Semantics-Guided Neural Networks for Efficient Skeleton-Based Human Action Recognition

Pengfei Zhang [1]*, Cuiling Lan [2]†, Wenjun Zeng [2], Junliang Xing [3], Jianru Xue [1], Nanning Zheng [1]
[1] Xi'an Jiaotong University, Shaanxi, China [2] Microsoft Research Asia, Beijing, China
[3] National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing, China
zpengfei@stu.xjtu.edu.cn, {culan,wezeng}@microsoft.com, jlxing@nlpr.ia.ac.cn, {jrxue,nnzheng}@mail.xjtu.edu.cn

Abstract

Skeleton-based human action recognition has attracted great interest thanks to the easy accessibility of the human skeleton data. Recently, there is a trend of using very deep feedforward neural networks to model the 3D coordinates of joints without considering the computational efficiency. In this paper, we propose a simple yet effective semantics-guided neural network (SGN) for skeleton-based action recognition. We explicitly introduce the high level semantics of joints (joint type and frame index) into the network to enhance the feature representation capability. In addition, we exploit the relationship of joints hierarchically through two modules, i.e., a joint-level module for modeling the correlations of joints in the same frame and a frame-level module for modeling the dependencies of frames by taking the joints in the same frame as a whole. A strong baseline is proposed to facilitate the study of this field. With an order of magnitude smaller model size than most previous works, SGN achieves the state-of-the-art performance on the NTU60, NTU120, and SYSU datasets.

1. Introduction

Human action recognition has a wide range of application scenarios, such as human-computer interaction and video retrieval [35, 50, 1]. In recent years, skeleton-based action recognition [56, 7, 36, 58] is attracting increasing interests. Skeleton is a type of well structured data with each joint of the human body identified by a joint type, a frame index, and a 3D position. There are several advantages of using the skeleton for action recognition. First, skeleton is a high level representation of the human body with the human pose and motion abstracted. Biologically, human is able to recognize the action category by observing only the motion of joints even without appearance information [17]. Second, the advance of cost effective depth cameras [61] and pose estimation technology [38, 4, 43] make the access of skeleton much easier. Third, compared with RGB video, the skeleton representation is robust to variation of viewpoint and appearance. Fourth, it is also computationally efficient because of low dimensional representation. Besides, skeleton-based action recognition is also complementary to the RGB-based action recognition [42]. In this work, we focus on skeleton-based action recognition.

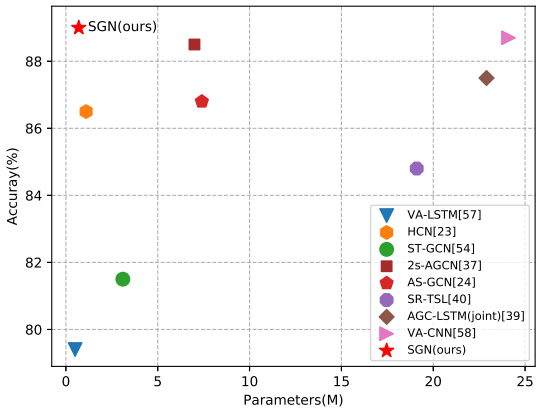


Figure 1: Comparisons of different methods on NTU60 (CS setting) in terms of accuracy and the number of parameters. The proposed SGN model achieves the best performance with an order of magnitude smaller model size.

For skeleton-based action recognition, deep learning is widely used to model the spatio-temporal evolution of the skeleton sequence [11, 47]. Various network structures have been exploited, such as Recurrent Neural Networks (RNN) [7, 63, 36, 41, 57, 40], Convolutional Neural Networks (CNN) [18, 58, 30, 51], and Graph Convolutional Networks (GCN) [54, 40, 44]. In the early years, RNN/LSTM was the favored network to be used to exploit the short and long term temporal dynamics. Recently, there is a trend of

*This work was done when P. Zhang was an intern at MSRA.
†Corresponding author.

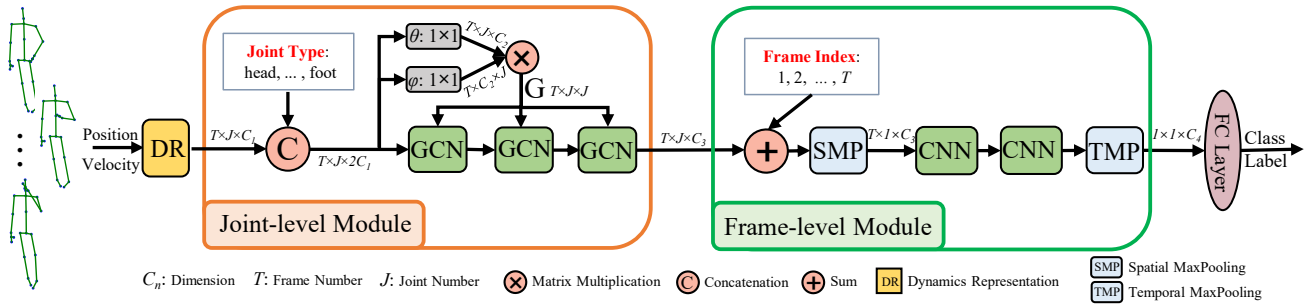


Figure 2: Framework of the proposed end-to-end Semantics-Guided Neural Network (SGN). It consists of a joint-level module and a frame-level module. In DR, we learn the dynamics representation of a joint by fusing the position and velocity information of a joint. Two types of semantics, *i.e.*, joint type and frame index, are incorporated into the joint-level module and the frame-level module, respectively. To model the dependencies of joints in the joint-level module, we use three GCN layers. To model the dependencies of frames, we use two CNN layers.

using feedforward (*i.e.*, non-recurrent) convolutional neural networks for modeling sequences in speech, language [34, 10, 53, 48], and skeleton [18, 58, 30, 51] due to their superior performance. Most skeleton-based approaches organize the coordinates of joints to a 2D map and resize the map to a size (*e.g.* 224×224) suitable for the input of a CNN (*e.g.* ResNet50 [12]). Its rows/columns correspond to the different types of joints/frames indexes. In these methods [18, 58, 30, 51], long-term dependencies and semantic information are expected to be captured by the large receptive fields of deep networks. This appears to be brutal and typically results in high model complexity.

