Beyond Joints: Learning Representations From Primitive Geometries for Skeleton-Based Action Recognition and Detection

Hongsong Wang and Liang Wang, *Senior Member, IEEE*

***Abstract*— Recently, skeleton-based action recognition becomes popular owing to the development of cost-effective depth sensors and fast pose estimation algorithms. Traditional methods based on pose descriptors often fail on large-scale datasets due to the limited representation of engineered features. Recent recurrent neural networks (RNN) based approaches mostly focus on the temporal evolution of body joints and neglect the geometric relations. In this paper, we aim to leverage the geometric relations among joints for action recognition. We introduce three primitive geometries: joints, edges, and surfaces. Accordingly, a generic end-to-end RNN based network is designed to accommodate the three inputs. For action recognition, a novel viewpoint transformation layer and temporal dropout layers are utilized in the RNN based network to learn robust representations. And for action detection, we first perform frame-wise action classification, then exploit a novel multi-scale sliding window algorithm. Experiments on the large-scale 3D action recognition benchmark datasets show that joints, edges, and surfaces are effective and complementary for different actions. Our approaches dramatically outperform the existing state-of-the-art methods for both tasks of action recognition and action detection.**

***Index Terms*— Skeleton-based action recognition, geometric relations, viewpoint transformation, action detection.**

I. Introduction

ACTION recognition and detection are important topics in computer vision with many related techniques such as optical flow estimation [1], human pose recovering [2], unsupervised deep feature learning [3] and feature encoding [4]. In the early days, most work focuses on analyzing human actions from RGB video. Despite a lot of effort has been made [5]–[13], it is still an unresolved challenge due to some factors like illumination changes, occlusion and background clutter. Recently, owing to the development of 3D sensors, 3D human activity analysis [14] has become popular. There are two main ways to obtain the 3D data. One is by using marker-based motion capture systems such as MoCap. The other is to use cost-effective range sensors such as Microsoft Kinect.

Human skeleton is a graph of joints connected by bones. Psychological experiments of Johansson show that participants are able to recognize actions of pedestrians by simply observing the movements generated by light bulbs attached to several joints over their bodies [15]. Thanks to the range sensors, skeletons can be reliably estimated from depth images with real-time pose estimation algorithms [16]. These facilitate the research of skeleton-based action recognition [17], [18] and various algorithms have been proposed accordingly.

Most traditional approaches focus on designing handcrafted pose descriptors [14], [19], [20]. These descriptors are divided into three categories: joint based representations, mined joint based descriptors and dynamics based descriptors. Joint based representations are intended to capture the correlation of the body joints. Mined joint based descriptors try to learn what body parts are involved, which are effective to discriminate among actions. Dynamics based descriptors treat the skeletons as 3D trajectories and model the temporal dynamics. These descriptors model the geometric relations among body parts or joints, but often fail on large-scale datasets due to the limited representation ability of engineered features. To the best of our knowledge, there is no deep learning methods which learn effective geometric representations from skeletons.

Recently, there is a growing trend toward recurrent neural networks (RNN) based methods. Different structures such as hierarchical RNN [18], RNN with regularizations [21], differential RNN [22], part-aware long short-term memory (LSTM) [23], spatio-temporal LSTM [24], spatial-temporal attention based RNN [25] and two-stream RNN [26] have been proposed to learn representations and recognize actions from raw skeletons. However, most of these methods just concatenate the coordinates of joints at each time step before applying RNN based methods. Thus, geometric relations among different joints are lost at this pre-processing step. In fact, an action is the evolution of articulated bones united by joints. The relative geometries among joints, i.e., bones or body parts,

Manuscript received November 26, 2017; revised April 17, 2018; accepted May 10, 2018. Date of publication May 17, 2018; date of current version June 1, 2018. This work was supported in part by the National Key Research and Development Program of China under Grant 2016YFB1001000, in part by the National Natural Science Foundation of China under Grant 61525306, Grant 61633021, Grant 61721004, and Grant 61420106015, in part by the Beijing Natural Science Foundation under Grant 4162058, and in part by the Capital Science and Technology Leading Talent Training Project under Grant Z181100006318030. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Dacheng Tao. *(Corresponding author: Liang Wang.)*

The authors are with the National Laboratory of Pattern Recognition, Center for Research on Intelligent Perception and Computing, Institute of Automation, Chinese Academy of Sciences, University of Chinese Academy of Sciences, Beijing 100190, China (e-mail: hongsong.wang@nlpr.ia.ac.cn; wangliang@nlpr.ia.ac.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2018.2837386

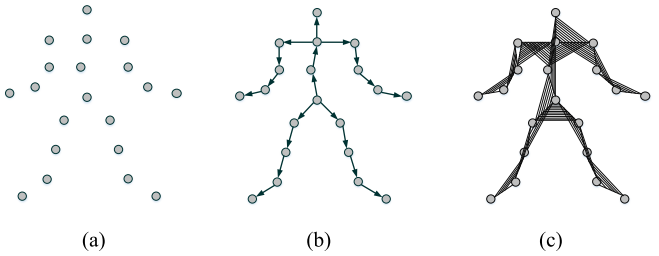


Fig. 1. Based on the physical structure of human body, three kinds of primitive data (i.e., joints, edges, surfaces) are used as the inputs of deep networks.

provide a meaningful description of actions. The trajectory of a single joint only carries motion information, and does not carry any shape or geometric information.

Regarding skeletons as a sequence of a graph of joints, we aim to leverage geometries of the graph structure for skeleton-based action recognition. Based on the physical structure of the human body, we consider three types of geometries: joints, edges and surfaces, which are illustrated in Figure 1. The joints are isolated points of human body, and are the focus of attention by most of the previous methods. The edges are bones which connect two adjacent joints. They can be denoted by the relative positions of the joints. The temporal dynamics of an edge is the relative movement of the two corresponding joints. The surfaces are the planes formed by two neighboring articulating bones. The plane represents the body part, and reflects the relative geometry among multiple adjacent joints. It also shows complex dynamics during an action. To facilitate effective training and make fair comparisons with the joints and edges, we use the normal vector to represent the surface.

To learn discriminative representations of actions from the primitive geometric data, we propose a novel deep architecture which adapts the inputs of joints, edges and surfaces. The backbone of this architecture is built on multiple layers of bidirectional LSTM (BiLSTM). To accommodate data observed from different viewpoints and learn robust representations, a viewpoint transformation layer is therefore proposed to transform the skeletons during training. For different inputs, we derive equations of the transformed input tensors in 3D space. In line with our expectations, the viewpoint transformation matrices of joints, edges and surfaces are the same. We apply our approach which learns representations from joints, edges and surfaces for both action recognition and action detection. For action recognition, we further introduce the temporal dropout layer to process the sequence and put it before the BiLSTM layer. After representation learning, a max pooling layer along the time axis is employed before the final classification. For action detection, we first design an RNN based architecture to perform frame-wise action classification. Then, a novel multi-scale sliding window approach is proposed to produce the detection results with arbitrary lengths.

In summary, the main contributions are listed as follows:

- We first learn geometric representations for skeleton-based action recognition, and introduce three primitive geometries, i.e., joints, edges and surfaces. We derive the viewpoint transformation matrices, and find that these matrices are the same.
- For action recognition, we propose a novel RNN based architecture by incorporating the viewpoint transformation layer and the temporal dropout layers.
- For action detection, we propose a generic framework by combining the proposed RNN based architecture for frame-wise action classification and a novel multi-scale sliding window search algorithm.
- Our methods dramatically outperform the previous state-of-the-art methods on large-scale 3D action recognition datasets for both action recognition and action detection.

The remainder of the paper is organized as follows. Section II reviews related work. Section III details the formulations of joints, edges and surfaces and the approaches of both action recognition and action detection. Experimental results and ablation studies are presented in Section IV. The conclusions are finally drawn in Section V.

II. Related Work

In this section, we briefly review approaches related to ours, i.e., handcrafted descriptors of skeletons, skeleton-based action recognition by recurrent neural networks (RNN) and skeleton-based action detection.

A. Handcrafted Descriptors of Skeletons

Traditional skeleton-based action recognition methods mainly focus on handcrafted features, which are divided into three categories: joint-based descriptors, mined joint based descriptors and dynamics-based descriptors. The reader can refer to a survey [20] for details.

Joint based descriptors aim to capture the relative body joint locations. For example, Ellis *et al.* [27] compute clustered pairwise joint distances between the current and the previous frames. Müller *et al.* [28] introduce a class of boolean features expressing geometric relations between body points of a pose. Mined joint based descriptors try to learn the discriminative body parts. For example, Ofli *et al.* [29] present an ordered set of the most informative joints in each temporal window by exploiting the relative informativeness of all the joint angles. Seidenari *et al.* [30] use joint positions to align multiple body sub-parts and learn the most informative body parts with a modified nearest-neighbor classifier. Dynamics-based descriptors model the temporal dynamics of either subsets or all the joints. For example, Chaudhry *et al.* [31] model a set of time series of shape context features by linear dynamical systems (LDS) and use the parameters to represent the action sequence. Slama *et al.* [32] represent an action by a dynamical system whose observability matrix is characterized as an element of a Grassmann manifold.

