Videos as Space-Time Region Graphs

Xiaolong Wang, Abhinav Gupta

Robotics Institute, Carnegie Mellon University

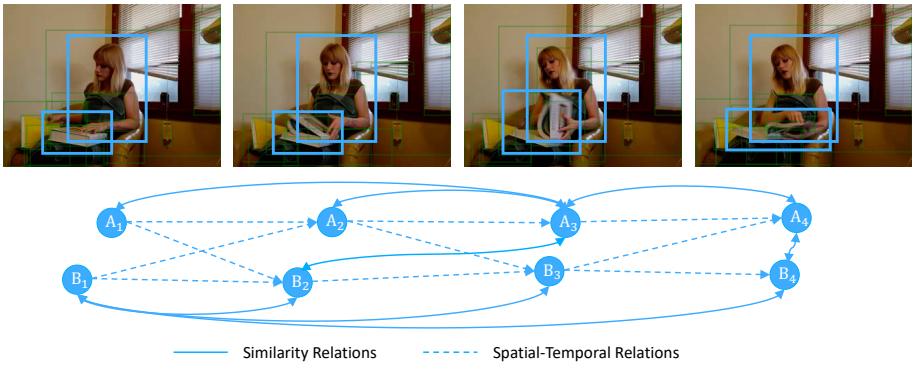

Figure 1. How do you recognize simple actions such as opening book? We argue action understanding requires appearance modeling but also capturing temporal dynamics (how shape of book changes) and functional relationships. We propose to represent videos as space-time region graphs followed by graph convolutions for inference.

Abstract How do humans recognize the action "opening a book"? We argue that there are two important cues: modeling temporal shape dynamics and modeling functional relationships between humans and objects. In this paper, we propose to represent videos as space-time region graphs which capture these two important cues. Our graph nodes are defined by the object region proposals from different frames in a long range video. These nodes are connected by two types of relations: (i) similarity relations capturing the long range dependencies between correlated objects and (ii) spatial-temporal relations capturing the interactions between nearby objects. We perform reasoning on this graph representation via Graph Convolutional Networks. We achieve state-of-the-art results on the Charades and Something-Something datasets. Especially for Charades with complex environments, we obtain a huge 4.4% gain when our model is applied in complex environments.

1 Introduction

Consider a simple action such as "opening a book" as shown in Fig. 1. When we humans see the sequence of images, we can easily recognize the action category;

yet our current vision systems (with hundreds of layers of 3D convolutions) struggle on this simple task. Why is that? What is missing in current video recognition frameworks?

Let's first take a closer look at the sequence shown in Fig. 1. How do humans recognize the action in the video corresponds to "opening a book"? We argue that there are two key ingredients to solving this problem: First, the shape of the book and how it changes over time (i.e., the object state changes from closed to open) is a crucial cue. Exploiting this cue requires temporally linking book regions across time and modeling actions as transformations. But just modeling temporal dynamics of objects is not sufficient. The state of objects change after interaction with human or other objects. Thus we also need to model human-object and object-object interactions as well for action recognition.

However, our current deep learning approaches fail to capture these two key ingredients. For example, the state-of-the-art approaches based on two-stream ConvNets [1,2] are still learning to classify actions based on individual video frame or local motion vectors. Local motion clearly fails to model the dynamics of shape changes. To tackle this limitation, recent work has also focused on modeling long term temporal information with Recurrent Neural Networks [3,4,5,6] and 3D Convolutions [7,8,9,10]. However, all these frameworks focus on the features extracted from the whole scenes and fail to capture long-range temporal dependencies (transformations) or region-based relationships. In fact, most of the actions are classified based on the background information instead of capturing the key objects (e.g., the book in "opening a book") as observed in [11].

On the other hand, there have been several efforts to specifically model the human-object or object-object interactions [12,13]. This direction have been recently revisited with ConvNets in an effort to improve object detection [14,15,16], visual relationship detection [17] and action recognition [18], etc. However, the relationship reasoning is still performed in static images failing to capture temporal dynamics of these interactions. Thus, it is very hard for these approaches to capture the changes of object states over time as well as the causes and effects of these changes.

In this paper, we propose to perform long-range temporal modeling of human-object and object-object relationships via a graph-based reasoning framework. Unlike existing approaches which focus on local motion vectors, our model takes in a long range video sequence (e.g., more than 100 frames or 5 seconds). We represent the input video as a **space-time region graph** where each node in the graph represent region of interest in the video. Region nodes are connected by two types of edges: appearance-similarity and spatio-temporal proximity. Specifically, **(i) Similarity Relations**: regions which have similar appearance or semantically related are connected together. With similarity relations, we can model how the states of the same object change and the long range dependencies between any two objects in any frames. **(ii) Spatial-Temporal Relations**: objects which overlap in space and close in time are connected together via these

edges. With spatial-temporal relations, we can capture the interactions between nearby objects as well as the temporal ordering of object state changes.

Given the graph representation, we perform reasoning on the graph and infer the action by applying the Graph Convolution Networks (GCNs) [19]. We conduct our experiments in the challenging Charades [20] and 20BN-Something-Something [21] datasets. Both datasets are extremely challenging as the actions cannot be easily inferred by the background of the scene and the 2D appearance of the objects or humans. Our model shows significant improvements over state-of-the-art results of action recognition. Especially in the Charades dataset, we obtain 4.4% boost.

Our contributions include: (a) A novel graph representation with variant relationships between objects in a long range video; (b) A Graph Convolutional Network for reasoning with multiple relation edges; (c) state-of-the-art performance with a significant gain in action recognition in complex environments.

2 Related Work

Video Understanding Models. Spatio-temporal reasoning is one of the core research areas in the field of video understanding and action recognition. However, most of the early work has focused on using spatio-temporal appearance features. For example, a large effort has been spent on manually designing the video features [22,23,24,25,26,27,28,29,30,31]. Some of the hand-designed features such as the Improved Dense Trajectory (IDT) [23] are still widely applied and show very competitive results in different video related tasks. However, instead of designing hand-crafted features, recent researches have focused towards learning deep representations from the video data [32,1,33,34,35,36,2,37]. One of the most popular model is the two-Stream ConvNets [1] where temporal information is model by a network with 10 optical flow frames as inputs (<1 second). To better model longer-term information, a lot of work has been focused on using Recurrent Neural Networks (RNNs) [3,4,38,39,40,5,41,42,43] and 3D ConvNets [44,45,8,9,46,47,48]. However, these frameworks focus on extracting features from the whole scenes and can hardly model the relationships between different object instances in space and time.