Intuitively, semantic information, *i.e.*, the joint type and the frame index, is very important for action recognition. Semantics together with dynamics (*i.e.*, 3D coordinates) reveal the spatial and temporal configuration/structure of human body joints. As we know, two joints of the same coordinates but different semantics would deliver very different information. For example, for a joint above the head, if this joint is a hand joint, the action is likely to be *raising hand*; if it is a foot joint, the action may be *kicking a leg*. Besides, the temporal information is also important for action recognition. Taking the two actions of *sitting down* and *standing up* as examples, they are different only in occurrence order of the frames. However, most approaches [11, 47] overlook the importance of the semantic information and underexplore it.

To address the above mentioned limitations of current approaches, we propose a semantics-guided neural network (SGN) which explicitly exploits the semantics and dynamics for high efficient skeleton-based action recognition. Fig. 2 shows the overall framework. We build a hierarchical network by sequentially exploring the joint-level and frame-level dependencies of the skeleton sequence. For better joint-level correlation modeling, besides the dynamics, we incorporate the semantics of joint type (*e.g.*, 'head', and 'hip') to the GCN layers which enables the content adaptive graph construction and effective message passing among joints within each frame. For better frame-level correlation modeling, we incorporate the semantics of temporal frame index to the network. Particularly, we perform a Spatial MaxPooling (SMP) operation over all the features of the joints within the same frame to obtain frame-level feature representation. Combined with the embedded frame index information, two temporal convolutional neural network layers are used to learn feature representations for classification. In addition, we develop a strong baseline which is of high performance and efficiency. Thanks to the efficient exploration of semantic information, the hierarchical modeling, and the strong baseline, our proposed SGN achieves the state-of-the-art performance with a much smaller number of parameters.

We summarize our three main contributions as follows:

- We propose to explicitly explore the joint semantics (frame index and joint type) for efficient skeleton-based action recognition. Previous works overlook the importance of semantics and rely on deep networks with high complexity for action recognition.
- We present a semantics-guided neural network (SGN) to exploit the spatial and temporal correlations at joint-level and frame-level hierarchically.
- We develop a lightweight strong baseline, which is more powerful than most previous methods. We hope the strong baseline will be helpful for the study of skeleton-based action recognition.

With the above technical contributions, we have obtained a high performance skeleton-based action recognition model with high computational efficiency. Extensive ablation studies demonstrate the effectiveness of the proposed model design. On the three largest benchmark

datasets for skeleton-based action recognition, the proposed model consistently achieves superior performances over many competing algorithms while having an order of magnitude smaller model size than many algorithms (see Fig. 1).

2. Related Work

Skeleton-based action recognition has attracted increasing attentions recently. Recent works using neural networks [11] have significantly outperformed traditional approaches that use hand-crafted features [11, 52, 46, 55, 9].

Recurrent Neural Network based. Recurrent neural networks, such as LSTM [14] and GRU [5], are often used to model the temporal dynamics of skeleton sequence [7, 36, 63, 41, 57, 59, 60]. The 3D coordinates of all joints in a frame are concatenated in some order to be the input vector of a time slot. They do not explicitly tell the networks which dimensions belong to which joint. Some other RNN-based works tend to design special structures in RNN to make it aware of the spatial structural information. Shahroudy *et al.* divide the cell of LSTM into five sub cells corresponding to five body parts, *i.e.*, torso, two arms, and two legs, respectively [36]. Liu *et al.* propose a spatial-temporal LSTM model to exploit the contextual dependency of joints in both the temporal and spatial domain [27], where they feed different types of joints at each step. To some extent, they distinguish the different joints.

Convolutional Neural Network based. In recent years, in the field of speech, language sequence modeling, convolutional neural networks demonstrate their superiority in both accuracy and parallelism [34, 10, 53, 48, 45]. The same is true for skeleton-based action recognition [6, 22, 18, 3]. These CNN-based works transform the skeleton sequence to skeleton map of some target size and then use a popular network, such as ResNet [12], to explore the spatial and temporal dynamics. Some works transform a skeleton sequence to an image by treating the joint coordinate (*x,y,z*) as the R, G, and B channels of a pixel [6, 22]. Ke *et al.* transform the skeleton sequence to four 2D arrays, which are represented by the relative position between four selected reference joints (*i.e.*, the left/right shoulder, the left/right hip) and other joints [18]. Skeleton is well structured data with explicit high level semantics, *i.e.*, frame index and joint type. However, the kernels/filters of CNNs are translation invariant [32] and thus cannot directly perceive the semantics from such input skeleton maps. The CNNs are expected to be aware of such semantics through large receptive fields of deep networks, which is not very efficient.

Graph Convolutional Network based. Graph convolutional networks [21], which have been proven to be effective for processing structured data, have also been used to model the structured skeleton data. Yan *et al.* propose a spatial and temporal graph convolutional network [54]. They treat each joint as a node of the graph. The presence of edge denoting the joint relationship is pre-defined by human based on prior knowledge. To enhance the predefined graph, Tang *et al.* define the edges for both physically disconnected and connected joint pairs for better constructing the graph [44]. A SR-TSL model [40] is proposed to learn the graph edge of five human body parts within each frame using a data-driven method instead of leveraging human definition. A two-stream GCN model [37] learns a content adaptive graph based on the non-local block and uses it to pass messages in GCN layers. However, the informative semantics is not utilized for learning the graph edge and message passing of GCN, which makes the network less efficient.

Explicit Exploration of Semantics Information. The explicit exploration of semantics has been exploited in other fields, *e.g.*, machine translation [45] and image recognition [62]. Ashish *et al.* explicitly encode the position of the tokens in the sequence to make use of the order of the sequence in machine translation tasks [45]. Zheng *et al.* encode the group index into convolutional channel representation to preserve the information of group order [62]. For skeleton-based action recognition, however, the joint type and frame index semantics are overlooked even though such information is very important. In our work, we propose to explicitly encode the joint type and frame index to preserve the important information of the spatial and temporal body structure. As an initial attempt to explore such semantics, we hope it will inspire more investigation and exploration in the community.

