Supervoxel Attention Graphs for Long-Range Video Modeling

Yang Wang[1], Gedas Bertasius[2], Tae-Hyun Oh[3]*, Abhinav Gupta[2], Minh Hoai[1], Lorenzo Torresani[2]
[1]Stony Brook University, [2]Facebook AI, [3]POSTECH
{wang33,minhhoai}@cs.stonybrook.edu, taehyun@postech.ac.kr, {gedas,gabhinav,torresani}@fb.com

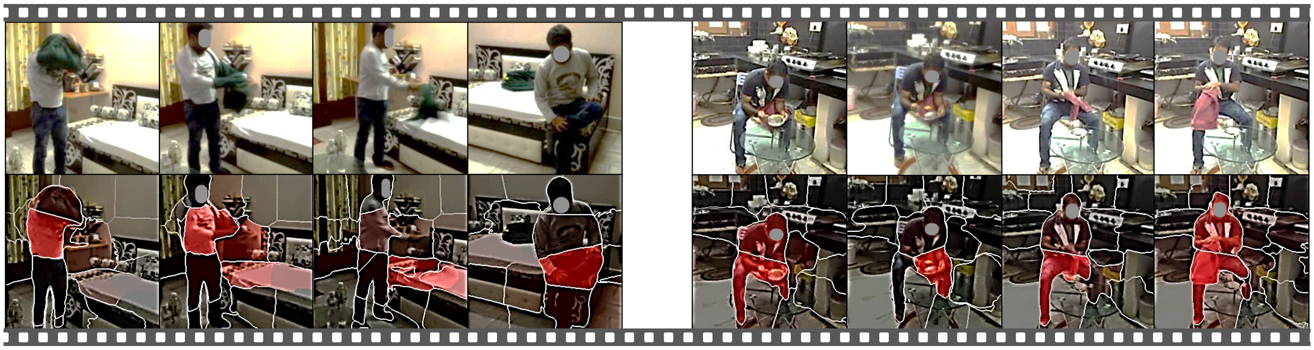


Figure 1: **Visualization of the attention learned by our method.** Our proposed Supervoxel Attention Graph (SVAG) uses groups of pixels, *i.e.*, supervoxels, as nodes for a graph-based representation of the video. A self-attention mechanism over nodes gives rise to a hierarchical representation learned by optimizing action recognition performance. The learned attention scores of each node are illustrated as heat maps for the frames in the figure. Our model spontaneously learns to attend humans, hands, and object regions near hands, by identifying the most relevant spatiotemporal regions for action recognition without direct annotation.

Abstract

A significant challenge in video understanding is posed by the high dimensionality of the input, which induces large computational cost and high memory footprints. Deep convolutional models operating on video apply pooling and striding to reduce feature dimensionality and to increase the receptive field. However, despite these strategies, modern approaches cannot effectively leverage spatiotemporal structure over long temporal extents. In this paper we introduce an approach that reduces a video of 10 seconds to a sparse graph of only 160 feature nodes such that efficient inference in this graph produces state-of-the-art accuracy on challenging action recognition datasets. The nodes of our graph are semantic supervoxels that capture the spatiotemporal structure of objects and motion cues in the video, while edges between nodes encode spatiotemporal relations and feature similarity. We demonstrate that a shallow network that interleaves graph convolution and graph pooling on this compact representation implements an effective mechanism of relational reasoning yielding strong recognition results on both Charades and Something-Something.

1. Introduction

Over the last few years we have witnessed significant progress in video understanding thanks to the emergence of deep spatiotemporal models learned end-to-end and effectively integrating contextual, appearance, motion and temporal features [19, 55, 3, 57, 75, 42, 9, 61, 22, 67, 69]. 3D convolutional neural networks (3D CNNs) in particular have effectively replaced hand-designed spatiotemporal descriptors [28, 59, 25, 6, 29, 45, 68, 86, 40, 27] as the representation of choice for video understanding. 3D CNNs implement a deep stack of *local* spatiotemporal convolutional operations. To capture longer-range dependencies and to reduce the computational cost, pooling and striding are commonly adapted over both spatial and temporal dimensions. Because the size of the receptive field increases with each layer, in theory, stacking more and more layers should enable these models to capture longer-term patterns in the video. However, in practice, due to memory constraints, 3D CNNs have limited depth and are typically optimized over short clips. This limits their ability to capture long-term dependencies. Recent work has leveraged non-local operations [65], self-attention [58, 4], and memory modules [71] to increase the range of spatiotemporal dependencies cap-

[*] This work was done when T.-H. Oh was with Facebook AI Research.

tured by these models. While effective, these mechanisms increase significantly the structural and computational complexity of the model. Thus, the key challenge is to design a model that can model long-term contextual dependencies efficiently and effectively.

