

Stacked Spatio-Temporal Graph Convolutional Networks for Action Segmentation

Pallabi Ghosh

University of Maryland
pallabi@umd.edu

Yi Yao

SRI International
yi.yao@sri.com

Larry S. Davis

University of Maryland
lsd@umiacs.umd.edu

Ajay Divakaran

SRI International
ajay.divakaran@sri.com

Abstract

We propose novel Stacked Spatio-Temporal Graph Convolutional Networks (Stacked-STGCN) for action segmentation, i.e., predicting and localizing a sequence of actions over long videos. We extend the Spatio-Temporal Graph Convolutional Network (STGCN) originally proposed for skeleton-based action recognition to enable nodes with different characteristics (e.g., scene, actor, object, action), feature descriptors with varied lengths, and arbitrary temporal edge connections to account for large graph deformation commonly associated with complex activities. We further introduce the stacked hourglass architecture to STGCN to leverage the advantages of an encoder-decoder design for improved generalization performance and localization accuracy. We explore various descriptors such as frame-level VGG, segment-level I3D, RCNN-based object, etc. as node descriptors to enable action segmentation based on joint inference over comprehensive contextual information. We show results on CAD120 (which provides pre-computed node features and edge weights for fair performance comparison across algorithms) as well as a more complex real-world activity dataset, Charades. Our Stacked-STGCN in general achieves improved performance over the state-of-the-art for both CAD120 and Charades. Moreover, due to its generic design, Stacked-STGCN can be applied to a wider range of applications that require structured inference over long sequences with heterogeneous data types and varied temporal extent.

1. Introduction

Inspired by the success of convolutional neural networks (on either grid-like or sequential data), graph neural networks (GNNs) including graph convolutional networks (GCNs) have been developed and have demonstrated improvements over a number of machine learning/computer vision tasks such as node classification [19], community clustering [4], link prediction [41], 3D point cloud segmentation [50], etc.

As a special case of GCNs, spatio-temporal graph con-

volutional networks (STGCN), have been proposed for skeleton-based activity recognition [56]. STGCN defines the node descriptor as the location (x and y) and confidence of detected joints of human body (i.e., the length of the node descriptor is three), leverages the spatial connection between these joints, and connects the same joints across consecutive time steps to form a spatio-temporal graph. STGCN has shown performance improvements on Kinetics-skeleton [18] and NTU RGB+D [42] datasets via exploiting primarily actor poses.

In addition to actor poses, there frequently exist abundant contextual cues that would help in recognizing an action. Leveraging these contextual cues becomes critical for improving accuracy and robustness of action recognition, especially for actions with subtle changes in the actor's movement/pose.

A graph is an intuitive data structure to jointly represent various contextual cues (e.g., scene graph, situation recognition). Therefore, in this paper, we plan to construct a comprehensive spatio-temporal graph (STG) to jointly represent an action along with its associated actors, objects, and other contextual cues. Specifically, graph nodes represent actions, actors, objects, and scenes, spatial edges represent spatial (e.g., next to, on top of, etc.) and functional relationships (e.g., attribution, role, etc.) between two nodes with importance weights, and temporal edges represent temporal and causal relationships. We exploit a variety of descriptors in order to capture these rich contextual cues. In the literature, there exist various techniques such as situation recognition [24], object detection, scene classification, and semantic segmentation. The output of these networks provides embeddings that can serve as the node features of the proposed STGs.

We perform action segmentation on top of this spatio-temporal graph via stacked spatio-temporal graph convolution. Our STGCN stems from the networks originally proposed for skeleton-based action recognition [56] and introduces two major advancements as our innovations. First, as mentioned before, to accommodate various contextual cues, the nodes of our STG have a wide range of characteristics,



Figure 1. System overview. Different from the original STGCN based on human skeleton [56], our graph allows nodes of various types (such as actors, objects, and scenes) and with varied feature length. Our graph also supports flexible temporal connections (green lines) that can span multiple time steps, for example the connections among the actor nodes (blue nodes). Note that other nodes can have such temporal connections but are not depicted to avoid congested illustration. This spatio-temporal graph is fed into a stack of hourglass STGCN blocks to output a sequence of predicted actions observed in the video.

leading to the need for using descriptors with varied length. Second, our STG allows arbitrary edge connections (even fully connected graph as an extreme case) to account for the large amount of graph deformation caused by missed detections, occlusions, and emerging/disappearing objects. The enhanced representational capacity with arbitrary edge connections especially along the temporal axis enables accurate temporal localization of action boundaries compared to fixed temporal connection of consecutive frames. Therefore, we can apply stacked-STGCN for action segmentation which involves generating not only action category but also temporal locations of the starting and end boundaries of the identified action. Extension from recognition to segmentation is a non-trivial task since we have to perform per frame prediction so that we get the exact boundary of a particular action despite large variations in the temporal spans of actions and ambiguity between two consecutive actions.

Another innovation we introduce is the extended use of stacked hourglass architecture on graph data. Stacked hourglass networks have been applied to grid-like data with regular connections (e.g., images using CNNs) and shown improved results for a number of tasks such as human pose estimation [29], facial landmark localization [57], etc. They allow repeated upsampling and downsampling of features and combine these features at different scales, leading to better performance. We propose to extend this encoder-

decoder architecture to graph data with irregular connections. Different from CNN, STGCN (or more general GCN) employs adjacency matrices to represent irregular connections among nodes. To address this fundamental difference, we adapt the hourglass networks by adding extra steps to down-sample the adjacency matrices at each encoder level to match the compressed dimensions of that level.