3. Semantics-Guided Neural Networks

For a skeleton sequence, we identify a joint by its semantics (joint type and frame index) and represent it together with its dynamics (position/3D coordinates and velocity). Without semantics, the skeleton data will lose the important spatial and temporal structure. Previous CNN-based works [18, 6, 58], however, typically overlook the semantics by implicitly hiding them in the 2D skeleton map (*e.g.* with rows corresponding to the different types of joints and columns corresponding to the frame indexes).

We propose a semantics-guided neural network (SGN) for skeleton-based action recognition and show the overall end-to-end framework in Fig. 2. It consists of a joint-level module and a frame-level module. We describe the details of the framework in the following subsections.

Specifically, for a skeleton sequence, we denote all the joints as a set $\mathcal{S} = \{X_t^k \mid t = 1, 2, \ldots, T; k = 1, 2, \ldots, J\}$, where X_t^k denotes the joint of type k at time t. T denotes the number of frames of the skeleton sequence and J denotes the total number of joints of a human body in a frame. For a given joint X_t^k of type k at time t, it can be identified by its dynamics and semantics. Dynamics are related to the 3D position of a joint. Semantics means the frame index t and joint type k.

3.1. Dynamics Representation

For a given joint X_t^k, we define its dynamics by the position $\mathbf{p}_{t,k} = (x_{t,k}, y_{t,k}, z_{t,k})^T \in \mathbb{R}^3$ in the 3D coordinate system, and the velocity $\mathbf{v}_{t,k} = \mathbf{p}_{t,k} - \mathbf{p}_{t-1,k}$. We encode/embed the position and velocity into the same high dimensional space, *i.e.*, $\widetilde{\mathbf{p}_{t,k}}$ and $\widetilde{\mathbf{v}_{t,k}}$, respectively, and fuse them together by summation as

$$\mathbf{z}_{t,k} = \widetilde{\mathbf{p}_{t,k}} + \widetilde{\mathbf{v}_{t,k}} \in \mathbb{R}^{C_1}, \tag{1}$$

where C_1 is the dimension of the joint representation. Take the embedding of position as an example, we encode the position $\mathbf{p}_{t,k}$ using two fully connected (FC) layers as

$$\widetilde{\mathbf{p}_{t,k}} = \sigma(W_2(\sigma(W_1\mathbf{p}_{t,k} + \mathbf{b}_1)) + \mathbf{b}_2), \tag{2}$$

where $W_1 \in \mathbb{R}^{C_1 \times 3}$ and $W_2 \in \mathbb{R}^{C_1 \times C_1}$ are weight matrices, $\mathbf{b}_1$ and $\mathbf{b}_2$ are the bias vectors, σ denotes the ReLU activation function [33]. Similarly, we obtain the embedding for velocity as $\widetilde{\mathbf{v}_{t,k}}$.

3.2. Joint-level Module

We design a joint-level module to exploit the correlations of joints in the same frame. We adopt graph convolutional networks (GCN) to explore the correlations for the structural skeleton data. Some previous GCN-based approaches take the joints as nodes and they pre-define the graph connections (edges) based on prior knowledge [54] or learn a content adaptive graph [37]. We also learn a content adaptive graph, but differently we incorporate the semantics of joint type to the GCN layers for more effective learning.

We enhance the power of GCN layers by making full use of the semantics from two aspects. First, we use the semantics of joint type and the dynamics to learn the graph connections among the nodes (different joints) within a frame. The joint type information is helpful for learning suitable adjacent matrix (*i.e.*, relations between joints in terms of connecting weights). Take two source joints, *foot* and *hand*, and a target joint *head* as an example, intuitively, the connection weight value from *foot* to *head* should be different from the value from *hand* to *head* even when the dynamics of *foot* and *hand* are the same. Second, as part of the information of a joint, the semantics of joint types takes part in the message passing process in GCN layers.

We denote the type of the k^{th} joint (also referred to as type k) by a one-hot vector $\mathbf{j}_k \in \mathbb{R}^{d_j}$, where the k^{th} dimension is one and the others are all zeros. Similar to the encoding of position as in Equ. (2), we obtain the embedding of the k^{th} joint type as $\widetilde{\mathbf{j}_k} \in \mathbb{R}^{C_1}$.

Given J joints of a skeleton frame, we build a graph of J nodes. We denote the joint representation of joint type k at frame t with both the dynamics and the semantics of joint type as $\mathbf{z}_{t,k} = [\mathbf{z}_{t,k}, \widetilde{\mathbf{j}_k}] \in \mathbb{R}^{2C_1}$. All the joints of frame t are then represented by $Z_t = (\mathbf{z}_{t,1}; \cdots; \mathbf{z}_{t,J}) \in \mathbb{R}^{J \times 2C_1}$.

Similar to [49, 48, 37], the edge weight from the i^{th} joint to the j^{th} joint in the same frame t is modeled by their similarity/affinity in the embeded space as

$$S_t(i,j) = \theta(\mathbf{z}_{t,i})^T \phi(\mathbf{z}_{t,j}), \tag{3}$$

where θ and ϕ denote two transformation functions, each implemented by an FC layer, *i.e.*, $\theta(\mathbf{x}) = W_3\mathbf{x} + \mathbf{b}_3 \in \mathbb{R}^{C_2}$ and $\phi(\mathbf{x}) = W_4\mathbf{x} + \mathbf{b}_4 \in \mathbb{R}^{C_2}$.

By computing the affinities of all the joint pairs in the same frame based on (3), we obtain the adjacency matrix $S_t \in \mathbb{J} \times \mathbb{J}$. Normalization using SoftMax as [45, 48] is performed on each row of S_t so that the sum of all the edge values connected to a target node is 1. We denote the normalized adjacency matrix by G_t. A residual graph convolution layer is used to realize the massage passing among nodes as

$$\begin{aligned} Y_t &= G_t Z_t W_y, \\ Z'_t &= Y_t + Z_t W_z, \end{aligned} \tag{4}$$

where W_y and W_z are transformation matrices. The weight matrices are shared for different temporal frames. Z'_t is the output. Note that one can stack multiple residual graph convolution layers to enable further message passing among nodes with the same adjacency matrix G_t.