For this purpose, we propose a model that captures long-range and contextual dependencies through relational inference over a graph-based representation of the video. The nodes of our graph are semantic supervoxels that conform to the spatiotemporal structure of objects and motion cues in the video, while edges between nodes capture spatiotemporal relations and feature similarity. We leverage *supervoxel pooling* so that each node encodes short-term 3D CNNs features pooled from the supervoxel region. The supervoxels effectively reduce the dimensionality of the convolutional tensor computed by the 3D CNNs, leading to a representation that facilitates efficient semantic relational reasoning over long spatiotemporal volumes.

In our experiments, we demonstrate that 16 nodes (supervoxels) per second are sufficient to provide a rich semantic description of the video. Our graph-based model provides a compact, yet effective representation that captures long-range relations and yields state-of-the-art performance on challenging action recognition datasets. Compared to recent approaches for long-range video modeling which rely on the non-local operation [65] or additional memory [71], our approach is more efficient and achieves higher accuracy.

2. Related Work

Video Action Recognition. Early methods in this genre focused on the manual design of spatiotemporal appearance and motion features that are useful for action recognition [28, 59, 25, 6, 29, 45, 68, 86, 15, 40, 27, 70]. More recently, the surge of deep convolutional models has enabled the learning of powerful spatiotemporal features directly optimized for video action recognition [22, 49, 60, 53, 30, 84, 62, 64, 37]. In the last few years, 3D CNNs [19, 55, 3, 57, 75, 42, 9, 17] have become a dominant approach for short-range video modeling. To leverage longer temporal extents, recurrent architectures [82, 7, 50, 51, 73, 33, 10, 2, 39, 36, 52] were proposed. However, these recurrent models are typically built on top of frame-level or video clip-level holistic scene descriptors, which collapse the scene information into a single vector. As a result, many contextual relationships are not properly captured. To address these shortcomings, we leverage a supervoxel graph representation that allows us to preserve local video information, while also enabling the capability for efficient long-term modeling.

Spatial Visual Relationships. Reasoning about object relationships has been shown useful for improving various computer vision tasks such as object detection [78], scene classification and segmentation [79, 26], visual question answering [46], and image captioning [80, 23]. Methods for visual relationship detection (VRD) [35, 16, 11, 87, 83, 5] and the prior work leveraging scene graphs [21, 13, 63, 77, 41] explicitly model object relationships. However, the basic relationship representation of VRD and scene graphs is restricted to objects from a predefined set of categories. Furthermore, the work in [14, 54, 38] leverages pairwise human-object and object-object relationships for image-level action recognition. All the aforementioned methods aim to model spatial relationships within a static image, whereas our goal is to capture spatiotemporal relationships useful for long-term video modeling.

Modeling Spatiotemporal Relationships. Non-local networks [65] were proposed for modeling unconstrained pairwise relations in both space and time. However, because a non-local operator is applied to every pixel of a given feature map, this model becomes computationally prohibitive when considering long-range videos. Additionally, a recently introduced long-term feature bank (LFB) method [71] deploys an external memory that stores features from a long temporal window, which is subsequently used for action classification. However, while LFB method is capable of capturing long-range dependencies, due to the structure of its memory indexing, it cannot model spatial relationships within individual video frames.

The model that is closest to ours is the space-time region graph of Wang and Gupta [66]. Their graph uses region proposals obtained from object detectors as nodes. However, top-ranked object proposals only capture a sparse set of the video content, essentially only regions corresponding to specific object classes that the detector was trained to recognize. Such an approach discards too much information too early and biases the graph to rely exclusively on information relating to this predefined set of classes. In comparison, we adopt a mid-level supervoxel [81, 18] node representation that allows us to model objects and their parts regardless of categories. Furthermore, using mid-level supervoxels as our graph nodes allows us to build a compact video representation that preserves relevant spatiotemporal cues. Unlike the method of Wang and Gupta [66], our approach condenses the information from the entire video in a hierarchical fashion using attention modules that selectively "attend" the most relevant regions for the end task. Our experiments show that such hierarchical filtering of information leads to significantly higher action recognition accuracy compared to that produced by space-time region graphs [66].