In summary, the proposed Stacked-STGCN offers the following unique innovations: 1) joint inference over a rich set of contextual cues, 2) flexible graph configuration to support a wide range of descriptors with varied feature length and to account for large amounts of graph deformation over long video sequences, and 3) stacked hourglass architecture specifically designed for graph data with irregular connection. These innovations promise improved recognition/localization accuracy, robustness, and generalization performance for action segmentation over long video sequences. We demonstrate such improvements via our experiments on the CAD120 and Charades datasets.

2. Related Works

2.1. Neural Networks on Graphs

In recent years, there have been a number of research directions for applying neural networks on graphs. The original work by Scarselli *et al.*, referred to as the GNN, was an extension of the recursive neural networks and was used for sub-graph detection[40]. Later, GNNs were extended

and a mapping function was introduced to project a graph and its nodes to an Euclidean space with a fixed dimension [39]. In 2016, Li *et al.* used gated recurrent units and better optimization techniques to develop the Gated Graph Neural Networks [26]. GNNs have been used in a number of different applications like situation recognition [24], human-object interaction [25], webpage ranking[39, 40], mutagenesis[39], etc.

The literature also mentions a number of techniques that apply convolutions on graphs. Duvenaud *et al.* were one of the first to develop convolution operations for graph propagation [13] whereas Atwood and Towsley developed their own technique independently [2]. Defferrard *et al.* used approximation in spectral domain [8] based on spectral graph introduced by Hammond *et al.* [16]. In [19], Kipf and Welling proposed GCNs for semi-supervised classification based on similar spectral convolutions, but with further simplifications that resulted in higher speed and accuracy.

2.2. Action Recognition

Action recognition is a classic example of computer vision problems. Since the development of two-stream [47] and 3D convolution architecture, a series of works were studied, including TSN [52], ST-ResNet [59], I3D [5], P3D [35], R(1+2)D [51], T3D [9], S3D [54], TGM [31], etc. Another popular type of DNNs used for action recognition is the Recurrent Neural Network (RNN) including Long Short-Term Memory networks (LSTM). The structural-RNN (S-RNN) is one such method that uses RNNs on spatio-temporal graphs for action recognition [17]. S-RNN relies on two independent RNNs, namely nodeRNN and edgeRNN, for iterative spatial and temporal inference. In contrast, our Stacked-STGCN performs joint spatio-temporal inference over a rich set of contextual cues.

Recently, graph-based representation becomes a popular option for action recognition, for instance skeleton-based activity recognition using STGCN [56], Graph Edge Convolution Networks [60], and Neural Graph Matching Networks [15]. In [53], GCN is applied to space-time graphs extracted from video segments to produce an accumulative descriptor, which is later combined with the aggregated frame-level features to generate action predictions. Their work is similar to ours, but is used for classification and not segmentation. Furthermore, their graphs are formed based on object nodes only while ours are more general connecting different types of features like scene descriptor, human pose feature etc.

The most related work is STGCN originally proposed for skeleton-based activity recognition [56]. The nodes of the original STGCN are the skeletal joints, spatial connections depend on physical adjacency of these joints in the human body, and temporal edges connect joints of the same type (e.g., right wrist to right wrist) across one consecutive time step. The original STG is based on an oversimplified struc-

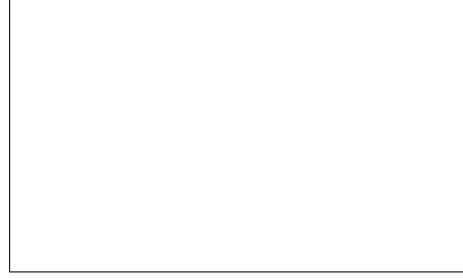


Figure 2. An illustration of spatio-temporal graphs. Each node v_i is represented by a feature vector denoted by f_i . The edge between node i and j has a weight $e_{i,j}$. These edge weights form the spatial and temporal adjacency matrices. Note that our spatio-temporal graph supports a large amount of deformation, such as missed detection (e.g., the actor node and the object 3 node) and emerging/disappearing nodes (e.g., the object 2 node).

ture for the variety and complexity our STG needs to handle in order to perform action segmentation with contextual cues and large graph deformation. Therefore, the original STGCN is not directly applicable.

2.3. Action Segmentation

Action segmentation presents a more challenging problem than action recognition in the sense that it requires identifying a sequence of actions with semantic labels and temporally localized starting and ending points of each identified actions [27, 11, 6]. Conditional Random Fields (CRFs) are traditionally used for temporal inference [28, 32, 43]. Language models and RNNs/LSTMs are also employed to leverage dependencies/correlations among actions to produce a long sequence of detected actions [37]. Lea *et al.* proposed temporal convolutional networks (TCNs) [22]. Later, a number of variations of TCNs were studied [10, 12, 23, 14]. Recently, weakly supervised approaches have gained increasing research interest to alleviate the demanding requirements on fully annotated video data [38, 49]. Most similar to our work is the graph parsing neural networks (GPNN) developed for the inference of human-object interactions as well as action segmentation [33]. However, GPNN relies on explicit object and actor detection whereas our graph-based inference operates on candidate proposals directly to avoid unrecoverable errors induced by missed object/actor detection.

3. Stacked Spatio-Temporal GCNs

The proposed Stacked-STGCN is illustrated in Figure 1. Related notations are given in 3.1. We describe the basic building block of Stacked-STGCN, i.e., generalized STGCN, in 3.2 and how to construct the stacked hourglass architecture in 3.3.