These approaches are not generic and the descriptors may not be optimal for large-scale recognition. In this paper, we design an end-to-end network to learn geometric representations and recognize actions.

B. Action Recognition by RNN

Recently, there are many deep learning methods which directly learn representations from raw skeletons

(e.g., [33]–[38]). Prior to our work, several RNN based models have been put forward. For example, Du *et al.* [18], [39] design an end-to-end hierarchical RNN architecture in which representations learned from skeletons of body parts are hierarchically fused. Loosening the restrictions that joints are connected in the same body part, Zhu *et al.* [21] propose a fully connected deep LSTM network with regularization terms to learn co-occurrence features of joints. Shahroudy *et al.* [23] propose a part-aware extension of LSTM to model the long-term temporal correlation of the features for each body part. Zhang *et al.* [40] evaluate a set of handcrafted features by using 3-layer LSTM and find that the distance between joints and selected lines outperforms other features. Liu *et al.* [24] extend LSTM to the spatial-temporal domain to explicitly model the dependencies between joints and introduce a new gating mechanism to handle noise and occlusion in skeleton data. Song *et al.* [25] design an end-to-end spatial and temporal attention based RNN structure to learn discriminative spatial and temporal features. Liu *et al.* [41], [42] present global context-aware attention LSTM unit to selectively focus on the informative joints for the input sequence. Wang and Wang [26] present a two-stream RNN architecture to leverage both temporal dynamics and spatial configurations of joints. Zhang *et al.* [43] design a view adaptive RNN which could automatically adapt to the most suitable observation viewpoints. Recently, Wang and Wang [44] explore several multi-task RNN structures by leveraging supervisions of both action recognition and person identification.

However, most of these approaches mentioned above merely focus on learning representations from isolated joints. In contrast, we investigate the end-to-end learning with different inputs of geometries.

C. Action Detection From Skeletons

Action detection aims to predict the classes as well as the starting and ending frames of actions within an untrimmed long sequence. While most of previous studies are about action recognition, fewer works concentrate on action detection by using human skeletons. These approaches mainly use sliding windows to generate segments and subsequently cast the problem as action recognition from segments. For example, Nowozin and Shotton [45] introduce the notion of action points and classify every overlapping 35-frames intervals. Based on the same notion of action points, Sharaf *et al.* [46] perform real-time and multi-scale action detection using a descriptor derived from angles and angular velocities. Zhao *et al.* [47] construct a dynamic matching based feature vector for each frame to detect the starting and ending frames of actions. Zanfir *et al.* [48] propose a moving pose descriptor and a modified non-parametric kNN classifier based on discriminative key frames with augmented temporal information, and show accurate action detection in unsegmented sequences.

RNN has already been investigated for skeleton-based action detection. Li *et al.* [49] incorporate frame-wise action classification and regression of the starting and ending points into the joint classification-regression RNN network. To facilitate large-scale 3D action detection, Liu *et al.* [50] collect a new benchmark covering a wide range of complex human activities for continuous multi-modal human action detection, and evaluate several action detection methods.

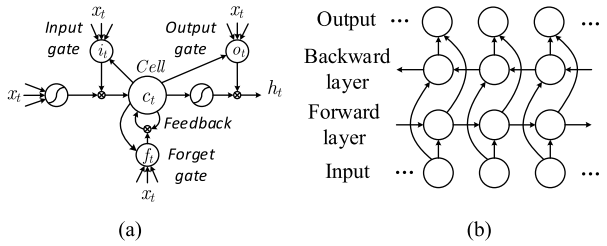


Fig. 2. (a) An LSTM block with input, output, and forget gates [51]. (b) An unfolded bidirectional network [52]. The solid line denotes the weighted connection between units and the weights are reused at every timestep. The outputs of the forward and backward layers are concatenated to present the output sequence.

Different from the above approaches, we learn representations from the input of geometries of skeletons, and perform end-to-end frame-wise action classification. We also propose a novel multi-scale sliding window algorithm which efficiently and effectively detects actions with arbitrary lengths.

III. Method

In this section, we first review some necessary backgrounds. Then, we introduce three types of geometries embedded in the skeleton data. Finally, a novel deep architecture is proposed and the approaches of both action recognition and action detection are discussed.

A. Preliminaries

Recurrent neural networks (RNN) have an internal state to exhibit dynamic temporal behavior, which make them naturally suitable for supervised sequence labeling. They map an input sequence to another output sequence, and can process sequences with arbitrary length. Due to the vanishing and exploding gradient problems, the standard RNN cannot store information for long periods of time. Long short-term memory (LSTM) [51] is accordingly proposed to address this problem. The structure of an LSTM unit is shown in Figure 2(a). The hidden state representation h_t of an unit at each time step t is updated as:

$$\begin{aligned} i_t &= \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i) \\ f_t &= \sigma(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f) \\ c_t &= f_t c_{t-1} + i_t \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \\ o_t &= \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o) \\ h_t &= o_t \tanh(c_t) \end{aligned} \tag{1}$$

where x_t denotes the input, and i_t, f_t, o_t denote the internal representations correspond to the input gate, forget gate and output gate, respectively. All the matrices W are the connection weights and all the variables b are biases. The gates are used to determine when the input is significant enough to remember, when it should continue to remember or forget the value, and when it should output the value.

Both past and future contexts are important for sequence labelling. For the task of action recognition, the current

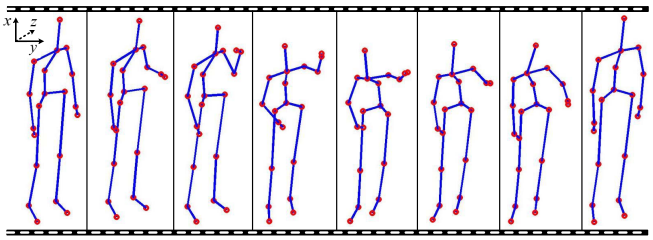


Fig. 3. The human skeleton plotted in a 3D coordinate system. In this example, the action is *forward punch*. The action is intuitively described using the relative geometry between neighboring joints in the moving hand.

prediction depends not only on the past but also on the expectations of the future. Bidirectional recurrent neural networks (BRNN) [52] elegantly combine both forward and backward dependencies by using two separate recurrent hidden layers to present the input sequence. An illustration of unfolded BRNN is shown in Figure 2(b). By using BRNN, the output sequence at each time step provides complete historical and future contexts.

RNN architectures are very suitable for sequence classification, e.g., skeleton-based action recognition, where an input sequence is assigned with a single class label. Deep RNN can be constructed by stacking multiple RNN layers on top of each other. The formulation of stacked RNN is:

$$h_t^{(l)} = f_h^{(l)}(h_t^{(l-1)}, h_{t-1}^{(l)}) \tag{2}$$

where $h_t^{(l)}$ is the hidden state of the l-th level at time step t, and $f_h^{(l)}$ is nonlinear function of the RNN unit. When $l = 1$, the state is computed using x_t instead of $h_t^{(l-1)}$.

B. Skeleton Data Beyond Joints

The skeleton data is a sequence of 3D coordinates of points which form the deformed structure of human body. During an action, various movements of the points occur when the body moves intentionally. These points can be connected according to the physical structure of body joints. The body structure can be regarded as a graph, with joints as points and bones as edges. A visualization of the evolution of connected bones in 3D space during an action is shown in Figure 3. Given a human subject, the skeleton data involves two geometric constraints. First, as a bone length is constant, the distance between two adjacent points along a connected segment is fixed. Second, three points which constitute two intersecting segments lie on the same plane.

Based on above observations, the skeleton data conveys three types of information: the isolated joints of human body, the edges which denote the connected segments, and the surfaces spanned by intersecting segments. To make the most of the ability of deep networks to learn representations from raw data, feature engineering techniques should be prevented and the simple primitive representations are encouraged. The details are presented as follows.

1) Joints: The isolated joints are shown in Figure 1(a). Assume there are M joints for the structure of a human body, the coordinates of points at a time step form a $M \times 3$ matrix. If the length of a sequence is T, the skeleton data can be denoted as a tensor X with dimensions $T \times M \times 3$. The coordinates of joints that change over time reflect temporal dynamics of actions. In fact, the experiments of Johansson [15] show that several main joints in adequate combinations of proximal movements give the visual system sufficient information about human action. Most previous methods simply reshape X by collapsing its second dimension to get a matrix with dimensions $T \times 3M$. With T varied for different sequences, some RNN variants are used to learn representations and recognize human actions.

The coordinates of joints of a given viewpoint can be transformed into another viewpoint by using a rotation matrix. Let p_k be the coordinate vector of a joint at a particular time step, the new coordinate vector can be obtained by:

$$\widetilde{p_k} = R p_k \tag{3}$$

where R is the rotation matrix with a dimension of 3×3.