3. Supervoxel Attention Graphs

Our goal is to design a model that captures cues related to humans and object interactions within a long-range input video. To achieve this goal we adopt a graph-based representation, which we refer to as a Supervoxel Attention Graph (SVAG). Given an input video, we first decompose it into a set of supervoxels, which serve as our graph nodes. Each node is represented as a 3D CNN feature, pooled within its supervoxel. The edges between nodes are established based on feature similarity. Such a design provides a compact and flexible video representation, which facilitates efficient long-term video modeling. We now describe each component of our approach in more detail.

3.1. Computation of Semantic Supervoxel Nodes

We design our graph node representation with the following characteristics in mind: 1) compactness (i.e., a small number of nodes for representing a long-range video), 2) coverage (i.e., the nodes of the graph should effectively represent most relevant cues in a video), 3) semantic representation (i.e., each node should represent a salient semantic unit such as an object or its part). We argue that supervoxels, i.e., spatiotemporal groups of pixels, meet all of these criteria.

Motivated by [18], we use a soft K-means assignment algorithm for supervoxel computation. Given an input video of N space-time pixels, our algorithm assigns each pixel to one of the K supervoxels. Each pixel is described by a C-dimensional feature vector. We encode the features of all pixels into matrix $\mathbf{F} \in \mathbb{R}^{N \times C}$, and represent the pixel-supervoxel association in matrix $\mathbf{Q} \in [0,1]^{N \times K}$. Our method iteratively updates the association while refining the supervoxel representation. Initially, we subdivide the video according to a regular grid uniformly partitioned along the spatial and temporal axes (Fig. 2a). The cells of this grid are taken as initial supervoxels. Thus, we initialize the supervoxel centroids $\mathbf{S}^0 \in \mathbb{R}^{K \times C}$ by averaging the features of pixels within each cell. We estimate the association map $\mathbf{Q}$ and the supervoxel representation $\mathbf{S}$ by iterating through the following procedure:

1. **Association update.** We compute a normalized similarity measure between a pixel p and a supervoxel k as: $\mathbf{Q}_{(p,k)} = \frac{\exp[-||\mathbf{F}_{(p,:)} - \mathbf{S}_{(k,:)}||^2]}{\sum_i \exp[-||\mathbf{F}_{(p,:)} - \mathbf{S}_{(i,:)}||^2]}$ (see notation[1]). Note that each row of $\mathbf{Q}$ (the association values of a pixel to all supervoxels) sums to one, *i.e.*, $\sum_k \mathbf{Q}_{(p,k)} = 1$.

2. **Centroid update.** The cluster centroid $\mathbf{S}_{(k,:)}$ of supervoxel k is updated by the weighted average of features, where the weights are based on the association strengths: $\mathbf{S} = \mathbf{Q}^\top \mathbf{F}$.

We point out that the choice of pixel-level feature representation is critical for obtaining supervoxels that capture semantic spatiotemporal structures in the video. To achieve this goal, we concatenate the (x, y, t) location and the RGB color values of each pixel to a semantic feature vector computed with an off-the-shelf Unified Perceptual Parsing network [74] (UPerNet) pretrained on the ADE20K dataset [85]. Because UPerNet has been explicitly designed to recognize as many visual concepts as possible from a given image, it fulfills our requirement for a rich semantic description on which we build our mid-level representation.

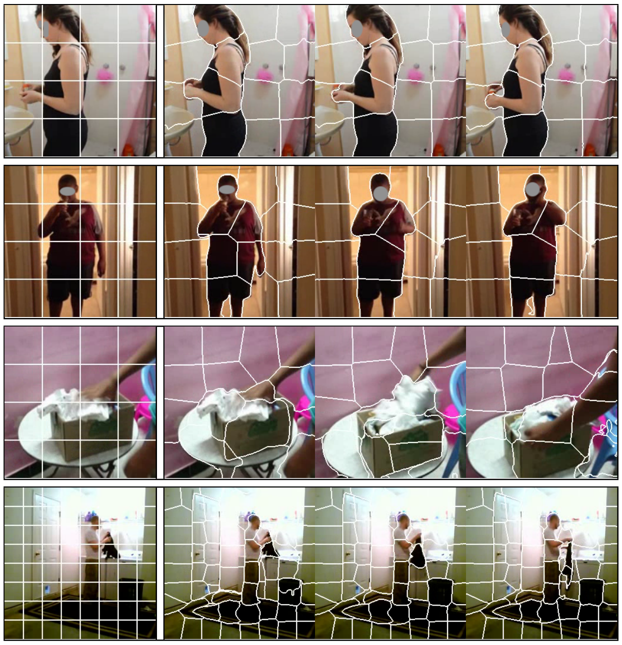

(a)
(b)

Figure 2: **Visualization of supervoxels on Charades dataset.** (a) A sample frame of input video overlayed with the initial supervoxel grid; (b) Supervoxels using segmentation features and TXYRGB channels (spatiotemporal coordinates and color values) computed for three frames of each sequence. The visualization only shows limited slices in the temporal video tubes.

