time windows. We finally construct our ST-GGN model by combining these two parts. Extensive experiments are conducted to evaluate ST-GGN on current largest 3D skeleton-based action recognition datasets, including NTU RGB+D, and NTU RGB+D 120. Comparison results to many state-of-the-art methods proves its efficiency and presents superior performance in both accuracy and model size.

CRediT authorship contribution statement

Wei Peng: Conceptualization, Methodology, Software, Data curation, Investigation, Writing - original draft. **Jingang Shi:** Data

curation, Writing - review & editing. **Tuomas Varanka:** Writing - review & editing. **Guoying Zhao:** Supervision, Conceptualization, Writing - review & editing, Project administration, Funding acquisition.

Declaration of Competing Interest

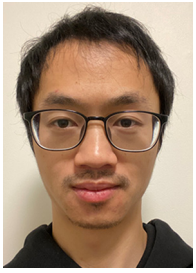
The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work was supported by Academy ICT 2023 project (grant 328115), the Academy of Finland for project MiGA (grant 316765), the National Natural Science Foundation of China under Grant 62002283, and Infotech Oulu. As well, the authors wish to acknowledge CSC-IT Center for Science, Finland, for computational resources.

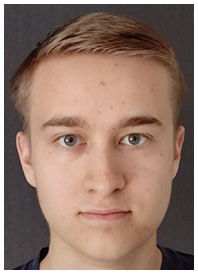
References

[1] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (7553) (2015) 436–444.
[2] A. Shahroudy, J. Liu, T.-T. Ng, G. Wang, Ntu rgb+ d: A large scale dataset for 3d human activity analysis, IEEE CVPR (2016) 1010–1019.
[3] S. Song, C. Lan, J. Xing, W. Zeng, J. Liu, An end-to-end spatio-temporal attention model for human action recognition from skeleton data, in: Thirty-first AAAI, 2017..
[4] P. Zhang, C. Lan, J. Xing, W. Zeng, J. Xue, N. Zheng, View adaptive recurrent neural networks for high performance human action recognition from skeleton data, IEEE ICCV (2017) 2117–2126.
[5] T.S. Kim, A. Reiter, InterpreTable 3d human action analysis with temporal convolutional networks, IEEE CVPR workshops 2017 (2017) 1623–1631.
[6] M. Liu, H. Liu, C. Chen, Enhanced skeleton visualization for view invariant human action recognition, Pattern Recogn. 68 (2017) 346–362.
[7] S. Yan, Y. Xiong, D. Lin, Spatial temporal graph convolutional networks for skeleton-based action recognition, in: Thirty-Second AAAI, 2018..
[8] T.N. Kipf, M. Welling, Semi-supervised classification with graph convolutional networks, arXiv (2016)..
[9] M. Defferrard, X. Bresson, P. Vandergheynst, Convolutional neural networks on graphs with fast localized spectral filtering, NeurIPS (2016) 3844–3852.
[10] C. Li, Z. Cui, W. Zheng, C. Xu, J. Yang, Spatio-temporal graph convolution for skeleton based action recognition, Thirty-Second AAAI, in, 2018.
[11] B. Li, X. Li, Z. Zhang, F. Wu, Spatio-temporal graph routing for skeleton-based action recognition (2019)..
[12] X. Gao, W. Hu, J. Tang, J. Liu, Z. Guo, Optimized skeleton-based action recognition via sparsified graph regression, in: Proceedings of the 2019 ACM International Conference on Multimedia, ACM, 2019.
[13] L. Shi, Y. Zhang, J. Cheng, H. Lu, Two-stream adaptive graph convolutional networks for skeleton-based action recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 12026–12035.
[14] M. Li, S. Chen, X. Chen, Y. Zhang, Y. Wang, Q. Tian, Actional-structural graph convolutional networks for skeleton-based action recognition, IEEE CVPR (2019).
[15] W. Peng, X. Hong, H. Chen, G. Zhao, Learning graph convolutional network for skeleton-based human action recognition by neural searching, The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI (2020)..
[16] J. Liu, A. Shahroudy, M.L. Perez, G. Wang, L.-Y. Duan, A.K. Chichung, Ntu rgb+ d 120: a large-scale benchmark for 3d human activity understanding, IEEE Trans. Pattern Anal. Mach. Intell. (2019).
[17] F. Monti, D. Boscaini, J. Masci, E. Rodola, J. Svoboda, M.M. Bronstein, Geometric deep learning on graphs and manifolds using mixture model cnns, IEEE CVPR (2017) 5115–5124.
[18] B. Zoph, Q.V. Le, Neural architecture search with reinforcement learning, 2016, arXiv preprint arXiv:1611.01578.
[19] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, Y. Bengio, Graph attention networks, ICLR (2018).
[20] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, NeurIPS (2017) 5998–6008.