3.1. Graph Convolutional Networks

Let a graph be defined as $G(V, E)$ with vertices V and edges E (see Figure 2). Vertex features of length d^0 are



Figure 3. Illustration of two STGCN implementations to support graph nodes with varied feature length. (a) Additional convolution layers to convert node features with varied length to a fixed length. (b) Multiple spatial GCNs each for one cluster of nodes (nodes with the same color) with a similar feature length. These spatial GCNs convert features with varied length to a fixed length.

denoted as f_i for $i \in \{1, 2, \dots, N\}$ where N is the total number of nodes. Edge weights are given as e_{ij} where $e_{ij} \in [0, 1]$ and $i, j \in \{1, 2, \dots, N\}$. The graph operation at the l^{th} layer is defined as:

$$\mathbf{H}^{l+1} = g(\mathbf{H}^l, \mathbf{A}) = (\hat{\mathbf{D}}^{-1/2} \hat{\mathbf{A}} \hat{\mathbf{D}}^{-1/2} \mathbf{H}^l \mathbf{W}^l) \quad (1)$$

where \mathbf{W}^l and \mathbf{H}^l are the $d^l \times d^{l+1}$ weight matrix and $N \times d^l$ input matrix of the l^{th} layer, respectively. $\hat{\mathbf{A}} = \mathbf{I} + \mathbf{A}$ where $\mathbf{A} = [e_{ij}]$, $\hat{\mathbf{D}}$ is the diagonal node degree matrix of $\hat{\mathbf{A}}$, and g represents a non-linear activation function (e.g., ReLU).

3.2. Spatio-Temporal GCNs

STGCN is originally designed for skeleton-based action recognition [56]. We apply STGCN for action segmentation of long video sequences using frame-based action graphs extracted via situation recognition [24]. To accommodate additional application requirements, our STG differs fundamentally in two aspects. First, the original STGCN is based on the human skeletal system with graph nodes corresponding to physical joints and spatial edges representing physical connectivity between these joints. Instead, we use human-object interactions to construct our spatial graph where nodes represent actors, objects, scenes, and actions whereas edges represent their spatial (e.g., next to) and/or functional (e.g., role) relationships. Various descriptors can be extracted either as the channels or nodes of the spatial graph to encode comprehensive contextual information about the actions. For example, we can use pose feature to describe actor nodes, appearance features including attributes at high semantic levels for object nodes, and frame-level RGB/flow features for scene nodes.

Second, the original STGCN only connects physical joints of the same type across consecutive time stamps, which indeed reduces to a fixed and grid-like connectivity. As a result, the temporal GCN degrades to conven-

tional convolution. To support flexible configurations and account for frequent graph deformation in complex activities (e.g., missed detections, emerging/disappearing objects, heavy occlusions, etc.), our graph allows arbitrary temporal connections. For example, an object node present at time t_0 can be connected to an object node of the same type at time t_n with $n \neq 1$ in comparison to the original STGCN with $n = 1$.

Let \mathbf{A}_s and \mathbf{A}_t denote the spatial and temporal adjacency matrices, respectively. Our proposed STGCN operation can be represented mathematically as follows:

$$\begin{aligned} \mathbf{H}^{l+1} &= g_t(\mathbf{H}_s^l, \mathbf{A}_t) = (\hat{\mathbf{D}}_t^{-1/2} \hat{\mathbf{A}}_t \hat{\mathbf{D}}_t^{-1/2} \mathbf{H}_s^l \mathbf{W}_t^l) \\ \mathbf{H}_s^l &= g_s(\mathbf{H}^l, \mathbf{A}_s) = \hat{\mathbf{D}}_s^{-1/2} \hat{\mathbf{A}}_s \hat{\mathbf{D}}_s^{-1/2} \mathbf{H}^l \mathbf{W}_s^l \end{aligned} \quad (2)$$

where \mathbf{W}_s^l and \mathbf{W}_t^l represents the spatial and temporal weight metrics of the l^{th} convolution layer, respectively. In comparison, the original STGCN reduces to

$$\mathbf{H}^{l+1} = g(\mathbf{H}^l, \mathbf{A}_s) = (\hat{\mathbf{D}}_s^{-1/2} \hat{\mathbf{A}}_s \hat{\mathbf{D}}_s^{-1/2} \mathbf{H}^l \mathbf{W}_s^l \mathbf{W}_t^l) \quad (3)$$

due to the fixed grid-like temporal connections.

Note that the original STGCN requires fixed feature length across all graph nodes, which may not hold for our applications where nodes of different types may require different feature vectors to characterize (e.g., features from Situation Recognition are of length 1024 while appearance features from Faster-RCNN[36] are of length 2048). To address the problem of varied feature length, one easy solution is to include an additional convolutional layer to convert features with varied length to fixed length (see Figure 3(a)). However, we argue that nodes of different types may require different length to embed different amounts of information. Converting features to a fixed length may decrease the amount of information they can carry. Therefore, we group nodes into clusters based on their feature length and design multiple spatial GCNs, each corresponding to one of the node cluster. These spatial GCNs will convert features to a fixed length. To allow spatial connections across these node clusters, we model these connections in the temporal adjacency matrix to avoid the use of an additional spatial GCN, since our temporal GCN already allows for arbitrary connections (see Figure 3(b)).

Notably, the S-RNN is developed for action recognition in [17] where node RNN and edge RNN are used iteratively to process graph-like input. In comparison, our model features a single graph network that can jointly process node features and edge connectivity in an interconnected manner. This, therefore, leads to improved performance and robustness.



Figure 4. Illustration of stacked hourglass STGCN with two levels.