We apply the procedure above, and obtain the final pixel-supervoxel association matrix $\mathbf{H} \in \mathbb{R}^N$ by hardening the soft association: $\mathbf{H}_{(p)} = \text{argmax}_k \, \mathbf{Q}_{(p,k)}$. Figure 2 visualizes examples of supervoxels computed with our procedure.

Supervoxel Pooling. To extract high-level features for each supervoxel, we first feed the original RGB video of size $T \times H \times W \times 3$ through a pre-trained video backbone network (*e.g.* a 3D CNN), which outputs a spatiotemporal tensor $\mathbf{V}$ of size $T' \times H' \times W' \times D$. Given the supervoxel assignment map $\mathbf{H}$ obtained from the supervoxel computation, we then define supervoxel average pooling as:

$$\mathbf{X} = \texttt{svx\_pool}(\mathbf{V}, \mathbf{H}), \tag{1}$$

where a vector $\mathbf{X}_{(k,:)}$ of the k-th supervoxel is obtained by

[1] We denote $(\cdot, \cdot)$ as row-column entry indexing following the Numpy index convention; $(:, \cdot)$ and $(\cdot, :)$ represent the column and row vector indexing, respectively.

averaging all the features within the k-th supervoxel. This effectively compresses a dense video representation $\mathbf{V}$ into a much smaller set of supervoxel features $\mathbf{X} \in \mathbb{R}^{K \times D}$ where $K \ll THW$.

3.2. Graph Network Architecture

Our graph network consists of two distinct branches: a relational branch and a local branch as illustrated in Fig. 3. The relational branch leverages graph convolutions for message passing, implementing relational learning. The local branch processes the feature vector of each node independently to model relevant local information. We now describe each of these branches in more detail.

Relational Branch. To deal with the irregular graph structure, we adopt the framework of graph convolutions [24]. Our model involves a stack of blocks, which interleave graph convolution and hierarchical pooling. The graph pooling operations force the model to identify salient graph nodes, whereas graph convolutions perform message passing to propagate information within the graph.

A distinctive feature of our graph network is the hierarchical node pooling performed using a self-attention mechanism. The block implementing this operation is illustrated in the gray color region of Fig. 3. It consists of graph convolution, skip connection, and self-attention graph pooling.

Let $(\mathbf{Z}_r^0, \mathbf{G}^0)$ represent the initial supervoxel graph that is the input to the relational branch. $\mathbf{Z}_r^0 \in \mathbb{R}^{K \times D}$ is the initial node feature matrix set equal to the original supervoxel features $\mathbf{X}$. $\mathbf{G}^0 \in \mathbb{R}^{K \times K}$ is the normalized graph affinity matrix. Entry $\mathbf{G}^0_{i \leftarrow j} := \mathbf{G}^0_{(i,j)}$ encodes the affinity of the directed edge connecting node j to node i. More details about the computation of $\mathbf{G}^0$ are given in the next section.

We define the graph convolution layer as:

$$\mathtt{GConv}[\mathbf{G}, \mathbf{Z}] \coloneqq \mathtt{LN}[\mathbf{G} f(\mathbf{Z}) \mathbf{W}_r], \tag{2}$$

where $\mathbf{W}_r \in \mathbb{R}^{D' \times D}$ is a trainable weight matrix, and $\mathtt{LN}[\cdot]$ is the layer normalization [1]. For node embedding operation $f(\cdot)$, we use a simple linear embedding, *i.e.*, $f(\mathbf{A}) = \mathbf{A}\mathbf{W}_f$, $\mathbf{W}_f \in \mathbb{R}^{D \times D'}$ $(D' \leq D)$. The formulation above enables our model to propagate relevant spatiotemporal cues from one node to another.