Visual Relationships. Reasoning about the pairwise relationships has been proven to be very helpful in a variety of computer vision tasks [12,13,49,50,51]. For example, object detection in cluttered scenes can be significantly improved by modeling the human-object interactions [13]. Recently, the visual relationships have been widely applied together with deep networks in the area of visual question answering [52], object recognition [14,15,16] and intuitive physics [53,54]. In action recognition, a lot of effort has been made on modeling pairwise human-object and object-object relationships [55,56,57]. However, the interaction reasoning framework in these efforts focuses on static images and the temporal information is usually modeled by a RNN on image level features. Thus, these approaches still cannot capture how a certain object state changes over time.

An attempt at modeling pairwise relations in space and time has been recently made in the Non-local Neural Networks [58]. However, the Non-local operator is applied in every pixel in the feature space (from low layers to higher layers), while our reasoning is based on a graph with object level features. Moreover, the non-local operator does not process any temporal ordering information, while this is explicit modeled in our spatial-temporal relations.

Graphical Models. The long range relationships in images and videos are usually captured by graphical models. One popular direction is using the Conditional Random Fields (CRF) [59,60]. In the context of deep learning, especially for semantic segmentation, the CRF model is often applied on the outputs of the ConvNets by performing mean-field inference [61,62,63,64,65,66]. Instead of using mean-field inference, variant simpler feedforward graph based neural network have been proposed recently [67,68,19,69,70,71]. In this paper, we apply the Graph Convolutional Networks (GCNs) [19] which was originally proposed for applications in Natural Language Processing. Our GCN is built by stacking multiple layers of graph convolutions with similarity relations and spatial-temporal relations. The outputs of the GCNs are updated features for each object node, which can be used to perform classification.

Our work is also related to video recognition with object cues [72,73,74] and object graph models [75,76,77,78]. For example, Structural-RNN [77] is proposed to model the spatial-temporal relations between objects (adjacent in time) for video recognition tasks. Different from these works, our space-time graph representation encodes not only local relations but also long range dependencies between any pairs of objects across space and time. By using graph convolutions with long range relations, it enables efficient message passing between starting states and ending states of the objects. This global graph reasoning framework provides significant boost over the state-of-the-art.

3 Overview

Our goal is to represent the video as a graph of objects and perform reasoning on the graph for action recognition. The overview of our model is visualized in Figure 2. Our model takes inputs as a long clip of video frames (more than 5 seconds) and forward them to a 3D Convolutional Neural Network [8,58]. The output of this 3D ConvNet is a feature map with the dimensions $\mathrm{T} \times \mathrm{H} \times \mathrm{W} \times d$, where T represents the temporal dimension, $\mathrm{H} \times \mathrm{W}$ represents the spatial dimensions and d represents the channel number.

Besides extracting the video features, we also apply a Region Proposal Network (RPN) [79] to extract the object bounding boxes (We have not visualized the RPN in Figure 2 for simplicity). Given the bounding boxes for each of the T feature frames, we apply RoIAlign [80,81] to extract the features for each bounding box. Note that the RoIAlign is applied on each feature frame independently. The feature vector for each object has d dimensions (first aligned to $7 \times 7 \times d$ and then maxpooled to $1 \times 1 \times d$). We denote the object number as N, thus the feature dimension is $\mathrm{N} \times d$ after RoIAlign.

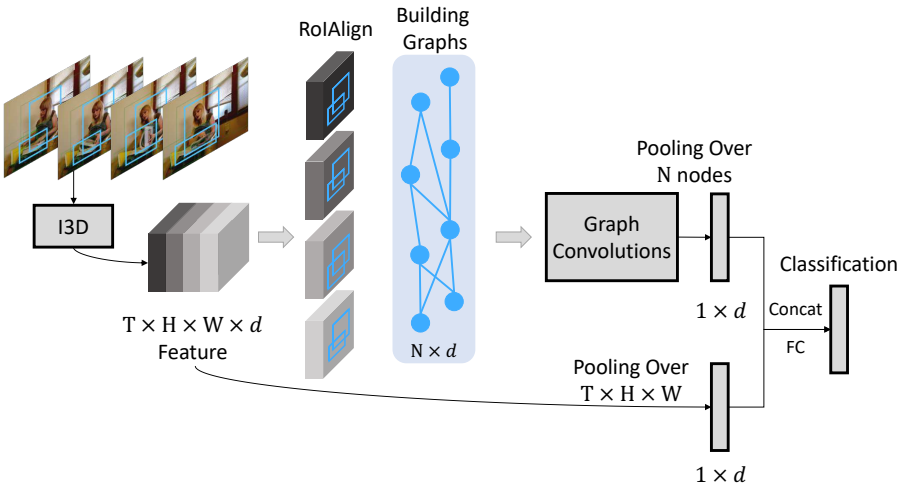

Figure 2. Model Overview. Our model uses 3D convolutions to extract visual features followed by RoIAlign extracting d-dimension feature for each object proposal. These features are provided as inputs to the Graph Convolutional Network which performs information propagation based on spatiotemporal edges. Finally, a d-dimension feature is extracted and appended to another d-dimension video feature to perform classification.

We now construct a graph which contains N nodes corresponding to N object proposals aggregated over T frames. There are mainly two types of relations in the graph: similarity relations and spatial-temporal relations. For simplicity, we decompose this big graph into two sub-graphs with the same nodes but two different relations: the similarity graph and the spatial-temporal graph.

With the graph representations, we apply the Graph Convolutional Networks (GCNs) to perform reasoning. The output for the GCNs are in the same dimension as the input features which is $\mathrm{N} \times d$. We perform average pooling over all the object nodes to obtain a d-dimension feature. Besides the GCN features, we also perform average pooling on the whole video representation ($\mathrm{T} \times \mathrm{H} \times \mathrm{W} \times d$) to obtain the same d-dimension feature as a global feature. These two features are then concatenated together for video level classification.

4 Graph Representations in Videos

In this section, we first introduce the feature extraction process with 3D ConvNets and then describe the the similarity graph as well as the spatial-temporal graph.