3.3. Stacking of hourglass STGCN

Hourglass networks consist of a series of downsampling and upsampling operations with skip connections. They follow the principles of the information bottleneck approach to deep learning models [3] for improved performance. They have also been shown to work well for tasks such as human pose estimation [29], facial landmark localization [57], etc. In this work, we incorporate the hourglass architecture with STGCN so as to leverage the encoder-decoder structure for action segmentation with improved accuracy.

Our Stacked-STGCN extends and adapts the hourglass structure, commonly applied to data with regular grids (e.g., images), to data with irregular connections (e.g., graphs). This entails the development of new techniques: 1) non-symmetric encoding and decoding since feature pooling on graphs is only required in encoding stage and 2) the dimensions of the spatial and temporal adjacency matrices need to be adjusted accordingly. Our deliberate design of Stacked-STGCN stemming from 1) and 2) above tackle the difficulties in adapting the traditional hourglass to data with irregular connections and produce consistent performance improvement. To the best of our knowledge, extending/adapting the hourglass structure to spatiotemporal graphs at multiple spatial and temporal resolutions has not been attempted before.

Particularly, our GCN hourglass network contains a series of a STGCN layer followed by a strided convolution layer as the basic building block for the encoding process. Conventional deconvolution layers comprise the basic unit for the decoding process to bring the spatial and temporal dimensions to the original size. Figure 4 depicts an example with two levels.

Note that, at each layer of STGCN, the dimension of the spatial and temporal adjacency matrices, A_s and A_t , needs to be adjusted accordingly to reflect the downsampling operation. Take the illustrative example in Figure 4 for instance and assume that the adjacency matrices A_t and A_s are of size $N_t \times N_t$ and $N_s \times N_s$, respectively, at level 1 and that a stride of two is used. At level 2, both A_t and A_s are sub-sampled by two and their dimensions become $N_t/2 \times N_t/2$ and $N_s/2 \times N_s/2$, respectively. Due to the information compression enabled by the encoder-decoder structure, using hourglass networks leads to performance

gain compared to using the same number of STGCN layers one after another.

4. Experiments

4.1. CAD120

Dataset. The CAD120 dataset is one of the more simplistic datasets available for activity recognition [20]. It provides RGBD Data for 120 videos on 4 subjects as well as skeletal data. We use the 10 actions classes as our model labels including reaching, moving, pouring, eating, drinking, opening, placing, closing, scrubbing and null.

The CAD120 dataset splits each video into segments of the above mentioned actions. For each segment, it provides features for object nodes, skeleton features for actor nodes, and spatial weights for object-object and skeleton-object edges. Across segments, it also provides temporal weights for object-object and actor-actor edges. The object node feature captures information about the object’s locations in the scene and the way it changes. The OpenNI’s skeleton tracker [1] is applied to RGBD videos producing skeleton features for actor nodes. The spatial edge weights are based on the relative geometric features among the objects or between an object and the actor. The temporal edge weights capture the changes from one temporal segment to another.

Implementation. We exploited all the node features and edge weights provided by the CAD120 dataset. The skeleton feature of an actor node is of length 630 and the feature of an object node is of length 180. We pass each of these descriptors through convolution layers to convert them to a fixed length of 512. The initial learning rate is 0.00035 and the learning rate scheduler has a drop rate of 0.9 with a step size of 1. While experimentation, four fold cross-validation is carried out, where videos from 1 of the 4 people are used for testing and the videos from the rest three for training.

Results. For the CAD120 dataset, the node features and edge weights are provided by the dataset itself. The same set of features were used by S-RNN [17] and Koppula et al [20, 21] who used spatio-temporal CRF to solve the problem. The S-RNN trains two separate RNN models, one for nodes (i.e., nodeRNN) and the other for edges (i.e., edgeRNN). The edgeRNN is a single layer LSTM of size 128 and the nodeRNN uses an LSTM of size 256. The actor nodeRNN outputs an action label at each time step. In Table 1, we show some of the previous results, including the best reported one from S-RNN, as well as the result of our STGCN. The F1 score is used as the evaluation metric. We cannot compare to [34] as they do not follow the 4 fold cross-validation, a convention most of the previous works used.

Our STGCN outperforms the S-RNN by about 5.3% in F1 score. Instead of using two independent RNNs to model interactions among edges and nodes, our STGCN collectively performs joint inference over these inherently inter-