[21] D.K. Hammond, P. Vandergheynst, R. Gribonval, Wavelets on graphs via spectral graph theory, Applied and Computational Harmonic Analysis 30 (2) (2011) 129–150.
[22] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 1–9.
[23] Z. Liu, H. Zhang, Z. Chen, Z. Wang, W. Ouyang, Disentangling and unifying graph convolutions for skeleton-based action recognition, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 143–152.
[24] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, M. Suleyman, A. Zisserman, The kinetics human action video dataset (2017). arXiv:1705.06950..
[25] Z. Cao, T. Simon, S.-E. Wei, Y. Sheikh, Realtime multi-person 2d pose estimation using part affinity fields, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Jul 2017.
[26] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al., Pytorch: An imperative style, high-performance deep learning library, Advances in Neural Information Processing Systems (2019) 8024–8035.
[27] J.-F. Hu, W.-S. Zheng, J. Lai, J. Zhang, Jointly learning heterogeneous features for rgb-d activity recognition, IEEE CVPR (2015) 5344–5352.
[28] Y. Tang, Y. Tian, J. Lu, P. Li, J. Zhou, Deep progressive reinforcement learning for skeleton-based action recognition, IEEE CVPR (2018) 5323–5332.
[29] C. Si, Y. Jing, W. Wang, L. Wang, T. Tan, Skeleton-based action recognition with spatial reasoning and temporal stack learning, ECCV (2018) 103–118.
[30] C. Si, W. Chen, W. Wang, L. Wang, T. Tan, An attention enhanced graph convolutional lstm network for skeleton-based action recognition, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2019, pp. 1227–1236.
[31] J. Liu, A. Shahroudy, D. Xu, G. Wang, Spatio-temporal lstm with trust gates for 3d human action recognition, European Conference on Computer Vision, Springer (2016) 816–833.
[32] J. Liu, A. Shahroudy, D. Xu, A.C. Kot, G. Wang, Skeleton-based action recognition using spatio-temporal lstm network with trust gates, IEEE Trans. Pattern Anal. Mach. Intell. 40 (12) (2017) 3007–3021.
[33] J. Liu, G. Wang, P. Hu, L.-Y. Duan, A.C. Kot, Global context-aware attention lstm networks for 3d action recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 1647–1656.
[34] Q. Ke, M. Bennamoun, S. An, F. Sohel, F. Boussaid, A new representation of skeleton sequences for 3d action recognition, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 3288–3297.
[35] J. Liu, G. Wang, L.-Y. Duan, K. Abdiyeva, A.C. Kot, Skeleton-based human action recognition with global context-aware attention lstm networks, IEEE Transactions on Image Processing 27 (4) (2017) 1586–1599.
[36] J.-F. Hu, W.-S. Zheng, L. Ma, G. Wang, J.-H. Lai, J. Zhang, Early action prediction by soft regression, IEEE Trans. Pattern Anal. Mach. Intell. (2018).
[37] Q. Ke, M. Bennamoun, S. An, F. Sohel, F. Boussaid, Learning clip representations for skeleton-based 3d action recognition, IEEE Trans. Image Process. 27 (6) (2018) 2842–2855.
[38] M. Liu, J. Yuan, Recognizing human actions as the evolution of pose estimation maps, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 1159–1168.
[39] J. Liu, A. Shahroudy, G. Wang, L.-Y. Duan, A.K. Chichung, Skeleton-based online action prediction using scale selection network, IEEE Trans. Pattern Anal. Mach. Intell. (2019).
[40] K. Cheng, Y. Zhang, X. He, W. Chen, J. Cheng, H. Lu, Skeleton-based action recognition with shift graph convolutional network, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2020.
[41] W. Peng, J. Shi, Z. Xia, G. Zhao, Mix Dimension in Poincaré Geometry for 3D Skeleton-Based Action Recognition, Association for Computing Machinery, New York, NY, USA, 2020, p. 1432–1440..
[42] W. Peng, T. Varanka, A. Mostafa, H. Shi, G. Zhao, Hyperbolic deep neural networks: A survey, arXiv preprint arXiv:2101.04562..
[43] B. Fernando, E. Gavves, J.M. Oramas, A. Ghodrati, T. Tuytelaars, Modeling video evolution for action recognition, IEEE CVPR (2015) 5378–5387.
[44] Wei Peng, Hong Xiaopeng, Zhao Guoying, Tripool: Graph triplet pooling for 3D skeleton-based action recognition, Patt. Recog. 115 (2021), https://doi.org/10.1016/j.patcog.2021.107921, In this issue.
[45] Wei Peng, Shi Jingang, Zhao Guoying, Spatial temporal graph deconvolutional network for skeleton-based human action recognition, IEEE Sig. Proc. Lett. 28 (2021) 244–248, https://doi.org/10.1109/LSP.2021.3049691.