Given an input sequence, we assume that R is the same for different joints and different time steps. Thus, for the joints tensor X, the new tensor observed from another viewpoint can be mathematically expressed as:

$$\widetilde{X} = X \times_3 R^T \tag{4}$$

where $\times_3$ denotes 3-mode tensor multiplication, $\widetilde{X}$ and X have the same dimension.

2) Edges: Beside the temporal dynamics of joints, the bone motion patterns distinguish actions. A graph is used to represent the physical connections of body joints. The joints are denoted by the nodes and the bones are denoted by the edges. Given a graph of M nodes, there exist $M - 1$ edges. The edge mainly denotes the direction of the bone. For the sake of convenience, we specify the directions of edges as in Figure 1(b). Each node has a coordinate vector, and each edge is represented by subtracting the vector of the start point from that of the end point. Mathematically,

$$e_k = p_i - p_j \tag{5}$$

where e_k, p_i, p_j are the coordinate vectors of the edge, end point and start point, respectively.

The edges of a skeleton sequence can be represented by a tensor Y with dimensions $T \times (M - 1) \times 3$. We can also use the vectors of edges to represent the nodes in Figure 1(b). Specifically, a node can be denoted by the vector of an edge which ends at that node. For a node that do not have end points of edges (e.g., the joint of hip-spine in Figure 1(b)), we denote it by a zero vector. In this way, we increment the second dimension of Y by one and make the dimensions of X and Y the same.

The coordinate vectors of edges of a given viewpoint can also be transformed into another viewpoint by a rotation matrix. For the coordinate vector of an edge at a particular time step, the transformation can be easily deduced based on Equations (5) and (3):

$$\widetilde{e_k} = \widetilde{p_i} - \widetilde{p_j} = R e_k \tag{6}$$

where $\widetilde{e_k}$ denotes the transformed vector of e_k. Similarly, it can be expressed in the form of 3-mode tensor multiplication:

$$\widetilde{Y} = Y \times_3 R^T \tag{7}$$

where $\widetilde{Y}$ denotes the transformed tensor of edges. Comparing Equations (4) and (7), we find that the rotation matrices of joints and edges are the same.

3) Surfaces: The edges model the adjacency relations between the joints. It cannot describe the situation when two joints are adjacent to the same joint, which involves two adjacent edges. Indeed, the relative movements of adjacent bones also contribute to action recognition. As two adjacent edges form a plane surface, we use the normal vector to denote the plane. Let e_i, e_j be the vectors representing the two adjacent edges, the normal vector s_k is:

$$s_k = e_i \times e_j \tag{8}$$

where $\times$ denotes the cross product in 3D space. Here, we do not normalize the vector as the magnitude reflects the intersecting angel of the corresponding edges. In order to keep the size of the normal vector the same as the coordinate vector, we simply multiply it by a constant of 100.

For human body with M joints, there are $(M+2)$ surfaces in total. To make a fair comparison with joints and edges, we exclude two surfaces with duplicate information (the normal vector can be represented by other normal vectors). This results in M surfaces defined in Figure 1(c). Thus, the normal vectors of a sequence can also be represented by a tensor Z with dimensions $T \times M \times 3$.

The normal vector of a given viewpoint can also be observed from another viewpoint. Based on Equations (8) and (6), the new normal vector of a surface at a particular time step is expressed as:

$$\widetilde{s_k} = (Re_i) \times (Re_j) = \mathrm{Co}(R)(e_i \times e_j) = \mathrm{Co}(R)s_k \tag{9}$$

where $\mathrm{Co}(R)$ is the cofactor matrix of R. The cofactor matrix is the transpose of the adjugate matrix. For an invertible matrix R, we have:

$$\mathrm{Co}(R) = (\det(R))(R^{-1})^T \tag{10}$$

where $(R^{-1})^T$ is the transpose of the inverse of R.

It should be noted that the determinant of a rotation matrix is 1 and the inverse of a rotation matrix is its transpose. We can further deduce the formula as:

$$\mathrm{Co}(R) = (R^{-1})^T = R \tag{11}$$

So, for the tensor representation of surfaces, it also follows that:

$$\widetilde{Z} = Z \times_3 R^T \tag{12}$$

where $\widetilde{Z}$ is the transformed tensor of surface normal vectors. Comparing Equations (4), (7) and (12), we conclude that joints, edges and surfaces share the same rotation matrix.

C. Method for Action Recognition

For action recognition, we aim to predict an action label for a given input sequence. Feeding three kinds of skeleton data as the inputs, the whole network architecture for action recognition is shown in Figure 4(a). As long short-term memory (LSTM) is preferred over standard recurrent neural networks (RNN) for skeleton-based action recognition [23], [26],

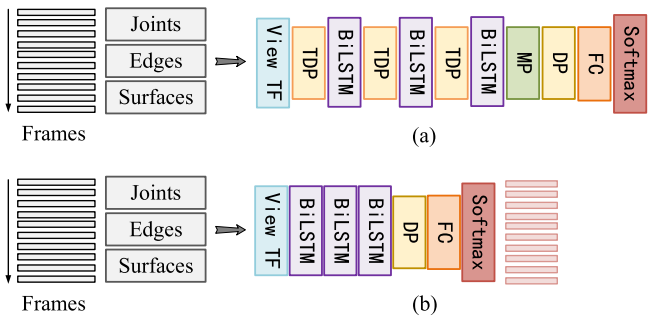


Fig. 4. The network structure for action recognition and detection by learning representations from joints, edges and surfaces. Here, View TF, TDP, BiLSTM, MP, DP and FC denote the viewpoint transformation, temporal dropout, bidirectional LSTM, temporal max pooling, dropout and fully-connected layers, respectively. (a) The end-to-end trainable architecture for action recognition. (b) The network structure for action detection. Here, for frame-wise classification, the DP, FC and Softmax are performed independently for the learned representation of each frame, and the parameters of FC are shared across different frames.

we adopt bidirectional LSTM (BiLSTM) for all the RNN layers. The backbone of our network consists of three BiLSTM layers due to its excellent performance for classification. A temporal max pooling (MP) layer along the time axis is placed on top of the last BiLSTM layer to obtain a time invariant vector representation of the sequence. After that, dropout (DP) is employed and a fully-connected (FC) layer with softmax activation is used to classify actions. In particular, to facilitate feature learning and improve model robustness, we introduce two novel layers: viewpoint transformation (View TF) layer and temporal dropout (TDP) layer. The details are described as follows.

1) Viewpoint Transformation: Skeletons may be observed from an arbitrary camera viewpoint in a realistic scenario. In order to reach view-invariant representations, we aim to use a viewpoint transformation layer to transform the skeleton data in 3D space. This layer is placed at the beginning of the network architecture to accommodate the inputs of joints, edges and surfaces.

For an input sequence, the tensors X, Y, Z are transformed with the same transformation matrix R. Based on Euler's rotation theorem, R can be expressed as a composition of rotations about x, y, z axes:

$$R = R_z(\gamma)R_y(\beta)R_x(\alpha) \tag{13}$$

where γ, β, α are rotate angles of z, y, x, respectively. Details about the formulation of the three basic rotation matrices in terms of the rotate angles are presented in [26] and [39]. For one skeleton sequence, the matrix R is determined by three independent angle parameters. These parameters can be estimated or learned from the skeletons by making some relevant assumptions. For example, Liu *et al.* [53] assumed that the z axis after transformation is aligned with the longer dimension of the torso and used principal component analysis (PCA) to estimate the three angle parameters. In contrast, we employ a simple but effective approach. During training, we randomly select the angles within a certain range and calculate the matrix R to transform the inputs. Here, β, α are sampled from

$(-\pi/2, \pi/2)$ and γ is fixed with 0, as the rotation plane of the camera is almost perpendicular to the z axis. During testing, γ, β, α are all fixed with 0, and the original tensors of joints, edges and surfaces are used.

2) Temporal Dropout: Skeletons collected by sensors like Kinect may not always be reliable due to noise and occlusion. To address this problem, some work [18], [21] smoothed the skeleton joint positions in the temporal domain by using the Svaitzky-Golay filter. Other work [24] designed a trust gate to Spatio-Temporal LSTM unit and analyzed the reliability of the coordinates at each spatio-temporal step.

We adopt an alternative approach based on dropout [54], which improves model robustness. For the standard dropout, each hidden unit is randomly omitted from the network with a probability of p_{drop} during training. For testing, all activations are used and $1 - p_{drop}$ is multiplied to account for the increase in the expected bias. Temporal dropout is slightly different from the standard dropout. Given the $T \times d$ matrix representation of a sequence, where T is the length of the sequence and d is the feature dimension, we only perform T dropout trials and extend the dropout value across the feature dimension. This technique is inspired by the spatial dropout [55] to process the convolution feature 4D tensor. We modify it for 3D tensor and apply it for feature learning from sequences. As shown in Figure 4(a), the temporal dropout is performed before the bidirectional LSTM layer.