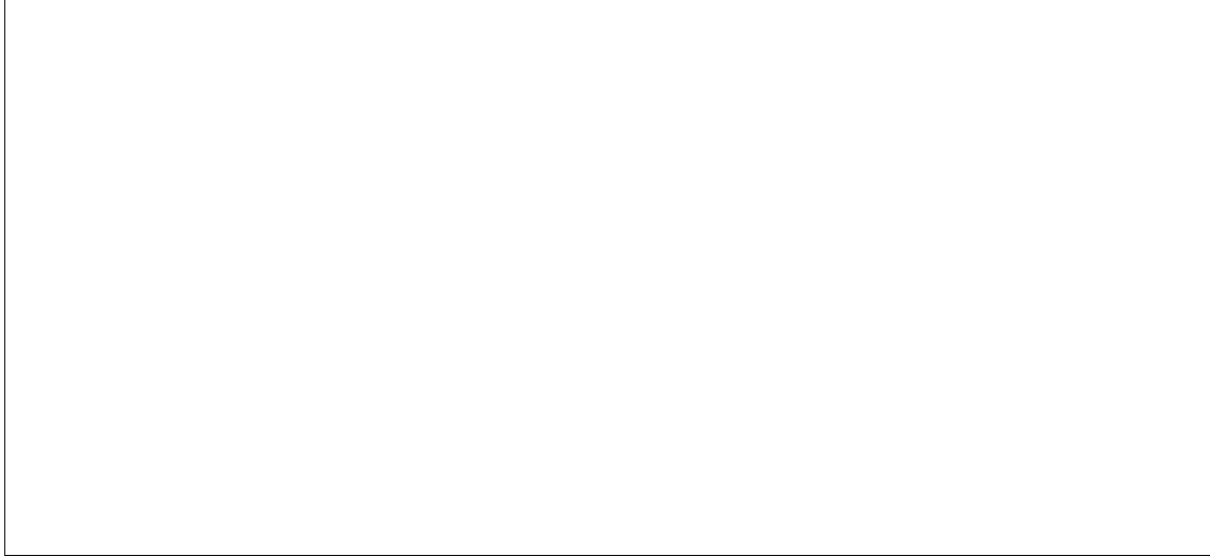


Figure 5. Action segmentation results of our Stacked-STGCN on CAD120. Green/red: correct/erroneous detection.

Method	F1-score (%)
Koppula et al. [20, 21]	80.4
S-RNN w/o edge-RNN [17]	82.2
S-RNN [17]	83.2
S-RNN(multitask) [17]	82.4
Ours (STGCN)	88.5

Table 1. Performance comparison based on the F1 score using the CAD120 dataset. Our STGCN improves the F1 score over the best reported result (i.e., S-RNN) by approximately 5.3%.

connected features. This, therefore, leads to the observed performance improvement.

In Figure 5, we can see a couple of errors in the second and third examples. For example, the third prediction is ‘opening’ instead of ‘moving’ in the second example. The previous action is ‘reaching’ which is generally what precedes ‘opening’ when the actor is standing in front of a microwave and looking at it. So probably that is the reason for the observed erroneous detection. Also the ninth frame is classified ‘reaching’ instead of ‘moving’. If we look at the ninth frame and the eleventh frame, everything appears the same except for the blue cloth in the actor’s hand. Our STGCN failed to capture such subtle changes and therefore predicted the wrong action label.

4.2. Charades

Dataset. The Charades is a recent real-world activity recognition/segmentation dataset including 9848 videos with 157 action classes, 38 object classes, and 33 verb classes [45, 46]. It contains both RGB and flow streams at a frame rate of 24fps. It poses a multi-label, multi-class problem in the sense that at each time step there can be more

Description
Scene Features N1. FC7 layer output of VGG network trained on RGB frames
Motion Features N2. FC7 layer output of VGG network trained on flow frames
Segment Features N3. I3D pre-final layer output trained on RGB frames N4. I3D pre-final layer output trained on flow frames
Actor Features N5. GNN-based Situation Recognition trained on the ImSitu dataset
Object Features N6. Top 5 object detection features from Faster-RCNN

Table 2. Features for the Charades dataset.

than one action label. The dataset provides ground-truth object and verb labels as well as FC7 features for every 4th frame obtained from a two-stream network trained on Charades. The entire dataset is split into 7985 training videos and 1863 testing videos.

Implementation. For the Charades dataset, we explored two types of features, one based on VGG [48] and the other based on I3D [5], for the scene nodes in our spatio-temporal graph. Further, we used the GNN-based situation recognition technique [24] trained on the ImSitu dataset [58] to generate the verb feature for the actor nodes. The top five object features of the Faster-RCNN network [36] trained on MSCOCO are used as descriptors of the object nodes. We chose top five object nodes through empirical study of the

videos where we observed that five is roughly the maximum number of objects. Note that our Stacked-STGCN operates directly on object candidate descriptors so that we can bypass explicit object detection and avoid the challenges in dealing with varying number of action associated objects. These descriptors can be unrelated or redundant (since no non-maximum suppression is applied) to the current action of interest. In total, the spatial dimension of our STG is eight. The VGG features are of length 4096, the verb features 1024, and the object features 2048. Each of these channels are individually processed using convolution layers to convert them to a fixed length (e.g., we used 512). Table 2 summarizes these features.

In this experiment, spatial nodes are fully connected and temporal edges allow connections across three time steps, i.e., at the t^{th} step there are edges from t , to $t + 1$ and $t + 2$ and $t + 3$. The connections are binary, meaning if there is a connection, it has weight 1. The adjacency matrix is normalized using the normalized graph laplacian function since it does better than the normalization technique used by [19]. We used a stack of three hourglass STGCN blocks. The output of the final Stacked-STGCN block is spatially pooled and passes through a fully connected layer to generate the probability scores of all possible classes. Since the Charades is a multi-label, multi-class dataset, the binary cross-entropy loss was used. We used an initial learning rate of 0.001 and a learning rate scheduler with a step size of 10 and a drop rate of 0.999.

To further improve action segmentation performance on Charades, we have used a trained I3D model on Charades to generate descriptors for the scene nodes replacing the VGG features. These feature descriptors are of length 1024. Since I3D already represents short-term temporal dependencies, one block of hourglass STGCN is sufficient for capturing long-term temporal dependencies. We also did not use object nodes with I3D since they did not result in improvements in performance. This means that the RGB and Flow I3D features are passed through separate temporal graph convolution networks and undergoes late fusion. The initial learning rate was 0.0005 and the learning rate scheduler was fixed at a drop rate of 0.995 at a step size of 10.

During training, we chose our maximum temporal dimension to be 50. If the length of a video segment is less than 50, we zero-pad the rest of the positions. But these positions are not used for loss or score computation. If the length of a video segment is greater than 50, we randomly select a starting point and use the 50 consecutive frames as the input to our graph.