3.3. Frame-level Module

We design a frame-level module to exploit the correlations across frames. To make the network know the order of frames, we incorporate the semantics of frame index to enhance the representation capability of a frame.

We denote the frame index by a one-hot vector $\mathbf{f}_t \in \mathbb{R}^{d_f}$. Similar to the encoding of position as in Equ. (2), we obtain the embedding of the frame index as $\widetilde{\mathbf{f}_t} \in \mathbb{R}^{C_3}$. We denote the joint representation corresponding to joint type k at frame t with both the semantics of frame index and the learned feature as $\mathbf{z}'_{t,k} = \mathbf{z}'_{t,k} + \widetilde{\mathbf{f}_t} \in \mathbb{R}^{C_3}$, where $\mathbf{z}'_{t,k} = Z'_t(k,:)$.

To merge the information of all joints in a frame, we apply one spatial MaxPooling layer to aggregate them across the joints. The dimension of feature of the sequence is thus $T \times 1 \times C_3$. Two CNN layers are applied. The first CNN layer is a temporal convolution layer to model the dependencies of frames. The second CNN layer is used to enhance the representation capability of learned features by mapping it to a high dimension space with kernel size of 1. After the two CNN layers, we apply a temporal MaxPooling layer to aggregate the information of all frames and obtain the sequence level feature representation of C_4 dimensions. This is then followed by a fully connected layer with Softmax to perform the classification.

4. Experiments

4.1. Datasets

NTU60 RGB+D Dataset (NTU60) [36]. This dataset is collected by the Kinect camera for 3D action recognition with 56,880 skeleton sequences. It contains 60 action classes performed by 40 different subjects. Each human skeleton is represented by 25 joints with 3D coordinates ($J = 25$). For the Cross Subject (CS) setting [36], half of the 40 subjects are used for training and the rest for testing. For the Cross-View (CV) setting [36], the sequences captured by two of the three cameras are used for training and those captured by the other camera are used for testing. Following [36], we randomly select 10% of the training sequences for validation for both the CS and CV settings.

NTU120 RGB+D Dataset (NTU120) [25]. This dataset is an extension of NTU60. It is the largest RGB+D dataset for 3D action recognition with 114,480 skeleton sequences. It contains 120 action classes performed by 106 distinct human subjects. For the Cross Subject (C-Subject) setting, half of the 106 subjects are used for training and the rest for testing. For the Cross Setup (C-Setup) setting, half of the setups are used for training and the rest for testing.

SYSU 3D Human-Object Interaction Dataset (SYSU) [15]. It contains 480 skeleton sequences of 12 actions performed by 40 different subjects. Each human skeleton has 20 joints ($J = 20$). We use the same evaluation protocols as [15]. For the Cross Subject (CS) setting, half of the subjects are used for training and the rest for testing. For the Same Subject (SS) setting, half of the samples of each activity are used for training and the rest for testing. We use the 30-fold cross-validation and show the mean accuracy for each setting [15].

4.2. Implementation Details

Network Setting. To obtain the dynamic representation (DR), the number of neurons is set to 64 for each FC layer (*i.e.*, $C_1 = 64$). Note that the weights of FC layers are not shared for position and velocity. To encode the joint type, the number of neurons of the two FC layers are both set to 64. To encode the frame index, the numbers of neurons of the two FC layers are set to 64 and 256, respectively and $C_3 = 256$. For the transformation functions in (3), the number of neuron of each FC layer is set to 256, *i.e.*, $C_2 = 256$. For the joint-level module, we set the numbers of neurons of the three GCN layers to 128, 256, and 256, respectively. For the fame-level module, we set the number of neurons of the first CNN layer to 256 with kernel size of 3 along the temporal dimension, and set the number of neurons of the second CNN layer to 512 with kernel size of 1 (*i.e.*, $C_4 = 512$). After each GCN or CNN layer, batch normalization [16] and ReLU nonlinear activation function are used.

Training. All experiments are conducted on the Pytorch platform with one P100 GPU card. We use the Adam [20] optimizer with the initial learning rate of 0.001. The learning rate decays by a factor of 10 at the 60^{th} epoch, the 90^{th} epoch, and the 110^{th} epoch, respectively. The training is finished at the 120^{th} epoch. We use a weight decay of 0.0001. The batch sizes for NTU60, NTU120, and SYSU datasets are set to 64, 64 and 16, respectively. Label smoothing [13] is utilized for all experiments and we set the smoothing factor to 0.1. Cross entropy loss for classification is used to train the networks.

Data Processing. Similar to [57], sequence level translation based on the first frame is performed to be invariant to the initial positions. If one frame contains two persons, we split the frame into two frames by making each frame contain one human skeleton. During training, according to [27], we segment the entire skeleton sequence into 20 clips equally, and randomly select one frame from each clip to have a new sequence of 20 frames. During testing, similar to [2], we randomly create 5 new sequences in the similar manner and the mean score is used to predict the class.

During training, we perform data argumentation by randomly rotating the 3D skeletons to some degrees at sequence level to be robust to the view variation. For the NTU60 (CS setting), NTU120, and SYSU datasets, we randomly select three degrees (around X, Y, Z axes, respectively) between $[-17^\circ, 17^\circ]$ for one sequence. Considering that the large view variation for NTU60 (CV setting), we randomly select three degrees between $[-30^\circ, 30^\circ]$.

4.3. Ablation Study

4.3.1 Effectiveness of Exploiting Semantics

Semantics contains the important structural information of a skeleton sequence which is important for skeleton-based action recognition. To demonstrate the effectiveness of exploiting semantics, by referencing our framework (see Fig. 2), we build eight neural networks and perform various experiments on the NTU60 dataset. Table 1 shows the comparisons. In the following, *JT* denotes the semantics of joint type, *FI* denotes the semantics of frame index, *G* denotes the learning of graph (adjacency matrix), *P* denotes the graph convolutional operations which enable the massage passing. *T-Conv* denotes the temporal convolutional layer, *i.e.*, the first CNN layer of the frame-level module. Three GCN layers and two CNN layers are used in the joint-level (**JL**) module and the frame-level (**FL**) module, respectively. *w* and *w/o* denote "with" and "without", respectively.