D. Method for Action Detection

For action detection, our goal is to predict all the actions and their corresponding starting and ending frames contained in a long untrimmed sequence. We cast this problem as frame-wise action classification. The action classes should be extended with a blank class which is utilized to annotate the frames without actions. After frame-wise action prediction, we leverage a sliding window based approach to fine-tune the predicted frame-wise extended class probabilities and obtain the desired outputs. The details are presented below.

1) Network Architecture: The architecture for frame-wise action classification is shown in Figure 4(b). Compared with the structure for action recognition of the whole sequence (see Figure 4(a)), the architecture for detection has no temporal max pooling layer and no temporal dropout layer. The global representation of the sequence obtained by the temporal max pooling layer has no benefit for frame-wise action classification. The temporal dropout layer randomly drops representations of the input frames, which is likely harmful to predict the actions of the omitted frames.

Similar to the method for action recognition, we feed joints, edges and surfaces as the inputs and utilize the viewpoint transformation layer to transform the skeletons ahead of the bidirectional LSTM layers. For each sequence, the fully-connected layer with softmax activation is operated on a matrix representation for which the row denotes the number of frames and the column denotes the feature dimensionality. The output is the class probabilities for all the frames of the sequence.

2) Multi-Scale Sliding Window: The network for frame-wise action classification could yield noisy predictions for some frames. For example, the network may wrongly predict the action class for a small number of frames in the intervals of an action, and occasionally predict an action during the gap without actions. We assume that two intervals of actions in the long sequence have no overlapping frames, i.e., one frame can only be annotated with an action at most. This assumption fits the examples of most datasets of action detection.

To reduce noise and preserve the continuity in predictions among neighboring frames, we design a multi-scale sliding based approach to search for actions. This approach simultaneously employs two windows: a large window to predict the actions, and a small window to predict the starting and ending frames. First, we use the large window to slide over the sequence and average the probabilities of the frames inside the window of the action predicted by the anchor frame. If the averaged probability is higher than a certain threshold, we assume that an action is detected and start to search for the ending frame by increasing the anchor frame by the size of this window. The step is iterated until the averaged probability is lower than this threshold. The starting frame is considered as the anchor frame before this iteration. Then, in order to determine the ending frame, we search backward with a small window in a similar way by decreasing the anchor frame until the averaged probability is higher than this threshold. When the action, the starting frame and the ending frame are determined, the new search starts from the position of the anchor frame. The details are summarized in Algorithm 1. This multi-scale sliding window approach could detect actions with arbitrary length. It is a fast search algorithm, and the time complexity is $O(n)$, where n is the length of the sequence.

IV. Experiments

We empirically evaluate the proposed models on standard benchmarks of both action recognition and action detection, and compare them with the state-of-the-art approaches. We also provide extensive analyses of joints, edges and surfaces.

A. Datasets

1) NTU RGB+D: The NTU RGB+D dataset [23] is currently the largest depth-based action recognition dataset. It is captured by Kincet v2 in various background conditions with 3D coordinates of 25 joints. There are 60 different action classes including daily, mutual, and health-related actions. The actions are performed by 40 different subjects, whose age range is from 10 to 35. The dataset contains more than 56 thousand sequences and 4 million frames. Numerous variations in subjects and views, and large amount of samples make it highly suitable for deep learning based methods. We follow the cross-subject and cross-view evaluations [23], and report the classification accuracy in percentage for each evaluation. For cross-subject evaluation, both the training set and the test set consist of 20 subjects. The IDs of training subjects are: 1, 2, 4, 5, 8, 9, 13, 14, 15, 16, 17, 18, 19, 25, 27, 28, 31, 34, 35, 38. The remaining subjects are reserved for testing. For cross-view evaluation, samples of cameras 2 and 3 are for training, and samples of camera 1 are for testing.

Algorithm 1 Multi-Scale Sliding Window for Sequences

Input: P – frame-wise probabilities of the extended classes.
Input: w_1 – a large window, w_2 – a small window, μ – a probability threshold parameter.

$A, S, E \leftarrow \{\}, \{\}, \{\}$.
$C \leftarrow$ a zero matrix of size of P.
$n \leftarrow$ the length of the sequence.
$m \leftarrow$ the number of the extended classes.
$D \leftarrow$ a zero vector of size n.
for $t = 1, \cdots, n$ **do**
  if $t > 1$ **then**
    $C_t \leftarrow C_{t-1}$.
  end if
  $k \leftarrow \underset{j}{\text{argmax}}\{P_{t,1}, \cdots, P_{t,j}, \cdots, P_{t,m}\}$.
  $D_t \leftarrow k$.
  $C_{t,k} \leftarrow C_{t,k} + 1$.
end for
$t_1, t_2 \leftarrow 0, 0$.
while $t_1 + w_1 \leq n$ **do**
  if $C_{i,j} > 0$ **then**
    $t_2 \leftarrow t_1$.
    $r \leftarrow (C_{t_1+w_1,D_{t_1}} - C_{t_1,D_{t_1}})/w_1$.
    while $r > \mu$ and $t_2 + 2 \cdot w_1 \leq n$ **do**
      $t_2 \leftarrow t_2 + w_1$.
      $r \leftarrow (C_{t_2+w_1,D_{t_1}} - C_{t_2,D_{t_1}})/w_1$.
    end while
    $r \leftarrow (C_{t_2,D_{t_1}} - C_{t_2-w_2,D_{t_1}})/w_2$.
    while $r \leq \mu$ and $t_2 > t_1$ **do**
      $t_2 \leftarrow t_2 - w_2$.
      $r \leftarrow (C_{t_2,D_{t_1}} - C_{t_2-w_2,D_{t_1}})/w_2$.
    end while
    if $t_2 > t_1$ **then**
      A.insert(D_{t_1})
      S.insert(t_1)
      E.insert(t_2)
      $t_1 \leftarrow t_2$.
    end if
  else
    $t_1 \leftarrow t_1 + 1$.
  end if
end while

Output: A – actions, S – starting frames, E – ending frames.

2) CMU Mocap: The CMU motion capture dataset is the largest publicly available motion capture dataset. It provides motion capture data for 144 different subjects. A large spectrum of movements are performed, including everyday movements (e.g., walking, running) as well as sport movements (e.g., climbing, dancing). Motions are recorded using a Vicon motion capture system that records the poses with 120 Hz. Human body 3D coordinates are provided for each frame. Following [21], the movements are categorized into 45 classes. The entire dataset has 3 train/test splits and the averaged recognition accuracy is reported.

3) PKU-MMD: The PKU-MMD dataset [50] is a new large-scale 3D dataset to facilitate study on action detection. It has 51 action categories, with 41 daily activities and 10 interaction actions. This dataset contains 1076 videos, and has 3,000 minutes and 5,400,000 frames totally. Each video has more than 20 action instances performed by 66 subjects recorded by 3 camera views. The ages of the subjects are between 18 and 40. The dataset is collected by Kinect v2 sensor and RGB frame, depth map and skeleton data are provided. We follow the same cross-subject and cross-view evaluations [50] and report the mean average precision (mAP) of different actions.

B. Implementation Details

To allow for batch learning, we convert the sequence to a fixed length of T. If the input sequence is longer than T, we sample subsequences from the beginning to the end with an interval of $T/2$ and average the predicted scores of subsequences. Otherwise, we fill it with zeros at the beginning. For a dataset, T should be larger than the length of most sequences to reduce loss of information caused by sampling. Here, $T = 200$ for the *PKU-MMD* and $T = 100$ for the other two datasets. The number of neurons of each recurrent layer is 512. The dropout ratio is 0.5 and the temporal dropout ratio is 0.05. The networks are trained using stochastic gradient descent with a batch size of 256. The momentum is 0.9, and weight decay is not applied. The learning rate is initialized as 0.01, and decreases by 30% after every 60 training epochs. The implementation of our proposed networks is based on Theano and one NVIDIA TITAN X GPU is used to run the experiments.

For action detection, since the gap clips without actions are a bit long, during training, we only sample subsequences from the T frame before the starting frame to the T frame after the ending frame of a particular action. For the multi-scale sliding window algorithm, the large window size $w_1 = 15$, the small window size $w_2 = 2$, and the probability threshold $\mu = 0.1$. The detection is correct when the predicted action class is true and the overlapping ratio between the predicted interval and the groundtruth interval exceeds a threshold θ. For evaluation, the default threshold of overlapping ratio $\theta = 0.5$. To evaluate the robustness of the proposed method, we analyze the sensitivities of parameters in Section IV-E.

C. Results of Action Recognition

For action recognition, we evaluate the proposed network on two large-scale datasets: the *NTU RGB+D* and the *CMU mocap*. We first compare the networks trained from different primitive inputs (e.g., joints, edges and surfaces), then compare our results with the state-of-the-arts reported in the literature.

1) Comparison Between Models: Table I summarizes the results of joints, edges and surfaces, as well as the combined results by averaging the predicted scores. Here, *Joints* denotes result of the proposed network by feeding the joints, *Joints + Edges* denotes the average of predictions of joints and edges, and *Joints + Edges + Surfaces* denotes the averaged prediction of the three individual results, and so on.