Wei Peng received the M.S. degree in computer science from the Xiamen University, Xiamen, China, in 2016. He is currently a machine learning researcher and a Ph.D. candidate with the Center for Machine Vision and Signal Analysis, University of Oulu, Oulu, Finland. His articles have published in mainstream conferences and journals, such as the AAAI Conference on Artificial Intelligence (AAAI), IEEE International Conference on Computer Vision (ICCV), ACM Multimedia, IEEE Transactions on Image Processing, Pattern Recognition. His current research interests include machine learning, affective computing, medical imaging, and human action analysis.

Jingang Shi (Member, IEEE) received the B.S. and Ph.D. degrees from the Department of Electronics and Information Engineering, Xi'an Jiaotong University, China. From 2017 to 2020, he was a Post-Doctoral Researcher with the Center for Machine Vision and Signal Analysis, University of Oulu, Finland. Since 2020, he has been an Associate Professor with the School of Software, Xi'an Jiaotong University. His current research interests include image restoration, face analysis, and biomedical signal processing.



Tuomas Varanka received his B.S. and M.S. degree in computer science and engineering from the University of Oulu, Finland, in 2019 and 2020 respectively. He is currently pursuing his Ph.D. degree in University of Oulu. His work has focused on micro-expression recognition.

Guoying Zhao (IEEE Senior member 2012), is currently a Professor with the Center for Machine Vision and Signal Analysis, University of Oulu, Finland, where she has been a senior researcher since 2005 and an Associate Professor since 2014. She received the Ph.D. degree in computer science from the Chinese Academy of Sciences, Beijing, China, in 2005. She has authored or co-authored more than 240 papers in journals and conferences. Her papers have currently over 14250 citations in Google Scholar (h-index 54). She is co-program chair for ACM International Conference on Multimodal Interaction (ICMI 2021), was co-publicity chair for FG2018, General chair of 3rd International Conference on Biometric Engineering and Applications (ICBEA 2019), and Late Breaking Results Co-Chairs of 21st ACM International Conference on Multimodal Interaction (ICMI 2019), has served as area chairs for several conferences and is associate editor for Pattern Recognition, IEEE Transactions on Circuits and Systems for Video Technology, and Image and Vision Computing Journals. She has lectured tutorials at ICPR 2006, ICCV 2009, SCIA 2013 and FG 2018, authored/edited three books and nine special issues in journals. Dr. Zhao was a Co-Chair of many International Workshops at ICCV, CVPR, ECCV, ACCV and BMVC. Her current research interests include image and video descriptors, facial-expression and micro-expression recognition, emotional gesture analysis, affective computing, and biometrics. Her research has been reported by Finnish TV programs, newspapers and MIT Technology Review.