Effectiveness of Exploiting Joint Type. We investigate four designed models (rows 1 to 4 in Table 1) to validate the effectiveness of the joint type on the joint-level module (**JL**) and all the four models do not include the semantics of temporal index. We explain one model here, and the other three models can be understood in a similar way. "JL(G w/o

Table 1: Effectiveness of exploiting semantics in the joint-level module (JL) and frame-level module (FL) on the NTU60 dataset in terms of accuracy (%). JT denotes joint type and FI denotes frame index.

Method	#Params(M)	CS	CV
JL(G w/o JT & P w/o JT) & FL	0.62	86.9	92.8
JL(G w JT & P w/o JT) & FL	0.66	87.5	93.7
JL(G w/o JT & P w JT) & FL	0.64	88.6	94.1
JL(G w JT & P w JT) & FL	0.67	**88.7**	**94.1**
JL & FL(w/o T-Conv) w/o FI	0.54	86.8	92.8
JL & FL(w/o T-Conv) w FI	0.56	**87.8**	**93.7**
JL & FL(w T-Conv) w/o FI	0.67	88.7	94.1
JL & FL(w T-Conv) w FI	0.69	**89.0**	**94.5**

JT & P w/o JT) & FL" denotes the scheme in which the semantics of joint type is not used for learning graph (G) (*i.e.*, G w/o JT) and does not take part in the graph convolutional operations for massage passing (P) (*i.e.*, P w/o JT) .

We have three main observations as follows.

1) For the learning of graph of skeleton sequence, by introducing the semantics of joint types, "JL(**G w JT** & P w/o JT) & FL" outperforms "JL(G w/o JT & P w/o JT) & FL" by 0.6% and 0.9% for the CS and CV settings, respectively. Intuitively, if the model does not know the types of the joints, it cannot distinguish the joints with the same coordinates even though their semantics are different. The semantics of joint type is beneficial for learning graph edges.

2) Joint type information is beneficial for message passing in GCN layers. "JL(G w/o JT & **P w JT**) & FL" is superior to "JL(G w/o JT & P w/o JT) & FL" by 1.7% and 1.3% for the CS and CV settings, respectively. The reason is that GCN itself is not aware of the order (type) of joints which makes it hard to learn features of the skeleton data with high structural information. For example, the information contributed from foot joint and wrist joint to a target joint should be different even when the 3D coordinates of the two joints are the same during the message passing. Introducing the joint type information makes GCN more efficient.

3) Using the semantics of joint type for both learning graph and the message passing at the same time ("JL(G w JT & P w JT) & FL") does not bring further benefits in comparison with "JL(G w/o JT & **P w JT**) & FL". For message passing $Y_t = G_t Z_t W$ in Equ. (4), the gradient back-propagated to G_t will also be influenced by Z_t which contains joint type information. Actually, G_t is aware of the joint type information implicitly even though we do not include joint type information in the similarity/affinity learning.

Effectiveness of Exploiting Frame Index. We investigate on two models (rows 5 and 6 in Table 1) to study the influence of the frame index on the frame-level module (**FL**) when the temporal convolution is degraded by setting its kernel size to 1. "JL & FL(w/o T-Conv) w FI" denotes the model using the semantics of frame index. Both models have incorporated the semantics of joint type.

Moreover, we investigate two models (rows 7 and 8 in Table 1) to study the influence of the frame index when the temporal convolution with kernel size of 3 is used. "JL & FL (w T-Conv) w FI" denotes the model using the semantics of frame index. Both models have incorporated the semantics of joint type.

We have two main observations here.

1) When the temporal convolution is disabled (*i.e.*, filter kernel size is 1 instead of 3), "JL & FL(w/o T-Conv) w **FI**" outperforms "JL & FL(w/o T-Conv) w/o FI" by 1.0% and 0.9% for the CS and CV settings, respectively. The frame index information "tells" the network the frame order of skeleton sequence which is beneficial for action recognition.

2) The frame index is helpful for temporal convolution. "JL & FL (w T-Conv) w **FI**" is superior to "JL & FL (w T-Conv) w/o FI" by 0.3% and 0.4% for the CS and CV settings, respectively. The benefits from the semantics of frame index are smaller than those models without temporal convoluitonal (with filter kernel size of 1). The main reason is the temporal convolutional layer enables the network to know the frame order of skeleton sequence to some extent through large kernel size. However, "telling" the networks the semantics of frame index explicitly further improves the performance with negligible cost. We take the scheme "JL & FL (w T-Conv) w **FI**" as our final scheme, which is also referred to as "SGN".

In summary, the explicit modeling of the joint type information benefits the learning of adjacent matrices and the message passing in the GCN layers. The frame index information enables the model to efficiently exploit the information of sequence order.

4.3.2 Effectiveness of Hierarchical Model

We hierarchically model the correlations of the joints in the joint-level module and the frame-level module. To demonstrate its effectiveness, we compare our SGN with two different models and show the results in Table 2.

"SGN w G-GCN" denotes a non-hierarchical scheme where we remove the spatial MaxPooling layer (SMP), and use the combined semantics (*i.e.*, joint type and frame index) and dynamics (position and velocity) in the GCN layers. Instead of constructing a graph for each frame, we build a global adaptive graph with all the joints in all the frames and conduct message passing among all those joints. "SGN w/o SMP" denotes that the spatial MaxPooling layer (SMP) is removed in our scheme "SGN".

We have the following two observations.

1) Modeling the correlations of joints of the same frame by GCN is much more effective than modeling the correlations

Table 2: Effectiveness of our hierarchical model on the NTU60 dataset in terms of accuracy (%).