When comparing the individual results, we can see that *Edges* significantly outperforms the others and achieves the state-of-the-art results on all evaluations. *Surfaces* beats *Joints* on the *NTU RGB+D* dataset, but gets inferior result on the *CMU mocap* dataset. The results indicate that although *Edges*

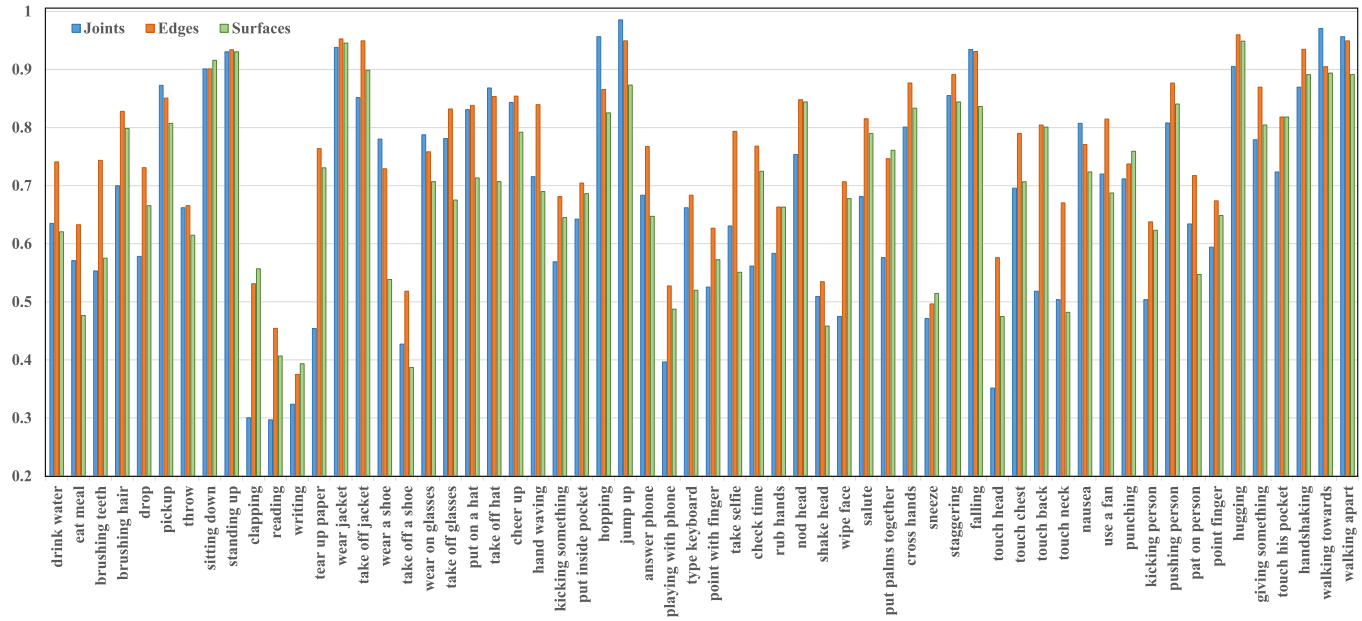


Fig. 5. Classification accuracy for each action on the NTU RGB+D dataset. There are 60 action classes including both individual actions and mutual actions.

TABLE I

EMPIRICAL EVALUATION OF THE PROPOSED METHODS FOR ACTION RECOGNITION. THE MEAN CLASSIFICATION ACCURACY IS REPORTED

Method	NTU RGB+D		CMU
	Cross-subject	Cross-view	mocap
Joints	68.2	74.5	83.7
Edges	**76.1**	**86.5**	**85.0**
Surfaces	69.7	75.3	81.2
Joints + Edges	77.8	87.4	85.8
Edges + Surfaces	78.3	86.6	85.2
Joints + Surfaces	75.9	81.2	84.9
Joints + Edges + Surfaces	**79.5**	**87.6**	**86.1**

and *Surfaces* are obtained through simple vector operations from *Joints*, the RNN based networks could hardly capture these information. Indeed, the RNN networks only learn the temporal dynamics of the individual joints, while the spatial configurations of related joints within the same frame are neglected [26]. Thus, how we present the data to the network is essential to learn better representations for action recognition.

While combing the results of joints, edges and surfaces, the performances are even higher. For example, on the NTU RGB+D dataset, the results of *Joints* + *Surfaces* are 6.2% and 5.9% higher than the best results of the two individuals for cross-subject evaluation and cross-view evaluation, respectively. For all evaluations, *Joints* + *Edges* + *Surfaces* yields the best performance, and consistently outperforms other approaches. The results demonstrate the relative geometries among joints (representations of *Edges, Surfaces*) reflect the distinctive characteristics of actions, and are complementary with the temporal dynamics of joints.

2) Accuracies of Different Actions: To investigate the difference of recognition rates of different actions, we depict and compare the accuracies of the actions. Figure 5 shows the results of cross-subject evaluation on the NTU RGB+D dataset. For most actions, *Edges* gets the highest accuracies. But for actions, i.e., *sitting down*, *clapping*, *writing*, *put palms together*, *sneeze* and *punching*, *Surfaces* gets the best results. For actions, i.e., *pick up*, *wear a shoe*, *wear on glasses*, *take off hat*, *hopping*, *jump up*, *falling*, *nausea*, *walking towards* and *walking apart*, *Joints* has the highest results. These findings indicate that although *Edges* generally performs better than *Joints* and *Surfaces*, the three representations are complementary for different actions.

3) Comparison With the State-of-the-Arts: To demonstrate the effectiveness of our approach, we compare it with the methods in the recent literature. We choose the method of *Joints* + *Edges* + *Surfaces* and denote it as *Beyond joints*.

Table II shows the results on the *NTU RGB+D* dataset. We first compare our method with traditional methods [56]–[59]. We observe that our performance is significantly higher, which shows the superiority of learning based approaches over the methods relying on handcrafted features. Then our method is compared with other deep learning approaches based on RNN. Our approach significantly outperforms HBRNN [18] and Part-aware LSTM [23], both of which only model temporal dynamics of joints. Moreover, our method performs much better than the newest methods, including STA-LSTM [25], GCA-LSTM [41] and Two-stream RNN [26], which leverage both temporal dynamics and spatial configurations of joints. It should be noted that we do not compare our approach with some convolutional neural network (CNN) based approaches (e.g., [33], [53]).

The results on the *CMU mocap* dataset are shown in Table III. Our approach shows considerable improvements over the state-of-the-art approaches, outperforming HBRNN [18] and Co-occurrence LSTM [21] by 11.1% and 5.1%, respectively.

TABLE II

COMPARISON WITH THE STATE-OF-THE-ART METHODS ON THE NTU RGB+D DATASET

Method	NTU RGB+D	
	Cross-subject	Cross-view
Lie Group [57]	50.1	52.8
LieNet [58]	61.3	67.0
Skeletal Quads [59]	38.6	41.4
FTP Dynamic [60]	60.2	65.2
HBRNN [18]	59.1	64.0
Deep LSTM [23]	62.9	70.3
Part-aware LSTM [23]	62.9	70.3
Trust Gate ST-LSTM [24]	69.2	77.7
STA-LSTM [25]	73.4	81.2
GCA-LSTM [41]	74.4	82.8
Two-stream RNN [26]	71.3	79.5
Multi-task RNN [44]	–	82.6
Beyond joints	**79.5**	**87.6**

TABLE III

COMPARISON WITH THE STATE-OF-THE-ART METHODS ON THE CMU MOTION CAPTURE DATASET

Method	CMU mocap
HBRNN [18]	75.0
Deep LSTM [21]	79.5
Co-occurrence LSTM [21]	81.0
Beyond joints	**86.1**

TABLE IV

EMPIRICAL EVALUATION OF THE PROPOSED METHODS FOR ACTION DETECTION. THE MAP PERCENTAGE IS REPORTED

Method	PKU-MMD	
	Cross-subject	Cross-view
Joints	76.5	**89.1**
Edges	**77.8**	88.3
Surfaces	72.1	83.8
Joints + Edges	79.6	90.7
Edges + Surfaces	79.4	89.7
Joints + Surfaces	79.2	90.0
Joints + Edges + Surfaces	**81.1**	**91.1**

In summary, for large-scale action recognition based on skeletons, our proposed network learning from primitive geometries achieves considerably superior result over the alternative approaches based on isolated joints of human body. Our approach works effectively for skeletons estimated from range sensors (e.g., Kinect) as well as skeletons collected by motion capture systems (e.g., MoCap).

D. Results of Action Detection

Large-scale 3D action detection is a relatively new research area, and there are less public benchmarks than those of action recognition. We evaluate our approaches on the recent PKU-MMD dataset and compare results with the state-of-the-arts.