4.1 Video Representation

Video Backbone Model. Given a long clip of video (around 5 seconds), we sample 32 video frames from it with the same temporal duration between every

layer		output size
conv_1	5×7×7, 64, stride 1, 2, 2	32×112×112
pool_1	1×3×3 max, stride 1, 2, 2	32×56×56
res_2	$\begin{bmatrix} 3\times1\times1, 64 \\ 1\times3\times3, 64 \\ 1\times1\times1, 256 \end{bmatrix} \times 3$	32×56×56
pool_2	3×1×1 max, stride 2, 1, 1	16×56×56
res_3	$\begin{bmatrix} 3\times1\times1, 128 \\ 1\times3\times3, 128 \\ 1\times1\times1, 512 \end{bmatrix} \times 4$	16×28×28
res_4	$\begin{bmatrix} 3\times1\times1, 256 \\ 1\times3\times3, 256 \\ 1\times1\times1, 1024 \end{bmatrix} \times 6$	16×14×14
res_5	$\begin{bmatrix} 3\times1\times1, 512 \\ 1\times3\times3, 512 \\ 1\times1\times1, 2048 \end{bmatrix} \times 3$	16×14×14
global average pool, fc		1×1×1

Table 1. Our baseline ResNet-50 I3D model. We use T×H×W to represent the dimensions of filter kernels and 3D output feature maps. For filter kernels, we also have number of channels following T×H×W. The input is in 32×224×224 dimensions and the residual blocks are shown in brackets.

two frames. We extract the features on these frames via a 3D ConvNet. Table 1 shows our backbone model based on the ResNet-50 architecture (motivated by [58]). The model takes input as 32 video frames with 224×224 dimensions and the output of the last convolutional layer is a $16 \times 14 \times 14$ feature map (i.e., 16 frames in the temporal dimension and 14×14 in the spatial dimension). The baseline in this paper adopts the same architecture, where the classification is simply performed by using a global average pooling on the final convolutional features following by a fully connected layer.

This backbone model is called Inflated 3D ConvNet (I3D) [48,8,58] as one can turn a 2D ConvNet into a 3D ConvNet by inflating the kernels during initialization. That is, a 3D kernel with $t \times k \times k$ dimensions can be inflated from a 2D $k \times k$ kernel by copying the weights t times and rescaling by $1/t$. Please refer to [48,8,58] for more initialization details.

Region Proposal Network. We apply the Region Proposal Network (RPN) in [79,82] to generate the object bounding boxes of interest on each video frame. More specifically, we use the RPN with ResNet-50 backbone and FPN [83]. The RPN is pre-trained with the MSCOCO object detection dataset [84] and there is no weight sharing between the RPN and our I3D video backbone model. Note that the bounding boxes extracted by the RPN are class-agnostic.

To extract object features on top of the last convolutional layer, we project the bounding boxes from the 16 input RGB frames (sampled from the 32 input frames for I3D, with the sampling rates of 1 frame every 2 frames) to the 16 output feature frames. Taking the video features and projected bounding boxes, we apply

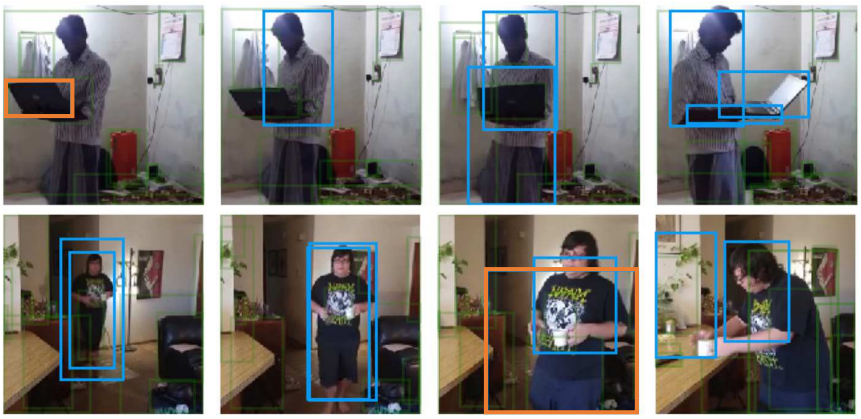

Figure 3. Similarity Graph $\mathbf{G}^{sim}$. Above figure shows our similarity graph not only captures similarity in visual space but also correlations (similarity in functional space). The query box is shown in orange, the nearest neighbors are shown in blue. The transparent green boxes are the other unselected object proposals.

RoIAlign [81] to extract the feature for each object proposal. In RoIAlign, each output frame is processed independently. The RoIAlign generates a $7 \times 7 \times d$ output features for each object which is then max-pooled to $1 \times 1 \times d$ dimensions.

4.2 Similarity Graph

We measure the similarity between objects in the feature space to construct the similarity graph. In this graph, we connect pairs of semantically related objects together. More specifically, we will have a high confidence edge between two instances which are: (i) the same object in different states in different video frames or (ii) highly correlated for recognizing the actions. Note that the similarity edges are computed between any pairs of objects.

Formally, assuming we have the features for all the object proposals in the video as $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N\}$, where N represents the number of object proposals and each object proposal feature $\mathbf{x}_i$ is a d dimensional vector. The pairwise similarity or the affinity between every two proposals can be represented as,

$$F(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^T \phi'(\mathbf{x}_j), \tag{1}$$

where ϕ and ϕ' represents two different transformations of the original features. More specifically, we have $\phi(\mathbf{x}) = \mathbf{w}\mathbf{x}$ and $\phi'(\mathbf{x}) = \mathbf{w}'\mathbf{x}$. The parameters $\mathbf{w}$ and $\mathbf{w}'$ are both $d \times d$ dimensions weights which can be learned via back propagation. By adding the transformation weights $\mathbf{w}$ and $\mathbf{w}'$, it allows us to not only learn the correlations between different states of the same object instance across frame, but also the relations between different objects. We visualize the top nearest neighbors for the object proposals in Figure 3. In the first example, we can see

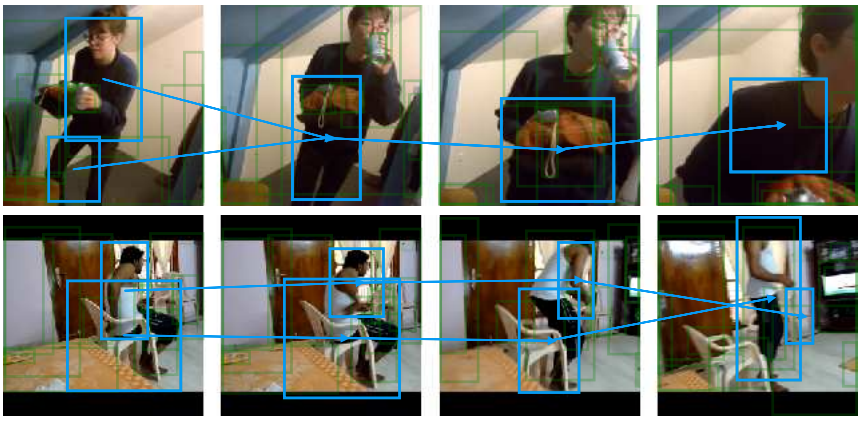

Figure 4. Spatial-Temporal Graph $\mathbf{G}^{front}$. Highly overlapping object proposals across neighboring frames are linked by directed edge. We plot some example trajectories with blue boxes and the direction shows the arrow of time.

the nearest neighbors of the laptop not only include the other laptop instances in other frames, but also the human who is operating it.