Method	#Params(M)	CS	CV
SGN w G-GCN	0.68	87.3	93.3
SGN w/o SMP	0.69	88.3	93.9
SGN	0.69	**89.0**	**94.5**

of all joints of all the frames. "SGN w/o SMP" is superior to "SGN w G-GCN" by 1.0% and 0.6% for the CS and CV settings, respectively. Learning a global content adaptive graph is more complicated and difficult.
2) "SGN" outperforms "SGN w/o SMP" by 0.7% and 0.6% for the CS and CV settings, respectively. Aggregating the information of all joints in a frame by MaxPooing (SMP) plays a role of extracting the representative discriminative information (that has large activation values) of a frame. In addition, the spatial MaxPooling layer reduces the subsequent computation burden.

4.3.3 Strong Baseline

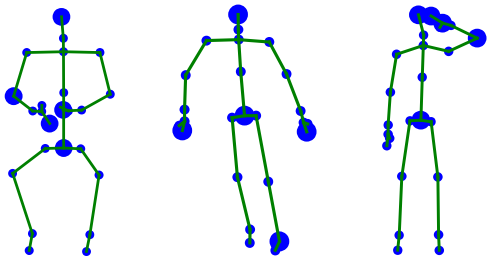
Previous works usually adopt heavy networks for modeling skeleton sequence of low dimensions [40, 39, 37, 58]. We exploit some techniques which have been proven very effective in previous works and build a lightweight strong baseline, which has achieved comparable performance as most other state-of-the-art methods [40, 57, 54, 8]. We hope this serves as a strong baseline for future research in the skeleton-based action recognition field. All models do not use semantics in this section.

We first build a basic baseline ("Baseline") with the overall pipeline similar to that in Fig. 2. There are three differences. 1) The velocity, joint type, and frame index information are not utilized. 2) Data augmentation (DA) (see Data Processing) is not adopted during training. 3) AveragePooling is used instead of Maxpooling as in [54, 37].

Table 3 shows the influence of our adopted techniques for constructing the strong baseline. We have the following three observations. **1)** Data augmentation improves the performance significantly for the CV setting. Through the augmentation on the observed views, some "unseen" views could be "seen" during the training. **2)** Two stream networks (using both position and velocity) [40] have proven effective, but two separate networks double the number of parameters. We fuse the two types of information in the early stage (in input) and it improves the performance significantly with only a negligible number of additional parameters (*i.e.*, 0.01M). **3)** MaxPooling is much more powerful than AveragePooling. The reason is that MaxPooling works like an attention module which drives to learn and select discriminative features.

Table 3: Influence of some techniques on NTU60 dataset in terms of accuracy (%) and number of parameters.

Method	#Params(M)	CS	CV
Baseline	0.61	79.2	81.4
+ DA	0.61	80.6	87.1
+ Velocity	0.62	85.3	91.4
+ MaxPooling	0.62	**86.9**	**92.8**



(a) clapping (b) kicking (c) salute

Figure 3: Visualization of the responses of the spatial MaxPooling layer with respect to three actions, *i.e.*, *clapping*, *kicking*, and *salute*. The top-5 joints selected by SMP are plotted with larger blue circles.

4.3.4 Visualization of SMP

The spatial Maxpooling (SMP) plays a similar role as attention mechanism. We visualize the selected joints by SMP for three actions *i.e.*, *clapping*, *kicking*, and *salute* in Fig. 3. The dimensions of the responses are 256 and each dimension corresponds to one selected joint. We count the times each joint is selected by SMP. The top five chosen joints are shown by large blue circles and the rest are shown by small blue circles. We observe that different actions correspond to different informative joints. The left foot is important for *kicking*. Only the left hand is of great value for *salute*, while both left and right hands are essential for *clapping*. These are consistent with human's perception.

4.3.5 Complexity of SGN

We discuss the complexity of SGN by comparing it with eight state-of-the-art methods for skeleton-based action recognition. As shown in Fig. 1, the number of parameters of VA-RNN [58] is the least, but the accuracy is the poorest. VA-CNN[58] and 2s-AGCN[37] achieve good accuracy, but the numbers of parameters are so large. In comparison with the RNN-based, GCN-based, and CNN-based methods, our proposed SGN achieves slightly better performance with much fewer parameters, which makes SGN attractive for many practical applications which have limited computational power.

Table 4: Performance comparisons on NTU60 with the CS and CV settings in terms of accuracy (%).

Method	Year	CS	CV
HBRNN-L [7]	2015	59.1	64.0
Part-Aware LSTM [36]	2016	62.9	70.3
ST-LSTM + Trust Gate [27]	2016	69.2	77.7
STA-LSTM [41]	2017	73.4	81.2
GCA-LSTM [29]	2017	74.4	82.8
Clips+CNN+MTLN [18]	2017	79.6	84.8
VA-LSTM [57]	2017	79.4	87.6
ElAtt-GRU[59]	2018	80.7	88.4
ST-GCN [54]	2018	81.5	88.3
DPRL+GCNN [44]	2018	83.5	89.8
SR-TSL [40]	2018	84.8	92.4
HCN [23]	2018	86.5	91.1
AGC-LSTM (joint) [39]	2019	87.5	93.5
AS-GCN [24]	2019	86.8	94.2
GR-GCN [8]	2019	87.5	94.3
2s-AGCN [37]	2019	88.5	**95.1**
VA-CNN [58]	2019	88.7	94.3
SGN w/o Sem.	-	86.9	92.8
SGN	-	**89.0**	94.5

Table 5: Performance comparisons on NTU120 with the C-Subject and C-Setup settings in terms of accuracy (%).

Method	Year	C-Subject	C-Setup
Part-Aware LSTM [36]	2016	25.5	26.3
ST-LSTM + Trust Gate [27]	2016	55.7	57.9
GCA-LSTM [29]	2017	58.3	59.2
Clips+CNN+MTLN [18]	2017	58.4	57.9
Two-Stream GCA-LSTM [28]	2017	61.2	63.3
RotClips+MTCNN [19]	2018	62.2	61.8
Body Pose Evolution Map [31]	2018	64.6	66.9
SGN w/o Sem.	-	77.4	79.2
SGN	-	**79.2**	**81.5**

4.4. Comparison with the State-of-the-arts

We compare the proposed SGN with other state-of-the-art methods on the NTU60, NTU 120, and SYSU datasets in Table 4, Table 5, and Table 6, respectively. "SGN w/o Sem." denotes our strong baseline without using semantics.