1) Comparison Between Models: Table IV summarizes the results of *Joints*, *Edges*, *Surfaces* and the combined models. The abbreviations (e.g., *Joints* + *Edges*) are similar to those of action recognition. For the individual results, both *Joints* and *Edges* yield good performances, outperforming *Surfaces* by considerable margins. For action detection, *Edges* is comparable with *Joints*. It beats *Joints* for the cross-subject evaluation, but is inferior to *Joints* for the cross-view evaluation. We also observe that the combined results are consistently better than the individual results. For both evaluations, *Joints* + *Edges*, *Edges* + *Surfaces*, *Joints* + *Surfaces* all achieve better performances than the best of the individual method. Specifically, *Joints* + *Edges* + *Surfaces* yields the best results. For the cross-subject and cross-view evaluations, it outperforms *Joints* by 4.6% and 2.0%, respectively. These results further confirm our findings that *Joints*, *Edges* and *Surfaces* are all effective and complimentary with each other.

2) Visualizations of Detection: We also provide visualizations of action detection. Figure 6 shows the results of samples on the validation set under the cross-subject evaluation on the *PKU-MMD* dataset. Each sample has thousands of frames and tens of actions. An action interval lasts from tens to hundreds of frames. The number of blank frames between two adjacent action intervals varies from a few to hundreds. For the sake of convenience, let (a, b, c) be the action interval which starts from frame b and ends at frame c, and the class index is a. For most actions, the action labels are correctly predicted and the predicted intervals are near the ground truth. For the precisions, most values are greater than 85%, and for *Edges*, there are five actions whose precisions are greater than 99%. In contrast, the values of recalls are much smaller, and most of the values are around 50%.

The mistakes mainly come from three aspects. First, there are some action intervals whose classes are wrongly predicted. For example, for the ground truth $(40, 5738, 5825)$ in sample (a), *Joints* predicts $(8, 5801, 5830)$, *Edges* predicts $(40, 5789, 5799)$ and $(10, 5801, 5830)$, and *Surfaces* predicts $(9, 5801, 5845)$. Second, there are some redundant and incorrect predictions. For example, no action occurs from frame 1 to 130 in sample (b), but *Joints* and *Edges* predict $(44, 1, 71)$, and *Surfaces* predicts $(28, 1, 37)$ and $(44, 38, 78)$. Third, when the ground truth action interval is a bit long, it could be mistakenly predicted as a set of action intervals. For example, for the ground truth $(48, 2913, 3649)$ in sample (c), *Joints* predicts $(48, 2914, 3325)$, $(48, 3349, 3400)$ and $(43, 3401, 3499)$, *Edges* predicts $(37, 2911, 3000)$, $(48, 3001, 3273)$, $(49, 3354, 3373)$, $(23, 3401, 3460)$ and $(31, 3463, 3486)$, and *Surfaces* predicts $(48, 2918, 3250)$, $(48, 3360, 3407)$ and $(43, 3409, 3498)$. These weaknesses suggest that the accuracy of frame-wise classification plays a vital role in the results of detection, and exploring more discriminative models for classification is a future direction.

3) Comparison With the State-of-the-Arts: In Table V, we compare our approach with the recent state-of-the-art methods in the literature. Here, we denote *Joints* + *Edges* + *Surfaces* as *Beyond joints*. Our approach gains remarkable margins over the state-of-the-art methods. For example, when the overlapping ratio threshold $\theta = 0.1$, our results are 43.0% and 47.7% higher than the reported results in [49] for cross-subject and cross-view evaluations, respectively. When $\theta = 0.5$, the margins are even more substantial. The result of our approach when $\theta = 0.1$ is nearly 5.2% higher than that when $\theta = 0.5$. It should be noted that JCRRNN [49]

Fig. 6. Visualizations of action detection. Here, we show results of five samples on the validation set of the *PKU-MMD* dataset. The x-axis represents the frame index, and each line segment along the time axis denotes an action interval. The number above the segment denotes the action class index.

TABLE V

COMPARISON OF MAP WITH THE STATE-OF-THE-ART METHODS ON THE PKU-MMD DATASET. HERE, θ IS THE OVERLAPPING RATIO THRESHOLD

Method	θ	Cross-subject	Cross-view
JCRRNN [49]	0.1	45.2	69.9
	0.5	32.5	53.3
BLSTM [50]	0.1	47.9	54.5
	0.5	13.0	15.9
STA-LSTM [25]	0.1	44.4	47.6
	0.5	13.1	15.5
Beyond joints	0.1	**87.4**	**95.3**
	0.5	81.1	91.1

is an online action detection method by joint classification and regression, and our method is dedicated to offline action detection. The *Joints* method won the second place and outperformed that of the third place by nearly 20% in large scale 3D human activity analysis challenge in depth videos in IEEE international conference on multimedia & expo (ICME) 2017 workshop.[1]

[1]http://www.icst.pku.edu.cn/struct/icmew2017/result.html

E. Discussion

We conduct comprehensive analyses including the fusion weights, convergence rates, model structures as well as parameter sensitivities.

1) Analysis of Weights: In Section IV-C, we find that the results can be improved by averaging the predictions of *Joints, Edges, Surfaces*. Here, we use the weighted average prediction and analyze the sensitivities of the weights. The results on the *NTU RGB+D* dataset are illustrated in Figure 7. For both evaluations, we observe that the best weights are near 0.5, which indicate that *Joints, Edges, Surfaces* contain equally important information of actions. For *Joints* + *Edges*, the accuracy increases slowly when the weight of *Joints* is smaller than 0.5 but drops quickly when this weight is greater than 0.6. For *Edges* + *Surfaces*, the accuracy increases steadily with the increase of the weight of *Edges* before reaching its highest value, then decreases a little when the weight is close to 1. For *Joints* + *Surfaces*, the accuracy first increases then decreases at a considerable rate. We can conclude that the accuracy is a bit sensitive to the weight, and the highest value is reached when the weight is around 0.5.

2) Analysis of Convergence Rates: To compare the convergence rates of *Joints, Edges, Surfaces*, we plot the accuracies

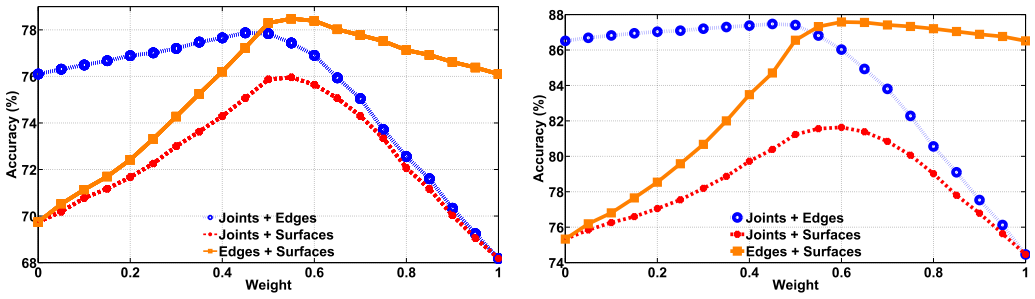


Fig. 7. Analysis of the weights by combining the predictions of *Joints, Edges, Surfaces* on the NTU RGB+D dataset. Each curve is the combined result of two individuals, and the weight corresponds to the former. (a) Cross-subject. (b) Cross-view.

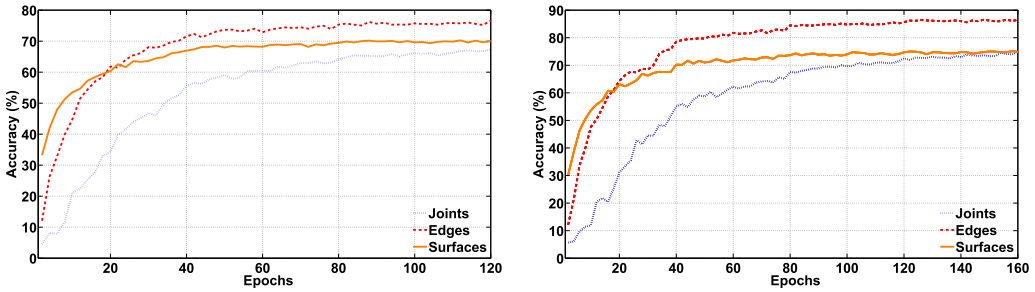


Fig. 8. Accuracies on the validation set with the number of training epochs on the NTU RGB+D dataset for both cross-subject and cross-view evaluations. (a) Cross-subject. (b) Cross-view.

TABLE VI

FOR THE BACKBONE OF THE PROPOSED NETWORK, WE COMPARE DIFFERENT RNN STRUCTURES

Method	Accuracy (%)
Three layers of LSTM	63.4
Three layers of BiGRU	65.7
Two layers of BiLSTM	68.3
Four layers of BiLSTM	65.8
Three layers of BiLSTM	68.2

on the validation set during training. The results on the *NTU RGB+D* dataset are shown in Figure 8. We can observe that all the curves converge to their maximum values, but the convergence rate of *Edges* is the fastest. *Surfaces* also enjoys a fast convergence rate. In contrast, *Joints* converges much slower when compared with *Edges*, and the number of epochs to reach its maximum value is nearly twice that of *Edges*.