At test time, we used a sliding window of length 50. Based on overlapping ratios, we applied a weighted average over these windowed scores to produce the final score. We used an overlap of 40 time steps. Following instructions in the Charades dataset, we selected 25 equally spaced points

from the available time steps in the video, to generate the final score vectors.

Ablation Studies. As to the Charades dataset, the mean average precision (mAP) is used as the evaluation metric. For fair comparison, we have used the scripts provided by the Charades dataset to generate mAP scores. We examined the performance of Stacked-STGCN using two types of descriptors for the scene nodes, namely frame-based VGG features and segment-based I3D features (see Table 2). We summarize our ablation studies in Table 3

(A1)	All Features; Baseline	8.13
(A2)	All Features; STGCN	10.26
(A3)	VGG-RGB; STGCN; 1 time step	6.77
(A4)	VGG-RGB; STGCN	7.06
(A5)	All Features; Stacked-STGCN; 1 time step	11.29
(A6)	VGG-RGB; Stacked-STGCN;	8.66
(A6)	VGG-RGB+VGG-Flow; Stacked-STGCN	10.94
(A7)	All Features; Stacked-STGCN	11.73

Table 3. Comparison of our Stacked-STGCN (A7) with baseline (A1), STGCN without hourglass (A2), different temporal connections (A3-A5), and different input features (A6). Input features include VGG-RGB for scene, VGG-Flow for motion, Situation Recognition for action, and Faster RCNN for object.

We first examine the performance improvement introduced by structured inference of contextual information represented in spatio-temporal graphs. We implemented a baseline method (A1) in Table 3 which employs a Fully Connected layer for joint inference of multiple types of features. We compare our Stacked-STGCN (A7) with this baseline (A1) and demonstrate an improvement of 3.6% .

We also compare our Stacked-STGCN (A7) with an implementation without the hourglass structure (A2) and demonstrate an improvement of 1.47% in Table 3. For fair comparison of this experiment, we design a network (A2) with the same number of convolutional layers as the encoder of our Stacked-STGCN. To maintain the same temporal resolution, these convolution layers have a stride of one, compared to a stride of two in the Stacked-STGCN.

We further implement a network that closely resembles the original STGCN: 1) nodes are represented by the same type of features (i.e., VGG-RGB); 2) pure graph convolutional operations (i.e., without hourglass); and 3) temporal connections across one time step. Comparing to this vanilla implementation (A3), our Stacked-STGCN (A7) produces an improvement of 4.96% in Table 3.

Next, we conduct a study on the performance of Stacked-STGCN with different input features. With one, two and four types of features, the performances are 8.66, 10.94, and 11.73, respectively, in Table 3 (A6, A7). This steady improvement is due to more context gained from enriched input features.

Method	VGG mAP	I3D mAP
Baseline [30]	6.56	17.22
LSTM [30]	7.85	18.12
Super-Events [30]	8.53	19.41
Stacked-STGCN (VGG only)	10.94	19.09
Stacked-STGCN (all features)	11.73	
Stacked-STGCN (I3D)		

Table 4. Performance comparison based on mAP between our Stacked-STGCN and the best reported results published in [30] using the Charades dataset. Our Stacked-STGCN yields an approximate 2.41% and 3.20% improvement in mAP using VGG features only and all four types of features, respectively.

Method	mAP
Random [44]	2.42
RGB [44]	7.89
Predictive-corrective [7]	8.90
Two-Stream [44]	8.94
Two-Stream + LSTM [44]	9.60
Sigurdsson <i>et al.</i> standard [44]	9.69
Sigurdsson <i>et al.</i> post-processing [44]	12.80
R-C3D [55]	12.70
I3D [5]	17.22
I3D +LSTM [30]	18.10
I3D+Temporal Pyramid [30]	18.20
I3D + Super-events [30]	19.41
I3D +Stacked-STGCN (ours)	19.09

Table 5. Performance comparison based on mAP with previous works using the Charades dataset.

Finally, we study the performance of our Stacked-STGCN with different temporal connections. Comparing (A7) vs. (A5) in Table 3, temporal connections with three time steps demonstrate an improvement of 0.44%. With a simpler network (i.e., without hourglass), we observe an improvement of 0.29%, (A4) vs. (A3). The optimal number of time steps can vary depending on networks and applications. The empirical optimal number for our Stacked-STGCN on Charades is three.

Comparison with SOTA. In Table 4, the performance of Stacked-STGCN is compared with a baseline, which uses two-stream VGG or I3D features directly for per frame action label prediction, an LSTM-based method, and the Super-Events approach proposed in [30]. Our Stacked-STGCN yields an approximate 2.41% and 3.20% improvement in mAP using VGG features only and all four types of features, respectively. Using I3D features, our Stacked-STGCN ranks the second.

In Table 5, we compare the performance of Stacked-STGCN against some selected works on Charades. We can see that our Stacked-STGCN outperforms all the meth-

ods except for the I3D+super-events [30], which employs an attention mechanism to learn proper temporal span per class. We believe that incorporating such attention mechanism could further improve the performance of our Stacked-STGCN. Furthermore, our method provides a principled way of structured inference over heterogeneous features, which most of the list methods are incapable of.

Another set of results on Charades is from the workshop held in conjunction with CVPR 2017. The results in that competition appear better. However, as mentioned in [30], that competition used a test set that is different from the validation set we used for performance evaluation. Besides those techniques could have used both the training and validation sets for training. Reference [30] also shows that the same algorithm (i.e., I3D) that produced 20.72 in the competition produced only 17.22 on the validation set.