As shown in Table 4, the introduction of semantics (*Sem.*) brings performance improvement of 2.1% and 1.7% in accuracy for the CS and CV settings, respectively. "ElAtt-GRU" [59] and "Clips+CNN+MTLN" [18] are two representative methods for RNN-based and CNN-based methods, respectively. SGN outperforms them by 8.3% and 9.4% in accuracy for the CS setting, respectively. To better explore the structural information of skeleton, some methods [54, 40] mix CNN and GCN, or LSTM and GCN together. Our proposed SGN is also superior to [54] and [40] by 5.5% and 4.2% in accuracy for the CS setting. The proposed SGN achieves competitive performance when compared to [37] and [58] but with only ten percent of their numbers of parameters as shown in Fig. 1.

Table 6: Performance comparisons on SYSU in terms of accuracy (%). * denotes the model uses parameters pre-trained on NTU60.

Method	Year	CS	SS
VA-LSTM [57]	2017	77.5	76.9
ST-LSTM [26]	2018	76.5	-
GR-GCN [8]	2019	77.9	-
Two stream GCA-LSTM [28]	2017	78.6	-
SR-TSL [40]	2018	81.9	80.7
ElAtt-GRU* [59]	2018	85.7	85.7
SGN	-	83.0	81.6
SGN*	-	**90.6**	**89.3**

As shown in Table 5 and Table 6, the proposed SGN achieves the best accuracy on NTU120 and SYSU. The NTU120 dataset is a newly released dataset and we compare with the results reported in [25]. Semantics (*sem.*) brings gains of 1.8% and 2.3% in accuracy for the C-Subject and the C-Setup settings, respectively.

5. Conclusion

In this work, we have presented a simple yet effective end-to-end semantics-guided neural network for high performance skeleton-based human recognition. We explicitly introduce the high level semantics, *i.e.*, joint type and frame index, as part of the network input. To model the correlations of joints, we have proposed a joint-level module for capturing the correlations of joints in the same frame and a frame-level module for modeling the dependencies of frames where all joints in the same frame are taken as a whole. The semantics helps improve the capability of both the GCN and CNN. In addition, we have developed a strong baseline which is better than most previous methods. With an order of magnitude smaller model size than some previous works, our proposed model achieves the state-of-the-art results on three benchmark datasets.

Acknowledgements

This work was partially supported by the Natural Science Foundation of China (Grant No. 61751308 and 61773311).

References

[1] Jake K Aggarwal and Michael S Ryoo. Human activity analysis: A review. *ACM Computing Surveys*, 2011.

[2] Fabien Baradel, Christian Wolf, Julien Mille, and Graham W Taylor. Glimpse clouds: Human activity recognition from unstructured feature points. In *CVPR*, 2018.

[3] Congqi Cao, Cuiling Lan, Yifan Zhang, Wenjun Zeng, Hanqing Lu, and Yanning Zhang. Skeleton-based action recognition with gated convolutional neural networks. *TCSVT*, 29(11):3247–3257, 2019.

[4] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *CVPR*, 2017.

[5] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv*, 2014.

[6] Yong Du, Yun Fu, and Liang Wang. Skeleton based action recognition with convolutional neural network. In *ACPR*, 2015.

[7] Yong Du, Wei Wang, and Liang Wang. Hierarchical recurrent neural network for skeleton based action recognition. In *CVPR*, 2015.

[8] Xiang Gao, Wei Hu, Jiaxiang Tang, Jiaying Liu, and Zongming Guo. Optimized skeleton-based action recognition via sparsified graph regression. In *ACMMM*, 2019.

[9] Guillermo Garcia-Hernando and Tae-Kyun Kim. Transition forests: Learning discriminative temporal transitions for action recognition and detection. In *CVPR*, 2017.

[10] Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. Convolutional sequence to sequence learning. In *ICML*, 2017.

[11] Fei Han, Brian Reily, William Hoff, and Hao Zhang. Space-time representation of people based on 3d skeletal data: A review. *CVIU*, 2017.

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[13] Tong He, Zhi Zhang, Hang Zhang, Zhongyue Zhang, Junyuan Xie, and Mu Li. Bag of tricks for image classification with convolutional neural networks. In *CVPR*, 2019.

[14] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 1997.

[15] Jian-Fang Hu, Wei-Shi Zheng, Jianhuang Lai, and Jianguo Zhang. Jointly learning heterogeneous features for rgb-d activity recognition. In *CVPR*, 2015.

[16] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv*, 2015.

[17] Gunnar Johansson. Visual perception of biological motion and a model for its analysis. *Perception & psychophysics*, 1973.

[18] Qiuhong Ke, Mohammed Bennamoun, Senjian An, Ferdous Sohel, and Farid Boussaid. A new representation of skeleton sequences for 3d action recognition. In *CVPR*, 2017.

[19] Qiuhong Ke, Mohammed Bennamoun, Senjian An, Ferdous Sohel, and Farid Boussaid. Learning clip representations for skeleton-based 3d action recognition. *TIP*, 2018.

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

[21] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv*, 2016.

[22] Chao Li, Qiaoyong Zhong, Di Xie, and Shiliang Pu. Skeleton-based action recognition with convolutional neural networks. In *ICMEW*, 2017.

[23] Chao Li, Qiaoyong Zhong, Di Xie, and Shiliang Pu. Co-occurrence feature learning from skeleton data for action recognition and detection with hierarchical aggregation. In *IJCAI*, 2018.

[24] Maosen Li, Siheng Chen, Xu Chen, Ya Zhang, Yanfeng Wang, and Qi Tian. Actional-structural graph convolutional networks for skeleton-based action recognition. In *CVPR*, 2019.

[25] 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. *TPAMI*, 2019.

[26] Jun Liu, Amir Shahroudy, Dong Xu, Alex C Kot, and Gang Wang. Skeleton-based action recognition using spatio-temporal lstm network with trust gates. *TPAMI*, 2018.