After every graph convolution, we perform graph pooling, which forces the model to identify the most salient graph nodes. To achieve this goal we apply the recently proposed Self-Attention Graph Pooling (SAGPool) [31]. The key idea of SAGPool is to compute an attention score for each node, and use the attention values to rank the nodes. First, the attention scores are computed by $\mathbf{a} = \mathtt{tanh}(\mathbf{Z}\mathbf{w}_{\mathtt{att}}) \in \mathbb{R}^K$ given node features $\mathbf{Z}$. Given a pooling ratio $\eta \in (0, 1]$, which determines the portion of nodes to keep, we obtain the indexes of the top $K' = \lceil \eta K \rceil$ nodes by sorting their respective attention values, *i.e.*, $\mathrm{idx} = \mathtt{top\_rank}(\mathbf{a}, K')$. We retain only the top-ranked nodes for the next round of graph convolutions. Furthermore, we scale their features according to the predicted attention values: $\mathbf{Z}' = \mathbf{Z}_{(\mathtt{idx},:)} \odot \mathbf{a}_{(\mathtt{idx})}$, where $\odot$ is the broadcasted element-wise product. After discarding supervoxels with low attention values, the newly-formed graph becames $\mathbf{G}' = \mathbf{G}_{(\mathtt{idx},\mathtt{idx})}$ followed by row normalization. We denote this entire procedure as:

$$(\mathbf{Z}', \mathbf{G}') = \mathtt{SAGPool}[\mathbf{Z}, \mathbf{G}]. \tag{3}$$

Our graph convolution block can then be expressed as:

$$(\mathbf{Z}_r^b, \mathbf{G}^b) = \mathtt{SAGPool}\left[\mathtt{relu}(\mathbf{Z}_r^{b\text{-}1} + \mathtt{GConv}[\mathbf{G}^{b\text{-}1}, \mathbf{Z}_r^{b\text{-}1}]), \mathbf{G}^{b\text{-}1}\right]$$
$$\text{for } b \geq 1, \tag{4}$$

where b denotes the b-th block of the network within the stack of B blocks. Note that we add a skip connection and a $\mathtt{relu}$ activation after each graph convolution.

One potential issue with this graph pooling design in Eq. (4) is that, it may discard relevant information too early. To mitigate this issue, we attach a lateral connection from $\mathbf{Z}_r^0$, which encodes original node features (see gray regions of Fig. 3). Mathematically, Eq. (4) becomes

$$(\mathbf{Z}_r^b, \mathbf{G}^b) = \mathtt{SAGPool}\left[\mathtt{relu}(\mathbf{Z}_r^{b\text{-}1} + \mathtt{GConv}[\mathbf{G}^{b\text{-}1}, \ \mathbf{Z}_r^0]), \mathbf{G}^{b\text{-}1}\right]$$
$$\text{for } b \geq 1, \tag{5}$$

where $\mathbf{G}^b = \mathbf{G}^{b\text{-}1}_{(\mathtt{idx}^{\mathtt{b\text{-}1}},:)}$ in $\mathtt{SAGPool}$. Note that in this formulation all columns of the affinity matrix are retained. The distinctive feature of Eq. (5) compared to Eq. (4) is that this variant discards the low-attentional supervoxel nodes in the pooled graph, but keeps them in message passing as information source only. This is beneficial because discarded nodes can still provide relevant contextual information. We refer to these two designs as *source-discard* (for the model in Eq. (4)) and *source-preserve* (for the model in Eq. (5)). An ablation study shown in Fig. 5(e) demonstrates the benefit of the source-preserve design.

Local Branch. At each block, our local branch processes and then selects graph nodes according to the pooling indexes obtained from $\mathtt{SAGPool}$ at the corresponding block in the relational branch. The other nodes are discarded, *i.e.*, $\bar{\mathbf{X}} = \mathbf{X}_{(\mathtt{idx},:)}$. Afterwards, we apply a residual layer: $\mathbf{Z}_l = \mathtt{relu}(\bar{\mathbf{X}}^B + \mathbf{W}\bar{\mathbf{X}}^B)$. The final number of remaining nodes is the same as that of the relational branch. We denote the final nodes as $[\mathbf{z}_{l,1}, \cdots, \mathbf{z}_{l,K_o}] \in \mathbb{R}^{D \times K_o}$. The aim of a local branch is to processes each graph node independently to retain relevant local information

Fusion and Classification. We concatenate the final outputs of the branches $[\mathbf{z}_{r,k}, \mathbf{z}_{l,k}]$ for each node, and then use a linear layer to map each feature vector into a class confidence score. The node scores are aggregated differently depending on the task, as discussed in Sec. 4.1. We use the cross-entropy loss to train the final model.