5. Conclusion

The proposed Stacked-STGCN introduces a stacked hourglass architecture to STGCN for improved generalization performance and localization accuracy. Its building block STGCN is generic enough to take in a variety of nodes/edges and to support flexible graph configuration. In this paper, we applied our Stacked-STGCN to action segmentation and demonstrated improved performances on the CAD120 and Charades datasets. We also note that adding spatial edge connections between nodes from same model lead to performance improvement on Charades rather than across different feature nodes. This is mainly due to the oversimplified edge model (i.e., with fixed weights). Instead of using a binary function to decide on the correlation between these nodes, more sophisticated weights could be explored. We leave this as future work. Furthermore, graph representation based on actor, action, object and scene provides inherent explanations of corresponding detection of action categories. However, such explanation requires visualizing the traces of most activated nodes/edges, which current GCN implementations can not support. We will also leave this as future work. Finally, we anticipate that due to its generic design Stacked-STGCN can be applied to a wider range of applications that require inference over a sequence of graphs with heterogeneous data types and varied temporal extent.

Acknowledgments

This work is supported by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Interior/Interior Business Center (DOI/IBC) contract number D17PC00343. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing endorsements, either expressed or implied, of IARPA, DOI/IBC, or the U.S. Government.

References

- [1] <http://openpi.org>.
- [2] J. Atwood and D. Towsley. Diffusion-convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 1993–2001, 2016.
- [3] L. Blier and Y. Ollivier. The description length of deep learning models. In *NIPS*, 2018.
- [4] J. Bruna and X. Li. Community detection with graph neural networks. *arXiv preprint arXiv:1705.08415*, 2017.
- [5] J. Carreira and A. Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, pages 4724–4733. IEEE, 2017.
- [6] G.-J. Chen, I.-C. Chang, and H.-Y. Yeh. Action segmentation based on bag-of-visual-words models. In *2017 10th International Conference on Ubi-media Computing and Workshops (Ubi-Media)*, pages 1–5. IEEE, 2017.
- [7] A. Dave, O. Russakovsky, and D. Ramanan. Predictive corrective networks for action detection. In *Proceedings of the Computer Vision and Pattern Recognition*, 2017.
- [8] M. Defferrard, X. Bresson, and P. Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in Neural Information Processing Systems*, pages 3844–3852, 2016.
- [9] A. Diba, M. Fayyaz, V. Sharma, A. H. Karami, M. M. Arzani, R. Yousefzadeh, and L. Van Gool. Temporal 3d convnets: New architecture and transfer learning for video classification. *arXiv preprint arXiv:1711.08200*, 2017.
- [10] L. Ding and C. Xu. Tricomet: A hybrid temporal convolutional and recurrent network for video action segmentation. *arXiv preprint arXiv:1705.07818*, 2017.
- [11] L. Ding and C. Xu. Video action segmentation with hybrid temporal networks. 2018.
- [12] L. Ding and C. Xu. Weakly-supervised action segmentation with iterative soft boundary assignment. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6508–6516, 2018.
- [13] D. K. Duvenaud, D. Maclaurin, J. Iparraguirre, R. Bombarell, T. Hirzel, A. Aspuru-Guzik, and R. P. Adams. Convolutional networks on graphs for learning molecular fingerprints. In *Advances in neural information processing systems*, pages 2224–2232, 2015.
- [14] Y. A. Farha and J. Gall. Ms-tcn: Multi-stage temporal convolutional network for action segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3575–3584, 2019.
- [15] M. Guo, E. Chou, D.-A. Huang, S. Song, S. Yeung, and L. Fei-Fei. Neural graph matching networks for fewshot 3d action recognition. In *European Conference on Computer Vision*, pages 673–689. Springer, 2018.
- [16] D. K. Hammond, P. Vandergheynst, and R. Gribonval. Wavelets on graphs via spectral graph theory. *Applied and Computational Harmonic Analysis*, 30(2):129–150, 2011.
- [17] A. Jain, A. R. Zamir, S. Savarese, and A. Saxena. Structural-rnn: Deep learning on spatio-temporal graphs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5308–5317, 2016.
- [18] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017.
- [19] T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.
- [20] H. S. Koppula, R. Gupta, and A. Saxena. Learning human activities and object affordances from rgb-d videos. *The International Journal of Robotics Research*, 32(8):951–970, 2013.
- [21] H. S. Koppula and A. Saxena. Anticipating human activities using object affordances for reactive robotic response. *IEEE transactions on pattern analysis and machine intelligence*, 38(1):14–29, 2016.
- [22] C. Lea, M. D. Flynn, R. Vidal, A. Reiter, and G. D. Hager. Temporal convolutional networks for action segmentation and detection. In *IEEE International Conference on Computer Vision (ICCV)*, 2017.
- [23] P. Lei and S. Todorovic. Temporal deformable residual networks for action segmentation in videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6742–6751, 2018.
- [24] R. Li, M. Tapaswi, R. Liao, J. Jia, R. Urtasun, and S. Fidler. Situation recognition with graph neural networks. *arXiv preprint arXiv:1708.04320*, 2017.
- [25] Y. Li, W. Ouyang, B. Zhou, J. Shi, C. Zhang, and X. Wang. Factorizable net: an efficient subgraph-based framework for scene graph generation. In *European Conference on Computer Vision*, pages 346–363. Springer, 2018.
- [26] Y. Li, D. Tarlow, M. Brockschmidt, and R. Zemel. Gated graph sequence neural networks. *arXiv preprint arXiv:1511.05493*, 2015.
- [27] Y.-M. Liang, S.-W. Shih, and A. C.-C. Shih. Human action segmentation and classification based on the isomap algorithm. *Multimedia tools and applications*, 62(3):561–580, 2013.
- [28] E. Mavroudi, D. Bhaskara, S. Sefati, H. Ali, and R. Vidal. End-to-end fine-grained action segmentation and recognition using conditional random field models and discriminative sparse coding. *arXiv preprint arXiv:1801.09571*, 2018.
- [29] A. Newell, K. Yang, and J. Deng. Stacked hourglass networks for human pose estimation. In *European Conference on Computer Vision*, pages 483–499. Springer, 2016.
- [30] A. Piergiovanni and M. S. Ryoo. Learning latent super-events to detect multiple activities in videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 4, 2018.
- [31] A. Piergiovanni and M. S. Ryoo. Temporal gaussian mixture layer for videos. *arXiv preprint arXiv:1803.06316*, 2018.
- [32] H. Pirsiavash and D. Ramanan. Parsing videos of actions with segmental grammars. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 612–619, 2014.
- [33] S. Qi, W. Wang, b. Jia, J. Shen, and S.-C. Zhu. Learning human-object interactions by graph parsing neural networks. In *ECCV*, 2018.