[27] Jun Liu, Amir Shahroudy, Dong Xu, and Gang Wang. Spatio-temporal lstm with trust gates for 3d human action recognition. In *ECCV*, 2016.

[28] Jun Liu, Gang Wang, Ling-Yu Duan, Kamila Abdiyeva, and Alex C Kot. Skeleton-based human action recognition with global context-aware attention lstm networks. *TIP*, 2017.

[29] Jun Liu, Gang Wang, Ping Hu, Ling-Yu Duan, and Alex C Kot. Global context-aware attention lstm networks for 3d action recognition. In *CVPR*, 2017.

[30] Mengyuan Liu, Hong Liu, and Chen Chen. Enhanced skeleton visualization for view invariant human action recognition. *PR*, 2017.

[31] Mengyuan Liu and Junsong Yuan. Recognizing human actions as the evolution of pose estimation maps. In *CVPR*, 2018.

[32] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015.

[33] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *ICML*, 2010.

[34] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. *arXiv*, 2016.

[35] Ronald Poppe. A survey on vision-based human action recognition. *Image and vision computing*, 2010.

[36] Amir Shahroudy, Jun Liu, Tian-Tsong Ng, and Gang Wang. Ntu rgb+ d: A large scale dataset for 3d human activity analysis. In *CVPR*, 2016.

[37] Lei Shi, Yifan Zhang, Jian Cheng, and Hanqing Lu. Two-stream adaptive graph convolutional networks for skeleton-based action recognition. In *CVPR*, 2019.

[38] Jamie Shotton, Andrew Fitzgibbon, Mat Cook, Toby Sharp, Mark Finocchio, Richard Moore, Alex Kipman, and Andrew Blake. Real-time human pose recognition in parts from single depth images. 2011.

[39] Chenyang Si, Wentao Chen, Wei Wang, Liang Wang, and Tieniu Tan. An attention enhanced graph convolutional lstm network for skeleton-based action recognition. In *CVPR*, 2019.

[40] Chenyang Si, Ya Jing, Wei Wang, Liang Wang, and Tieniu Tan. Skeleton-based action recognition with spatial reasoning and temporal stack learning. In *ECCV*, 2018.

[41] Sijie Song, Cuiling Lan, Junliang Xing, Wenjun Zeng, and Jiaying Liu. An end-to-end spatio-temporal attention model for human action recognition from skeleton data. In *AAAI*, 2017.

[42] Sijie Song, Cuiling Lan, Junliang Xing, Wenjun Zeng, and Jiaying Liu. Skeleton-indexed deep multi-modal feature learning for high performance human action recognition. In *ICME*, 2018.

[43] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *CVPR*, 2019.

[44] Yansong Tang, Yi Tian, Jiwen Lu, Peiyang Li, and Jie Zhou. Deep progressive reinforcement learning for skeleton-based action recognition. In *CVPR*, 2018.

[45] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NIPS*, 2017.

[46] Jiang Wang, Zicheng Liu, Ying Wu, and Junsong Yuan. Mining actionlet ensemble for action recognition with depth cameras. In *CVPR*, 2012.

[47] Pichao Wang, Wanqing Li, Philip Ogunbona, Jun Wan, and Sergio Escalera. Rgb-d-based human motion recognition with deep learning: A survey. *CVIU*, 2018.

[48] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *CVPR*, 2018.

[49] Xiaolong Wang and Abhinav Gupta. Videos as space-time region graphs. In *ECCV*, 2018.

[50] Daniel Weinland, Remi Ronfard, and Edmond Boyer. A survey of vision-based methods for action representation, segmentation and recognition. *CVIU*, 2011.

[51] Junwu Weng, Mengyuan Liu, Xudong Jiang, and Junsong Yuan. Deformable pose traversal convolution for 3d action and gesture recognition. In *ECCV*, 2018.

[52] Lu Xia, Chia-Chih Chen, and Jake K Aggarwal. View invariant human action recognition using histograms of 3d joints. In *CVPRW*, 2012.

[53] Wayne Xiong, Lingfeng Wu, Fil Alleva, Jasha Droppo, Xuedong Huang, and Andreas Stolcke. The microsoft 2017 conversational speech recognition system. In *ICASSP*, 2018.

[54] Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *AAAI*, 2018.

[55] Gang Yu, Zicheng Liu, and Junsong Yuan. Discriminative orderlet mining for real-time recognition of human-object interaction. In *ACCV*, 2014.

[56] Kiwon Yun, Jean Honorio, Debaleena Chattopadhyay, Tamara L Berg, and Dimitris Samaras. Two-person interaction detection using body-pose features and multiple instance learning. In *CVPRW*, 2012.

[57] Pengfei Zhang, Cuiling Lan, Junliang Xing, Wenjun Zeng, Jianru Xue, and Nanning Zheng. View adaptive recurrent neural networks for high performance human action recognition from skeleton data. In *ICCV*, 2017.

[58] Pengfei Zhang, Cuiling Lan, Junliang Xing, Wenjun Zeng, Jianru Xue, and Nanning Zheng. View adaptive neural networks for high performance skeleton-based human action recognition. *TPAMI*, 2019.

[59] Pengfei Zhang, Jianru Xue, Cuiling Lan, Wenjun Zeng, Zhanning Gao, and Nanning Zheng. Adding attentiveness to the neurons in recurrent neural networks. In *ECCV*, 2018.

[60] Pengfei Zhang, Jianru Xue, Cuiling Lan, Wenjun Zeng, Zhanning Gao, and Nanning Zheng. Eleatt-rnn: Adding attentiveness to neurons in recurrent neural networks. *TIP*, 2019.

[61] Zhengyou Zhang. Microsoft kinect sensor and its effect. *IEEE multimedia*, 2012.

[62] Heliang Zheng, Jianlong Fu, Zheng-Jun Zha, and Jiebo Luo. Learning deep bilinear transformation for fine-grained image representation. In *NIPS*, 2019.

[63] Wentao Zhu, Cuiling Lan, Junliang Xing, Wenjun Zeng, Yanghao Li, Li Shen, and Xiaohui Xie. Co-occurrence feature learning for skeleton based action recognition using regularized deep lstm networks. In *AAAI*, 2016.