After computing the affinity matrix with Eq. 1, we perform normalization on each row of the matrix so that the sum of all the edge values connected to one proposal i will be 1. Motivated by the recent works [85,58], we adopt the softmax function for normalization to obtain the similarity graph,

$$\mathbf{G}^{sim}_{ij} = \frac{\exp F(\mathbf{x}_i, \mathbf{x}_j)}{\sum_{j=1}^{\mathrm{N}} \exp F(\mathbf{x}_i, \mathbf{x}_j)}. \tag{2}$$

4.3 Spatial-Temporal Graph

Although the similarity graph captures even the long term dependencies between any two object proposals, it does not capture the relative spatial relation between objects and the ordering of the state changes. To encode these spatial and temporal relations between objects, we propose to use spatial-temporal graphs, where objects in nearby locations in space and time are connected together.

Given a object proposal in frame t, we calculate the value of Intersection Over Unions (IoUs) between this object bounding box and all other object bounding boxes in frame $t+1$. We denote the IoU between object i in frame t and object j in frame $t+1$ as σ_{ij}. If σ_{ij} is larger than 0, we will link object i to object j using a directed edge $i \to j$ with value σ_{ij}. After assigning the edge values, we normalize the sum of the edge values connected to each proposal i to be 1 by

$$\mathbf{G}^{front}_{ij} = \frac{\sigma_{ij}}{\sum_{j=1}^{\mathrm{N}} \sigma_{ij}}, \tag{3}$$

where $\mathbf{G}^{front}$ is taken as the adjacency matrix for a spatial-temporal graph. We visualize some of the object proposals and the trajectories in Figure 4.

Besides building the forward graph which connects objects from frame t to frame $t+1$, we also construct a backward graph in a similar way which connect objects from frame $t+1$ to frame t. We denote the adjacency matrix of this backward graph as $\mathbf{G}^{back}$. Specifically, for the overlapping object i in frame t and object j in frame $t+1$, we construct an edge $i \leftarrow j$ and assign the values to $\mathbf{G}^{back}_{ji}$ according to the IoU values. By building the spatial-temporal graphs in a bidirectional manner, we can obtain richer structure information and enlarge the number of propagation neighborhoods during graph convolutions.

5 Convolutions on Graphs

To perform reasoning on the graph, we apply the Graph Convolutional Networks (GCNs) proposed in [19]. Different from standard convolutions which operates on a local regular grid, the graph convolutions allow us to compute the response of a node based on its neighbors defined by the graph relations. Thus performing graph convolutions is equal to performing message passing inside the graphs. The outputs of the GCNs are updated features of each object node, which can be aggregated together for video classification. We can represent one layer of graph convolutions as,

$$\mathbf{Z} = \mathbf{G}\mathbf{X}\mathbf{W}, \tag{4}$$

where $\mathbf{G}$ represents one of the adjacency graph we have introduced ($\mathbf{G}^{sim}$, $\mathbf{G}^{front}$ or $\mathbf{G}^{back}$) with N $\times$ N dimensions, $\mathbf{X}$ is the input features of the object nodes in the graph with N $\times$ d dimensions, and $\mathbf{W}$ is the weight matrix of the layer with dimension $d \times d$ in our case. Thus the output of one graph convoltuional layer $\mathbf{Z}$ is still in $N \times d$ dimensions. The graph convolution operation can be stacked into multiple layers. After each layer of graph convolutions, we apply two non-linear functions including the Layer Normalization [86] and then ReLU before the feature $\mathbf{Z}$ is forwarded to the next layer.

To combine multiple graphs in GCNs, we can simply extend Eq. 4 as,

$$\mathbf{Z} = \sum_i \mathbf{G}_i \mathbf{X} \mathbf{W}_i, \tag{5}$$

where $\mathbf{G}_i$ indicates different types of graphs, and the weights for different graphs $\mathbf{W}_i$ are not shared. Note that in this way, each hidden layer of the GCN is updated though the relationships from different graphs. However, we find that the direct combination of 3 graphs ($\mathbf{G}^{sim}$, $\mathbf{G}^{front}$ and $\mathbf{G}^{back}$) with Eq. 5 actually hurts the performance compared to the situation with a single similarity graph.

The reason is that our similarity graph $\mathbf{G}^{sim}$ contains learnable parameters (Eq. 1) and requires back propagation for updating, while the other two graphs do not require learning. Fusing these graphs together in every GCN layer increases the optimization difficulties. Thus we create two branches of graph convolutional networks, and only fuse the results from two GCNs in the end: one GCN adopts Eq. 4 with $\mathbf{G}^{sim}$ and the other GCN adopts Eq. 5 with $\mathbf{G}^{front}$ and $\mathbf{G}^{back}$. These two branches of GCNs perform convolutions separately for L layers and the final layer features are summed together, which is in N $\times$ d dimensions.

Video Classification. As illustrated in Figure 2, the updated features after graph convolutions are forwarded to an average pooling layer, which calculates the mean of all the proposal features and leads to a $1 \times d$ dimensions representation. Besides the GCN features, we also perform average pooling on the whole video representation and obtain the another $1 \times d$ dimensions global features. These two features are then concatenated together for video classification.

6 Experiments

We perform the experiments on two recent challenging datasets: Charades [20] and Something-Something [21]. We first introduce the implementation details.

Training. The training of our backbone models involves pre-training on 2 different datasets following [58,8]. The model is first pre-trained as a 2D ConvNet with the ImageNet dataset [87] and then inflated into a 3D ConvNet (i.e., I3D) as [8]. We then fine-tuned the 3D ConvNet with the Kinetics action recognition dataset [88] following the same training scheme for longer sequences (around 5 second video) in [58]. Given this initialization, we now introduce how to further fine-tune the network on our target datasets as following.

As specified in Table 1, our network takes 32 video frames as inputs. These 32 video frames are sampled in the frame rate of 6fps, thus the temporal length of the video clip is around 5 seconds. The spatial dimensions for input is 224×224. Following [89], the input frames are randomly cropped from a randomly scaled video whose shorter side is sampled in $[256, 320]$ dimensions. To reduce the number of GCN parameters, we add one more $1 \times 1 \times 1$ convolutional layer on top of the I3D baseline model, which reduces the output channel number from 2048 to $d = 512$. As Charades and Something-Something dataset are in similar scales in number of video frames, we adopt the same learning rate schedule.

Our baseline I3D model is trained with a 4-GPU machine. The total batch size is 8 clips during training. Note that we freeze the parameters in all Batch Normalization layers during training. Our model is trained for 100K iterations in total, with learning rate 0.00125 in the first 90K iterations and it is reduced by a factor of 10 during training the last 10K iterations. Dropout [90] is applied on the last global pooling layer with a ratio of 0.3.