- [34] S. Qi, W. Wang, B. Jia, J. Shen, and S.-C. Zhu. Learning human-object interactions by graph parsing neural networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 401–417, 2018.
- [35] Z. Qiu, T. Yao, and T. Mei. Learning spatio-temporal representation with pseudo-3d residual networks. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 5534–5542. IEEE, 2017.
- [36] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.
- [37] A. Richard and J. Gall. Temporal action detection using a statistical language model. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.
- [38] A. Richard, H. Kuehne, and J. Gall. Action sets: Weakly supervised action segmentation without ordering constraints. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5987–5996, 2018.
- [39] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini. The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1):61–80, 2009.
- [40] F. Scarselli, S. L. Yong, M. Gori, M. Hagenbuchner, A. C. Tsoi, and M. Maggini. Graph neural networks for ranking web pages. In *Proceedings of the 2005 IEEE/WIC/ACM International Conference on Web Intelligence*, pages 666–672. IEEE Computer Society, 2005.
- [41] M. Schlichtkrull, T. N. Kipf, P. Bloem, R. van den Berg, I. Titov, and M. Welling. Modeling relational data with graph convolutional networks. In *European Semantic Web Conference*, pages 593–607. Springer, 2018.
- [42] A. Shahroudy, J. Liu, T.-T. Ng, and G. Wang. Ntu rgb+ d: A large scale dataset for 3d human activity analysis. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1010–1019, 2016.
- [43] Q. Shi, L. Wang, L. Cheng, and A. Smola. Discriminative human action segmentation and recognition using semi-markov model. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2008.
- [44] G. A. Sigurdsson, S. K. Divvala, A. Farhadi, and A. Gupta. Asynchronous temporal fields for action recognition. In *CVPR*, volume 5, page 7, 2017.
- [45] G. A. Sigurdsson, A. Gupta, C. Schmid, A. Farhadi, and K. Alahari. Actor and observer: Joint modeling of first and third-person videos. In *CVPR*, 2018.
- [46] G. A. Sigurdsson, G. Varol, X. Wang, A. Farhadi, I. Laptev, and A. Gupta. Hollywood in homes: Crowdsourcing data collection for activity understanding. In *European Conference on Computer Vision*, 2016.
- [47] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in neural information processing systems*, pages 568–576, 2014.
- [48] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.
- [49] Y. Souri, M. Fayyaz, and J. Gall. Weakly supervised action segmentation using mutual consistency. *arXiv preprint arXiv:1904.03116*, 2019.
- [50] G. Te, W. Hu, Z. Guo, and A. Zheng. Rgcnn: Regularized graph cnn for point cloud segmentation. *arXiv preprint arXiv:1806.02952*, 2018.
- [51] D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun, and M. Paluri. A closer look at spatiotemporal convolutions for action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6450–6459, 2018.
- [52] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *European Conference on Computer Vision*, pages 20–36. Springer, 2016.
- [53] X. Wang and A. Gupta. Videos as space-time region graphs. *arXiv preprint arXiv:1806.01810*, 2018.
- [54] X. Xiang, Y. Tian, A. Reiter, G. D. Hager, and T. D. Tran. S3d: Stacking segmental p3d for action quality assessment. In *2018 25th IEEE International Conference on Image Processing (ICIP)*, pages 928–932. IEEE, 2018.
- [55] H. Xu, A. Das, and K. Saenko. R-c3d: region convolutional 3d network for temporal activity detection. In *IEEE Int. Conf. on Computer Vision (ICCV)*, pages 5794–5803, 2017.
- [56] S. Yan, Y. Xiong, and D. Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *AAAI*, 2018.
- [57] J. Yang, Q. Liu, and K. Zhang. Stacked hourglass network for robust facial landmark localisation. In *Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2017 IEEE Conference on, pages 2025–2033. IEEE, 2017.
- [58] M. Yatskar, L. Zettlemoyer, and A. Farhadi. Situation recognition: Visual semantic role labeling for image understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5534–5542, 2016.
- [59] J. Zhang, Y. Zheng, and D. Qi. Deep spatio-temporal residual networks for citywide crowd flows prediction. In *AAAI*, pages 1655–1661, 2017.
- [60] X. Zhang, C. Xu, and D. Tao. Graph edge convolutional neural networks for skeleton based action recognition. *arXiv preprint arXiv:1805.06184*, 2018.