3) Comparison of RNN Structures: As the backbone of our network consists of three bidirectional LSTM (BiLSTM) layers, here we discuss the other backbone structures based on RNN. Without loss of generality, we only use joints as input, the results of cross-subject evaluation on the *NTU RGB+D* dataset are shown in Table VI. We find that both three layers of BiLSTM and two layers of BiLSTM outperform the other alternatives for action recognition. In this paper, we choose three layers of BiLSTM due to the large representational capacity of deep networks.

4) Ablation Studies: To investigate of the effects of the viewpoint transformation (View TF) layer and the temporal dropout (TDP) layer, we conduct ablation studies for the task

TABLE VII

INVESTIGATION OF THE EFFECTS OF THE VIEWPOINT TRANSFORMATION (VIEW TF) LAYER AND TEMPORAL DROPOUT (TDP) LAYER FOR ACTION RECOGNITION ON THE *NTU RGB+D* DATASET

Method		Joints	Edges	Surfaces
Cross-subject	Original	**68.2**	**76.1**	**69.7**
	No View TF	68.2	75.5	68.8
	No TDP	68.1	73.1	65.8
Cross-view	Original	74.5	**86.5**	**75.3**
	No View TF	77.3	79.4	57.8
	No TDP	**79.8**	85.2	70.6

of action recognition. We use *Original* to denote the network in Figure 4(a). The structure without the viewpoint transformation layer is denoted as *No View TF*, and the structure without the temporal dropout layers is denoted as *No TDP*. The results on the *NTU RGB+D* dataset are provided in Table VII. We observe that for *Edges* and *Surfaces*, both View TF layer and TDP layer could considerably improve the accuracies. But for *Joints*, for the cross-view evaluation, the TDP layer seems to have a negative effect on the result, and for cross-subject evaluation, both layers have little influences.

5) Sensitivity Analysis: In Section III-D for action detection, we assign default values to the parameters of the multi-scale sliding window algorithm, i.e., the large window size w_1, the small window size w_2, and the probability threshold μ. Here we evaluate the sensitivities of these parameters by varying one parameter from a wide range while keeping the others with the default values. We plot the values of mAP for the cross-subject evaluation on the *PKU-MMD* dataset.

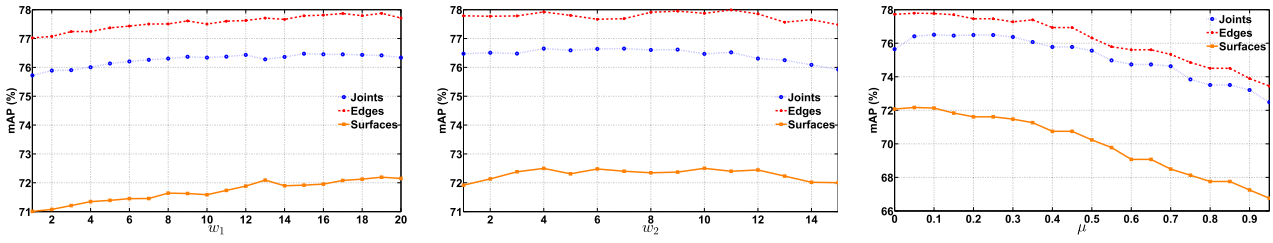


Fig. 9. Sensitivity analysis of parameters of the multi-scale sliding window algorithm for action detection on the *PKU-MMD* dataset. Here, $w_1 \in \{1, 2, \cdots, 20\}$, $w_2 \in \{1, 2, \cdots, 15\}$, $\mu \in \{0, 0.1, \cdots, 0.9\}$. (a) The large window size. (b) The small window size. (c) The probability threshold.

Figure 9(a) shows the results w.r.t. the large window size w_1. We can see that the mAP increases slowly with a larger value of w_1 when $w_1 < 15$, and keeps a high value when $w_1 > 15$. The results indicate that a relatively large value of w_1 is preferred for good action detection. Figure 9(b) shows the results w.r.t. the small window size w_2. We find that the performance maintains a high value within a broad range as long as $w_2 < w1$. Figure 9(c) shows the results w.r.t. the probability threshold μ. The mAP decreases with the larger value of μ when $\mu > 0.2$, and the best performance is reached when μ is near 0.1. We conclude that our detection algorithm is not sensitive to the sizes of the two windows, and a bit sensitive to the probability threshold when the value is high (e.g., greater than 0.2).

V. Conclusions

In this paper, we aim to learn representations from primitive geometries of human skeletons, i.e., joints, edges and surfaces. For action recognition, we propose a novel RNN based architecture to accommodate the three inputs. For action detection, we first perform frame-wise action classification, then design a multi-scale sliding window search algorithm to generate detection results. Experiments on large-scale datasets show that joints, edges and surfaces are all effective and complimentary with each other. For both tasks, our approach significantly outperforms the current state-of-the-arts. Moreover, the optimal weight to combine two kinds of geometries (e.g., joints and edges) is near 0.5. We also find that the three geometries all have their preferred actions with high recognition accuracies. While comparing the convergence rates, we show that the input of edges converges fastest. Our experiments demonstrate that geometric relations contribute to action recognition, and beyond joints representations (e.g., edges and surfaces) are essential for excellent performance. In the future, we will explore deeper into the geometric relations and consider to leverage temporal dynamics of the structure of skeletons for action recognition and detection.

References

[1] A. Dosovitskiy *et al.*, "FlowNet: Learning optical flow with convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 2758–2766.

[2] J. Yu, C. Hong, Y. Rui, and D. Tao, "Multitask autoencoder model for recovering human poses," *IEEE Trans. Ind. Electron.*, vol. 65, no. 6, pp. 5060–5068, Jun. 2018.

[3] J. Zhang, J. Yu, and D. Tao, "Local deep-feature alignment for unsupervised dimension reduction," *IEEE Trans. Image Process.*, vol. 27, no. 5, pp. 2420–2432, May 2018.

[4] Y. Huang, Z. Wu, L. Wang, and T. Tan, "Feature coding in image classification: A comprehensive study," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 3, pp. 493–506, Mar. 2014.

[5] M. Asadi-Aghbolaghi *et al.*, "Deep learning for action and gesture recognition in image sequences: A survey," in *Gesture Recognition*. Springer, 2017, pp. 539–578. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-319-57021-1_19 and https://www.researchgate.net/publication/318533769_Deep_Learning_for_Action_and_Gesture_Recognition_in_Image_Sequences_A_Survey

[6] R. Poppe, "A survey on vision-based human action recognition," *Image Vis. Comput.*, vol. 28, no. 6, pp. 976–990, 2010.

[7] B. Zhang, L. Wang, Z. Wang, Y. Qiao, and H. Wang, "Real-time action recognition with deeply transferred motion vector CNNs," *IEEE Trans. Image Process.*, vol. 27, no. 5, pp. 2326–2339, May 2018.

[8] B. Zhang, Y. Yang, C. Chen, L. Yang, J. Han, and L. Shao, "Action recognition using 3D histograms of texture and a multi-class boosting classifier," *IEEE Trans. Image Process.*, vol. 26, no. 10, pp. 4648–4660, Oct. 2017.

[9] C. Jia, M. Shao, S. Li, H. Zhao, and Y. Fu, "Stacked denoising tensor auto-encoder for action recognition with spatiotemporal corruptions," *IEEE Trans. Image Process.*, vol. 27, no. 4, pp. 1878–1887, Apr. 2017.

[10] L. Wang, Y. Qiao, and X. Tang, "Action recognition with trajectory-pooled deep-convolutional descriptors," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 4305–4314.

[11] G. Varol, I. Laptev, and C. Schmid, "Long-term temporal convolutions for action recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 6, pp. 1510–1517, Jan. 2018, doi: 10.1109/TPAMI.2017.2712608.

[12] R. Girdhar, D. Ramanan, A. Gupta, J. Sivic, and B. Russell, "ActionVLAD: Learning spatio-temporal aggregation for action classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 3165–3174.

[13] R. Girdhar and D. Ramanan, "Attentional pooling for action recognition," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 34–45.

[14] J. K. Aggarwal and L. Xia, "Human activity recognition from 3D data: A review," *Pattern Recognit. Lett.*, vol. 48, pp. 70–80, Oct. 2014.

[15] G. Johansson, "Visual perception of biological motion and a model for its analysis," *Perception Psychophys.*, vol. 14, no. 2, pp. 201–211, 1973.

[16] J. Shotton *et al.*, "Real-time human pose recognition in parts from single depth images," *Commun. ACM*, vol. 56, no. 1, pp. 116–124, Jan. 2013.

[17] W. Li, Z. Zhang, and Z. Liu, "Action recognition based on a bag of 3D points," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2010, pp. 9–14.