We set the layer number of our Graph Convolutional Network to 3. The first two layers are randomly initliazed and the last layer is initialized as zero inspired by [91]. To train the GCN together with the I3D backbone, we propose to apply stage-wise training. We first finetune the I3D model as mentioned above, then we apply RoIAlign and GCN on top of the final convolutional features as shown in Figure 2. We fix the I3D features and train the GCN with the same learning rate schedules as for training the backbone. Then we train the I3D and GCN together end-to-end for 30K more iterations with the reduced learning rate.

Task specific settings. We apply different loss functions when training for Charades and Something-Something datasets. For Something-Something dataset, we apply the softmax loss function. For Charades, we apply binary sigmoid loss to handle the multi-label property. We also extract different numbers of object

model, R50, I3D	mAP
baseline	31.8
Proposal+AvgPool	32.1
Spatial-Temporal GCN	34.2
Similarity GCN	35.0
Joint GCN	**36.2**

(a) GCN ablation studies.

model, R50, I3D	mAP
baseline	31.8
Non-local	33.5
Joint GCN	36.2
Non-local + Joint GCN	**37.5**

(b) Comparing our approach with Non-local Net.

Table 2. Ablations on Charades. We show the mean Average Precision (mAP%).

bounding boxes with RPN in two different datasets. For Charades, the scenes are more cluttered and we extract 50 object proposals for each frame. However, for Something-Something, there is usually only one or two objects in the center of video frame. We find that extracting 10 object proposals per frame is enough.

Inference. We perform fully-convolutional inference in space as [89,58] during inference. The shorter side of each video frame is scaled to 256 while maintaining the aspect ratios. During testing one whole video, we sample 10 clips for Charades and 2 clips for Something-Something according to the average video length in two different datasets. Scores from multiple clips are aggregated by Max-Pooling.

6.1 Experiments on Charades

In the Charades experiments, we follow the official split with 8K training videos and 1.8K validation videos. The average video duration is 30 seconds. There are 157 action classes and multiple actions can happen at the same time.

How much each graph helps? We first perform analysis on each component of our framework, with the backbone of ResNet-50 I3D, as illustrated in Table 2a. We first show that the result of I3D baseline without any proposal extractions and graph convolutions is 31.8% mAP on the validation set.

One simple extension on this baseline is: obtain the region proposals with RPN, extract the features for each proposal and perform average pooling over them as an extra feature. We concatenate the video level feature and the proposal feature together for classification. However, we can only obtain 0.3% boost with this approach. Thus, a naive aggregation of proposal features does not help much.

We then perform evaluations using GCNs with the similarity graph and the spatial-temporal graph individually. We observe that our GCN with only spatial-temporal graph can obtain a 2.4% boost over the baseline and achieve 34.2%. With the similarity graph, we can achieve a better performance of 35.0%. By combining two graphs together and train GCNs with multiple relations, our method achieves 36.2% mAP which is a significant boost of 4.4% over the baseline.

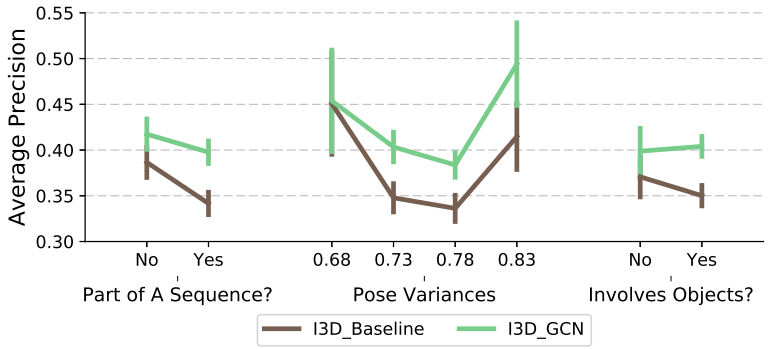

Figure 5. Error Analysis. We compare our approach against baseline I3D approach across three different attributes. Our approach improves significantly when action is part of sequence, involves interaction with objects and has high pose variance.

Robustness to Proposal Numbers. We also analyze how the number of object proposals generated by the RPN affect our method. Note that our method achieves 36.2% with extracting 50 object proposals per video frame. If we use 25 (100) proposals per frame, the mAP of our method is 35.9% (36.1%) mAP. Thus our approach is actually very stable with the changes of RPN.

Model Complexity. Given this large improvement in performance, the extra computation cost of the GCN over the baseline is actually very small. In the Charades dataset, our graph is defined based on 800 object nodes per video (with 16 output frames and 50 object proposals per frame). The FLOPs of the baseline I3D model is 153×10^9 and the total FLOPs of our model (I3D + Joint GCN) is 158×10^9. Thus there is only around 3% increase in FLOPs.

Comparing to the Non-local Net. One of the related work is the recent proposed Non-local Neural Networks [58], where they propose to perform spatial-temporal reasoning on different layers of feature maps. As shown in Table 2b, the Non-local operations gives 1.7% improvements over the baseline and our approach performs 2.7% better than the Non-local Net. These two approaches are actually complementary to each other. By replacing the I3D backbone with Non-local Net, we have another 1.3% boost, leading to 37.5%.

Error analysis Given this significant improvements, we will also like to find out in what cases our methods improve over the baselines most. Following the attributes set up in [11], we show 3 different situations where our approach get more significant gains over the baselines in Figure 5. Specifically, for each video in Charades, besides the action class labels, it is also labeled with different attributes (e.g., whether the actions are happening in a sequence?).

Part of A Sequence? This attribute specifies whether an action category is part of a sequence of actions. For example, "holding a cup" and then "sitting down" are usually in a sequence of actions, while "running" often happens in

model	backbone	modality	mAP
2-Stream [93]	VGG16	RGB + flow	18.6
2-Stream +LSTM [93]	VGG16	RGB + flow	17.8
Asyn-TF [93]	VGG16	RGB + flow	22.4
MultiScale TRN [36]	Inception	RGB	25.2
I3D [8]	Inception	RGB	32.9
I3D [58]	ResNet-101	RGB	35.5
NL I3D [58]	ResNet-101	RGB	37.5
NL I3D + GCN	ResNet-50	RGB	37.5
I3D + GCN	ResNet-101	RGB	39.1
NL I3D + GCN	ResNet-101	RGB	**39.7**

Table 3. Classification mAP (%) in the **Charades** dataset [20]. NL indicates Non-Local.

isolation. As shown in the left plots in Figure 5, the baseline I3D method fails dramatically when an action is part of a sequence of actions, while our approach is more stable. If an action is not happening in isolation, we have more than 5% gain over the baseline.

Pose Variances. This attribute is computed by averaging the Procrustes distance [92] between any two poses in an action category. If the average distance is large, it means the poses change a lot in an action. As visualized in the middle plots in Figure 5, we can see that our approach has similar performance as the baseline when the pose variance is small. However, the performance of the baseline drops dramatically again as the variance of pose becomes larger (from 0.68 to 0.73) in the action, while the slope of our curve is much smaller. The performance of both approaches improve as the pose variability reaches 0.83, where our approach has around $8\% \sim 9\%$ boost over the baseline.

Involves Objects? This attribute specifies whether an object is involved in the action. For example, "drinking from a cup" involves the object cup while "running" does not require interactions with objects. As shown in the right plots in Figure 5, we can see the baseline perform worse when the actions require interactions with objects. Interestingly, our approach actually performs slightly better when objects are involved.

Thus our approach is better in modeling a long sequence of actions as well as actions that require object interactions, and robust to pose changes.

Training with a larger backbone. Besides the ResNet-50 backbone architecture, we also verify our method on a much larger backbone model which is applied in [58]. This backbone is larger than our baseline in 3 aspects: (i) instead of using ResNet-50, this backbone is based on the ResNet-101 architecture; (ii) instead of using 224×224 spatial inputs, this backbone takes in 288×288 images; (iii) instead of sampling 32 frames with 6fps, this backbone performs sampling more densely by using 128 frames with 24fps as inputs. Note that the temporal output dimension of both our baseline model and this ResNet-101 backbone are still the same (16 dimensions). With all the modifications on the backbone architecture, the FLOPs are 3 times as many as our ResNet-50 baseline model.

model	backbone	*val* top-1	*val* top-5	*test* top-1
C3D [21]	C3D[7]	-	-	27.2
MultiScale TRN [36]	Inception	34.4	63.2	33.6
I3D	ResNet-50	41.6	72.2	-
I3D + GCN	ResNet-50	43.3	75.1	-
NL I3D	ResNet-50	44.4	76.0	-
NL I3D + GCN	ResNet-50	46.1	76.8	45.0

Table 4. Classification accuracy (%) in the **Something-Something** dataset [21].

We show the results together with all the state-of-the-art methods in Table 3. The Non-local Net [58] with ResNet-101 backbone achieves the mAP of 37.5%. We can actually obtain the same performance with our method by using a much smaller ResNet-50 backbone (with around 1/3 FLOPs). By applying our method with the ResNet-101 backbone, our method (I3D+GCN) can still give 3.6% improvements and reaches 39.1%. This is another evidence showing that our method is modeling very different things from just increasing the spatial inputs and the depth of the ConvNets. By combining the non-local operation together with our approach, we obtain the final performance of 39.7%.

6.2 Experiments on Something-Something

In the Something-Something dataset, there are 86K training videos, around 12K validation videos and 11K testing videos. The number of classes is 174. The data in the Something-Something dataset is very different from the Charades dataset. In Charades, most of the actions are performed by agents in a cluttered indoor scene. However, in Something-Something, all videos are object centric and there is usually only one or two hands in actions. The background in the Something-Something dataset is also very clean in most cases.

We report our results in Table 4. The baseline I3D approach achieves 41.6% in top-1 accuracy and 72.2% in top-5 accuracy. By applying our method with the I3D backbone (I3D + Joint GCN), we achieve 1.7% improvements in the top-1 accuracy. We observe that the improvement of top-1 accuracy here is not as huge as the gains we have in the Charades dataset. The reason is mainly because the videos are already well calibrated with objects in the center of the frames.

We have also combined our method with the Non-local Net. As shown in Table 4, the Non-local I3D method achieves 44.4% in top-1 accuracy. By combining our approach with the Non-local Net, we achieve another 1.7% gain in top-1 accuracy, which leads to the state-of-the-art results 46.1%. We also test our final model on the test set by submitting to the official website. By using a single RGB model, we achieve the best result 45.0% in the leaderboard.

Acknowledgement: This work was supported by ONR MURI N000141612007, Sloan Fellowship, Okawa Fellowship to AG and Facebook Fellowship, NVIDIA Fellowship, Baidu Scholarship to XW. We would like to thank Xinlei Chen, Gunnar Sigurdsson, Yin Li, Ross Girshick and Kaiming He for helpful discussions.

References

1. Simonyan, K., Zisserman, A.: Two-stream convolutional networks for action recognition in videos. In: Neural Information Processing Systems (NIPS). (2014) 2, 3
2. Wang, L., Xiong, Y., Wang, Z., Qiao, Y., Lin, D., Tang, X., Van Gool, L.: Temporal segment networks: Towards good practices for deep action recognition. In: ECCV. (2016) 2, 3
3. Yue-Hei Ng, J., Hausknecht, M., Vijayanarasimhan, S., Vinyals, O., Monga, R., Toderici, G.: Beyond short snippets: Deep networks for video classification. In: Computer Vision and Pattern Recognition (CVPR). (2015) 2, 3
4. Donahue, J., Hendricks, L.A., Rohrbach, M., Venugopalan, S., Guadarrama, S., Saenko, K., Darrell, T.: Long-term recurrent convolutional networks for visual recognition and description. In: Computer Vision and Pattern Recognition (CVPR). (2015) 2, 3
5. Li, F., Gan, C., Liu, X., Bian, Y., Long, X., Li, Y., Li, Z., Zhou, J., Wen, S.: Temporal modeling approaches for large-scale youtube-8m video understanding. arXiv preprint arXiv:1707.04555 (2017) 2, 3
6. Miech, A., Laptev, I., Sivic, J.: Learnable pooling with context gating for video classification. arXiv preprint arXiv:1706.06905 (2017) 2
7. Tran, D., Bourdev, L., Fergus, R., Torresani, L., Paluri, M.: Learning spatiotemporal features with 3d convolutional networks. In: International Conference on Computer Vision (ICCV). (2015) 2, 14
8. Carreira, J., Zisserman, A.: Quo vadis, action recognition? a new model and the kinetics dataset. In: Computer Vision and Pattern Recognition (CVPR). (2017) 2, 3, 4, 6, 10, 13
9. Tran, D., Wang, H., Torresani, L., Ray, J., LeCun, Y., Paluri, M.: A closer look at spatiotemporal convolutions for action recognition. In: CVPR. (2018) 2, 3
10. Xie, S., Sun, C., Huang, J., Tu, Z., Murphy, K.: Rethinking spatiotemporal feature learning for video understanding. In: arXiv:1712.04851. (2017) 2
11. Sigurdsson, G.A., Russakovsky, O., Gupta, A.: What actions are needed for understanding human actions in videos? In: ICCV. (2017) 2, 12
12. Gupta, A., Kembhavi, A., Davis, L.S.: Observing human-object interactions: Using spatial and functional compatibility for recognition. Transactions on Pattern Analysis and Machine Intelligence (TPAMI) (2009) 2, 3
13. Yao, B., Fei-Fei, L.: Modeling mutual context of object and human pose in human-object interaction activities. In: Computer Vision and Pattern Recognition (CVPR). (2010) 2, 3
14. Yatskar, M., Zettlemoyer, L., Farhadi, A.: Situation recognition: Visual semantic role labeling for image understanding. In: CVPR. (2016) 2, 3
15. Hu, H., Gu, J., Zhang, Z., Dai, J., Wei, Y.: Relation networks for object detection. In: Computer Vision and Pattern Recognition (CVPR). (2018) 2, 3
16. Gkioxari, G., Girshick, R., Dollár, P., He, K.: Detecting and recognizing human-object intaractions. CVPR (2018) 2, 3
17. Lu, C., Krishna, R., Bernstein, M., Fei-Fei, L.: Visual relationship detection with language priors. ECCV (2016) 2
18. Gkioxari, G., Girshick, R., Malik, J.: Contextual action recognition with r*cnn. In: ICCV. (2015) 2
19. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. In: International Conference on Learning Representations (ICLR). (2017) 3, 4, 9