[18] Y. Du, W. Wang, and L. Wang, "Hierarchical recurrent neural network for skeleton based action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 1110–1118.

[19] J. Zhang, W. Li, P. O. Ogunbona, P. Wang, and C. Tang, "RGB-D-based action recognition datasets: A survey," *Pattern Recognit.*, vol. 60, pp. 86–105, Dec. 2016.

[20] L. L. Presti and M. La Cascia, "3D skeleton-based human action classification: A survey," *Pattern Recognit.*, vol. 53, pp. 130–147, May 2016.

[21] W. Zhu *et al.*, "Co-occurrence feature learning for skeleton based action recognition using regularized deep LSTM networks," in *Proc. 30th AAAI Conf. Artif. Intell.*, 2016, pp. 1–7.

[22] V. Veeriah, N. Zhuang, and G.-J. Qi, "Differential recurrent neural networks for action recognition," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 4041–4049.

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

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

[25] S. Song, C. Lan, J. Xing, W. Zeng, and J. Liu, "An end-to-end spatio-temporal attention model for human action recognition from skeleton data," in *Proc. 31st AAAI Conf. Artif. Intell.*, 2017, pp. 1–8.

[26] H. Wang and L. Wang, "Modeling temporal dynamics and spatial configurations of actions using two-stream recurrent neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 3633–3642.

[27] C. Ellis, S. Z. Masood, M. F. Tappen, J. J. LaViola, Jr., and R. Sukthankar, "Exploring the trade-off between accuracy and observational latency in action recognition," *Int. J. Comput. Vis.*, vol. 101, no. 3, pp. 420–436, 2013.

[28] M. Müller, T. Röder, and M. Clausen, "Efficient content-based retrieval of motion capture data," *ACM Trans. Graph.*, vol. 24, no. 3, pp. 677–685, 2005.

[29] F. Ofli, R. Chaudhry, G. Kurillo, R. Vidal, and R. Bajcsy, "Sequence of the most informative joints (SMIJ): A new representation for human skeletal action recognition," *J. Vis. Commun. Image Represent.*, vol. 25, no. 1, pp. 24–38, 2014.

[30] L. Seidenari, V. Varano, S. Berretti, A. Del Bimbo, and P. Pala, "Recognizing actions from depth cameras as weakly aligned multi-part bag-of-poses," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2013, pp. 479–485.

[31] R. Chaudhry, F. Ofli, G. Kurillo, R. Bajcsy, and R. Vidal, "Bio-inspired dynamic 3D discriminative skeletal features for human action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2013, pp. 471–478.

[32] R. Slama, H. Wannous, M. Daoudi, and A. Srivastava, "Accurate 3D action recognition using learning on the grassmann manifold," *Pattern Recognit.*, vol. 48, no. 2, pp. 556–567, 2015.

[33] Q. Ke, M. Bennamoun, S. An, F. Sohel, and F. Boussaid, "A new representation of skeleton sequences for 3D action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 4570–4579.

[34] H. Rahmani and M. Bennamoun, "Learning action recognition model from depth and skeleton videos," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Oct. 2017, pp. 5833–5842.

[35] I. Lee, D. Kim, S. Kang, and S. Lee, "Ensemble deep learning for skeleton-based action recognition using temporal sliding LSTM networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 1012–1020.

[36] W. Li, L. Wen, M.-C. Chang, S. N. Lim, and S. Lyu, "Adaptive RNN tree for large-scale human action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Oct. 2017, pp. 1453–1461.

[37] A. H. Ruiz, L. Porzi, S. R. Bulò, and F. Moreno-Noguer, "3D CNNs on distance matrices for human action recognition," in *Proc. ACM Multimedia Conf.*, 2017, pp. 1087–1095.

[38] Y. Yang, C. Deng, S. Gao, W. Liu, D. Tao, and X. Gao, "Discriminative multi-instance multitask learning for 3D action recognition," *IEEE Trans. Multimedia*, vol. 19, no. 3, pp. 519–529, Mar. 2017.

[39] Y. Du, Y. Fu, and L. Wang, "Representation learning of temporal dynamics for skeleton-based action recognition," *IEEE Trans. Image Process.*, vol. 25, no. 7, pp. 3010–3022, Jul. 2016.

[40] S. Zhang, X. Liu, and J. Xiao, "On Geometric features for skeleton-based action recognition using multilayer LSTM networks," in *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, Mar. 2017, pp. 148–157.

[41] J. Liu, G. Wang, P. Hu, L.-Y. Duan, and A. C. Kot, "Global context-aware attention LSTM networks for 3D action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 3671–3680.

[42] J. Liu, G. Wang, L.-Y. Duan, K. Abdiyeva, and A. C. Kot, "Skeleton-based human action recognition with global context-aware attention LSTM networks," *IEEE Trans. Image Process.*, vol. 27, no. 4, pp. 1586–1599, Apr. 2017.

[43] P. Zhang, C. Lan, J. Xing, W. Zeng, J. Xue, and N. Zheng, "View adaptive recurrent neural networks for high performance human action recognition from skeleton data," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 2136–2145.

[44] H. Wang and L. Wang, "Learning content and style: Joint action recognition and person identification from human skeletons," *Pattern Recognit.*, vol. 81, pp. 23–35, Sep. 2018.

[45] S. Nowozin and J. Shotton, "Action points: A representation for low-latency online human action recognition," Microsoft Res. Cambridge, Cambridge, U.K., Tech. Rep. MSR-TR-2012-68, 2012.

[46] A. Sharaf, M. Torki, M. E. Hussein, and M. El-Saban, "Real-time multi-scale action detection from 3D skeleton data," in *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, Jan. 2015, pp. 998–1005.

[47] X. Zhao, X. Li, C. Pang, Q. Z. Sheng, S. Wang, and M. Ye, "Structured streaming skeleton—A new feature for online human gesture recognition," *Trans. Multimedia Comput., Commun., Appl.*, vol. 11, no. 1s, 2014, Art. no. 22.

[48] M. Zanfir, M. Leordeanu, and C. Sminchisescu, "The moving pose: An efficient 3D kinematics descriptor for low-latency action recognition and detection," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 2752–2759.

[49] Y. Li, C. Lan, J. Xing, W. Zeng, C. Yuan, and J. Liu, "Online human action detection using joint classification-regression recurrent neural networks," in *Proc. Eur. Conf. Comput. Vis.*, Springer, 2016, pp. 203–220.

[50] C. Liu, Y. Hu, Y. Li, S. Song, and J. Liu, "PKU-MMD: A large scale benchmark for skeleton-based human action understanding," in *Proc. Workshop Vis. Anal. Smart Connected Communities*, 2017, pp. 1–8.

[51] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[52] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Trans. Signal Process.*, vol. 45, no. 11, pp. 2673–2681, Nov. 1997.

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

[54] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov. (2012). "Improving neural networks by preventing co-adaptation of feature detectors." [Online]. Available: https://arxiv.org/abs/1207.0580

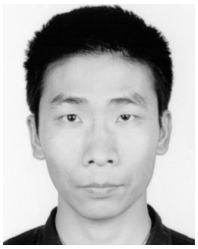
[55] J. Tompson, R. Goroshin, A. Jain, Y. LeCun, and C. Bregler, "Efficient object localization using convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 648–656.

[56] R. Vemulapalli, F. Arrate, and R. Chellappa, "Human action recognition by representing 3D skeletons as points in a lie group," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 588–595.

[57] Z. Huang, C. Wan, T. Probst, and L. Van Gool, "Deep learning on lie groups for skeleton-based action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 1243–1252.

[58] G. Evangelidis, G. Singh, and R. Horaud, "Skeletal quads: Human action recognition using joint quadruples," in *Proc. Int. Conf. Pattern Recognit.*, Aug. 2014, pp. 4513–4518.

[59] J.-F. Hu, W.-S. Zheng, J. Lai, and J. Zhang, "Jointly learning heterogeneous features for RGB-D activity recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 2186–2200.



Hongsong Wang received the B.S. degree in automation from the Huazhong University of Science and Technology in 2013. He is currently pursuing the Ph.D. degree with the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing, China. His research interests include action recognition, video classification, and deep learning.

Liang Wang (SM'09) received the B.S. and M.S. degrees from Anhui University in 1997 and 2000, respectively, and the Ph.D. degree from the Institute of Automation, Chinese Academy of Sciences (CASIA), in 2004. From 2004 to 2010, he was a Research Assistant with Imperial College London, U.K., and Monash University, Australia, a Research Fellow with the University of Melbourne, Australia, and a Lecturer with the University of Bath, U.K., respectively. He is currently a Full Professor of the Hundred Talents Program at the National Laboratory of Pattern Recognition, CASIA. His major research interests include machine learning, pattern recognition, and computer vision. He has widely published at highly-ranked international journals such as the IEEE TPAMI and the IEEE TIP, and leading international conferences such as CVPR, ICCV, and ICDM. He is currently an IAPR Fellow. He is an Associate Editor of the IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS: PART B.