20. Sigurdsson, G.A., Varol, G., Wang, X., Farhadi, A., Laptev, I., Gupta, A.: Hollywood in homes: Crowdsourcing data collection for activity understanding. In: European Conference on Computer Vision (ECCV). (2016) 3, 10, 13
21. Goyal, R., Kahou, S.E., Michalski, V., Materzynska, J., Westphal, S., Kim, H., Haenel, V., Fründ, I., Yianilos, P., Mueller-Freitag, M., Hoppe, F., Thurau, C., Bax, I., Memisevic, R.: The "something something" video database for learning and evaluating visual common sense. arXiv:1706.04261 (2017) 3, 10, 14
22. Laptev, I.: On space-time interest points. IJCV (2005) 3
23. Wang, H., Schmid, C.: Action recognition with improved trajectories. In: International Conference on Computer Vision (ICCV). (2013) 3
24. Klaser, A., Marszalek, M., Schmid, C.: A spatio-temporal descriptor based on 3d-gradients. In: BMVC. (2008) 3
25. Dalal, N., Triggs, B., Schmid, C.: Human detection using oriented histograms of flow and appearance. In: ECCV. (2006) 3
26. Laptev, I., Marszalek, M., Schmid, C., Rozenfeld, B.: Learning realistic human actions from movies. In: CVPR. (2008) 3
27. Sadanand, S., Corso, J.J.: Action bank: A high-level representation of activity in video. In: CVPR. (2012) 3
28. Wang, Y., Mori, G.: Hidden part models for human action recognition: Probabilistic vs. max-margin. TPAMI (2011) 3
29. Zhu, J., Wang, B., Yang, X., Zhang, W., Tu, Z.: Action recognition with actons. In: ICCV. (2013) 3
30. Peng, X., Zou, C., Qiao, Y., Peng, Q.: Action recognition with stacked fisher vectors. In: ECCV. (2014) 3
31. Lan, Z., Lin, M., Li, X., Hauptmann, A.G., Raj, B.: Beyond gaussian pyramid: Multi-skip feature stacking for action recognition. In: CVPR. (2015) 3
32. Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., Fei-Fei, L.: Large-scale video classification with convolutional neural networks. In: CVPR. (2014) 3
33. Wang, L., Qiao, Y., Tang, X.: Action recognition with trajectory-pooled deep-convolutional descriptors. In: CVPR. (2015) 3
34. Taylor, G.W., Fergus, R., LeCun, Y., Bregler, C.: Convolutional learning of spatio-temporal features. In: ECCV. (2010) 3
35. Le, Q.V., Zou, W.Y., Yeung, S.Y., Ng, A.Y.: Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis. In: CVPR. (2011) 3
36. Zhou, B., Andonian, A., Torralba, A.: Temporal relational reasoning in videos. arXiv (2017) 3, 13, 14
37. Wang, X., Farhadi, A., Gupta, A.: Actions ˜ transformations. In: CVPR. (2016) 3
38. Srivastava, N., Mansimov, E., Salakhutdinov, R.: Unsupervised learning of video representations using lstms. arXiv:1502.04681 (2015) 3
39. Sun, C., Shetty, S., Sukthankar, R., Nevatia, R.: Temporal localization of fine-grained actions in videos by domain transfer from web images. In: ACM Multimedia. (2015) 3
40. Wu, Z., Wang, X., Jiang, Y.G., Ye, H., Xue, X.: Modeling spatial-temporal clues in a hybrid deep learning framework for video classification. arXiv:1504.01561 (2015) 3
41. Gan, C., Yao, T., Yang, K., Yang, Y., Mei, T.: You lead, we exceed: Labor-free video concept learning by jointly exploiting web videos and images. In: CVPR. (2016) 3

42. Bian, Y., Gan, C., Liu, X., Li, F., Long, X., Li, Y., Qi, H., Zhou, J., Wen, S., Lin, Y.: Revisiting the effectiveness of off-the-shelf temporal modeling approaches for large-scale video classification. arXiv:1708.03805 (2017) 3
43. Pan, P., Xu, Z., Yang, Y., Wu, F., Zhuang, Y.: Hierarchical recurrent neural encoder for video representation with application to captioning. In: CVPR. (2016) 3
44. Ji, S., Xu, W., Yang, M., Yu, K.: 3d convolutional neural networks for human action recognition. TPAMI (2013) 3
45. Tran, D., Bourdev, L., Fergus, R., Torresani, L., Paluri, M.: Learning spatiotemporal features with 3d convolutional networks. In: ICCV. (2015) 3
46. Xie, S., Girshick, R., Dollár, P., Tu, Z., He, K.: Aggregated residual transformations for deep neural networks. In: Computer Vision and Pattern Recognition (CVPR). (2017) 3
47. Qiu, Z., Yao, T., Mei, T.: Learning spatio-temporal representation with pseudo-3d residual networks. In: ICCV. (2017) 3
48. Feichtenhofer, C., Pinz, A., Wildes, R.: Spatiotemporal residual networks for video action recognition. In: Neural Information Processing Systems (NIPS). (2016) 3, 6
49. Yao, J., Fidler, S., Urtasun, R.: Describing the scene as a whole: Joint object detection, scene classification and semantic segmentation. In: CVPR. (2012) 3
50. Kumar, M.P., Koller, D.: Efficiently selecting regions for scene understanding. In: CVPR. (2010) 3
51. Russell, B.C., Freeman, W.T., Efros, A.A., Sivic, J., Zisserman, A.: Using multiple segmentations to discover objects and their extent in image collections. In: CVPR. (2006) 3
52. Santoro, A., Raposo, D., Barrett, D.G., Malinowski, M., Pascanu, R., Battaglia, P., Lillicrap, T.: A simple neural network module for relational reasoning. In: Neural Information Processing Systems (NIPS). (2017) 3
53. Battaglia, P., Pascanu, R., Lai, M., Rezende, D.J., et al.: Interaction networks for learning about objects, relations and physics. In: Neural Information Processing Systems (NIPS). (2016) 3
54. Watters, N., Tacchetti, A., Weber, T., Pascanu, R., Battaglia, P., Zoran, D.: Visual interaction networks. In: Neural Information Processing Systems (NIPS). (2017) 3
55. Ma, C.Y., Kadav, A., Melvin, I., Kira, Z., AlRegib, G., Graf, H.P.: Attend and interact: Higher-order object interactions for video understanding. In: CVPR. (2018) 3
56. Gkioxari, G., Girshick, R., Malik, J.: Actions and attributes from wholes and parts. In: ICCV. (2015) 3
57. Ni, B., Yang, X., Gao, S.: Progressively parsing interactional objects for fine grained action detection. In: CVPR. (2016) 3
58. Wang, X., Girshick, R., Gupta, A., He, K.: Non-local neural networks. In: CVPR. (2018) 4, 6, 8, 10, 11, 12, 13, 14
59. Lafferty, J., McCallum, A., Pereira, F.C.: Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In: International Conference on Machine Learning (ICML). (2001) 4
60. Krähenbühl, P., Koltun, V.: Efficient inference in fully connected crfs with gaussian edge potentials. In: Neural Information Processing Systems (NIPS). (2011) 4
61. Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: Semantic image segmentation with deep convolutional nets and fully connected CRFs. arXiv:1412.7062 (2014) 4
62. Zheng, S., Jayasumana, S., Romera-Paredes, B., Vineet, V., Su, Z., Du, D., Huang, C., Torr, P.H.: Conditional random fields as recurrent neural networks. In: International Conference on Computer Vision (ICCV). (2015) 4

63. Chandra, S., Usunier, N., Kokkinos, I.: Dense and low-rank Gaussian CRFs using deep embeddings. In: International Conference on Computer Vision (ICCV). (2017) 4
64. Schwing, A.G., Urtasun, R.: Fully connected deep structured networks. arXiv preprint arXiv:1503.02351 (2015) 4
65. Krähenbühl, P., Koltun, V.: Efficient inference in fully connected crfs with gaussian edge potentials. In: NIPS. (2011) 4
66. Harley, A., Derpanis, K., Kokkinos, I.: Segmentation-aware convolutional networks using local attention masks. In: International Conference on Computer Vision (ICCV). (2017) 4
67. Liu, S., De Mello, S., Gu, J., Zhong, G., Yang, M.H., Kautz, J.: Learning affinity via spatial propagation networks. In: Neural Information Processing Systems (NIPS). (2017) 4
68. Scarselli, F., Gori, M., Tsoi, A.C., Hagenbuchner, M., Monfardini, G.: The graph neural network model. IEEE Transactions on Neural Networks (2009) 4
69. Li, Y., Tarlow, D., Brockschmidt, M., Zemel, R.: Gated graph sequence neural networks. In: ICLR. (2016) 4
70. Marino, K., Salakhutdinov, R., Gupta, A.: The more you know: Using knowledge graphs for image classification. In: CVPR. (2017) 4
71. Yan, S., Xiong, Y., Lin, D.: Spatial temporal graph convolutional networks for skeleton-based action recognition. In: AAAI. (2018) 4
72. Alayrac, J.B., Sivic, J., Laptev, I., Lacoste-Julien, S.: Joint discovery of object states and manipulation actions. In: ICCV. (2017) 4
73. Wu, Z., Fu, Y., Jiang, Y.G., Sigal, L.: Harnessing object and scene semantics for large-scale video understanding. In: CVPR. (2016) 4
74. Heilbron, F.C., Barrios, W., Escorcia, V., Ghanem, B.: Scc: Semantic context cascade for efficient action detection. In: CVPR. (2017) 4
75. Brendel, W., Todorovic, S.: Learning spatiotemporal graphs of human activities. In: ICCV. (2011) 4
76. Chen, C.Y., Grauman, K.: Efficient activity detection with max-subgraph search. In: CVPR. (2012) 4
77. Jain, A., Zamir, A.R., Savarese, S., Saxena, A.: Structural-rnn: Deep learning on spatio-temporal graphs. In: CVPR. (2016) 4
78. Yuan, Y., Liang, X., Wang, X., Yeung, D.Y., Gupta, A.: Temporal dynamic graph lstm for action-driven video object detection. In: ICCV. (2017) 4
79. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: Towards real-time object detection with region proposal networks. In: Neural Information Processing Systems (NIPS). (2015) 4, 6
80. Girshick, R.: Fast R-CNN. In: International Conference on Computer Vision (ICCV). (2015) 4
81. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask R-CNN. In: International Conference on Computer Vision (ICCV). (2017) 4, 7
82. Girshick, R., Radosavovic, I., Gkioxari, G., Dollár, P., He, K.: Detectron. https://github.com/facebookresearch/detectron (2018) 6
83. Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: Computer Vision and Pattern Recognition (CVPR). (2017) 6
84. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft COCO: Common objects in context. In: European Conference on Computer Vision (ECCV). (2014) 6

85. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. In: Neural Information Processing Systems (NIPS). (2017) 8
86. Ba, J.L., Kiros, J.R., Hinton, G.E.: Layer normalization. arXiv preprint arXiv:1607.06450 (2016) 9
87. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L.: ImageNet Large Scale Visual Recognition Challenge. International Journal of Computer Vision (IJCV) (2015) 10
88. Kay, W., Carreira, J., Simonyan, K., Zhang, B., Hillier, C., Vijayanarasimhan, S., Viola, F., Green, T., Back, T., Natsev, P., et al.: The kinetics human action video dataset. arXiv:1705.06950 (2017) 10
89. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: International Conference on Learning Representations (ICLR). (2015) 10, 11
90. Hinton, G.E., Srivastava, N., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.R.: Improving neural networks by preventing co-adaptation of feature detectors. arXiv:1207.0580 (2012) 10
91. Goyal, P., Dollár, P., Girshick, R., Noordhuis, P., Wesolowski, L., Kyrola, A., Tulloch, A., Jia, Y., He, K.: Accurate, large minibatch sgd: Training imagenet in 1 hour. arXiv:1706.02677 (2017) 10
92. Kendall, D.G.: A survey of the statistical theory of shape. Statistical Science (1989) 87–99 13
93. Sigurdsson, G.A., Divvala, S., Farhadi, A., Gupta, A.: Asynchronous temporal fields for action recognition. In: Computer Vision and Pattern Recognition (CVPR). (2017) 13