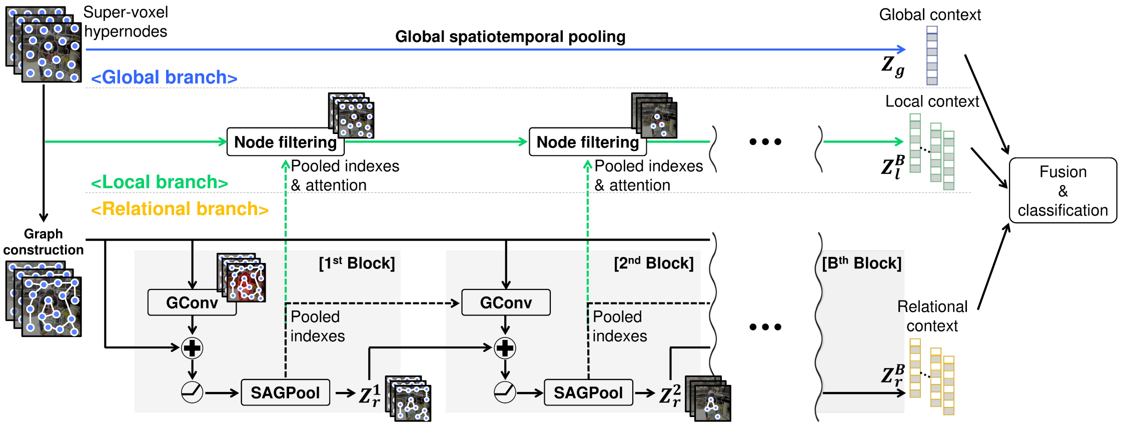


Figure 3: **Illustration of our Spatiotemporal Voxel Attention Graph (SVAG).** To perform reasoning on the graph we use two branches: a relational branch and a local branch (the global branch is not used in our final model but is drawn for reference of the ablation study). The relational branch leverages hierarchical graph convolution and attentional pooling to capture salient long-range dependencies. The local branch processes each node independently using a residual layer.

3.3. Graph Construction

As was done in [66], we construct the graph $\mathbf{G}$ with a learnable similarity metric. Given the node representation $\mathbf{X} \in \mathbb{R}^{K \times D}$, we construct the similarity-based graph $\mathbf{G}^{sim} \in \mathbb{R}^{K \times K}$ with entries computed as:

$$\mathbf{G}^{sim}_{i \leftarrow j} = \frac{\exp[F(\mathbf{x}_i, \mathbf{x}_j)]}{\sum_{j'=1}^{K} \exp[F(\mathbf{x}_i, \mathbf{x}_{j'})]}, \tag{6}$$

where $F(\mathbf{x}_i, \mathbf{x}_j) = \frac{1}{\sqrt{D}} \texttt{query}(\mathbf{x}_i)^\top \texttt{key}(\mathbf{x}_j)$ measures the pairwise similarity, and $\texttt{query}(\cdot)$ and $\texttt{key}(\cdot)$ represent two different transformations defined by simple linear transformations as $\texttt{query}(\mathbf{x}) = \mathbf{W}_q \mathbf{x}$ and $\texttt{key}(\mathbf{x}) = \mathbf{W}_y \mathbf{x}$ with learnable weights $\mathbf{W}_q$ and $\mathbf{W}_y$ following [58].

The similarity measure is learned via backpropagation with supervision from the action recognition task, such that it captures task-specific relations that are useful for recognition. Thus, the directed edge $\mathbf{G}^{sim}_{i \leftarrow j}$ connecting node j to node i is expected to have a high value if two nodes are semantically related.

Wang and Gupta [66] suggested adding another type of hand-designed graph, called spatiotemporal graph. We found that, with our framework, our learnable similarity graph provides superior performance compared to a spatiotemporal graph defined over our supervoxels (we refer to the supplementary material for the corresponding ablation).

Sparsification. We observe that the density of the similarity graph affects the performance of the graph convolutional layers. We adopt a simple but effective sparsification strategy. In the similarity matrix, we zero out the entries below the top $\chi(\%)$ of the similarity scores within each row.

4. Experiments

We conduct experiments on Something-Something[2] [12] and Charades [48]. In this section, we first describe the implementation details specific to the task of each dataset, and then present the results. To understand the behavior of our framework, we perform ablation studies on the Charades validation set. Additional results and further details can be found in the supplementary material.

4.1. Implementation Details

Backbone Models. We use several backbones to extract video features: R101-NL[71], TSM [34], and CSN [56]. The backbones are pre-trained on a pre-training video dataset and then finetuned on the target dataset.

Supervoxel Sampling. As described in Sec. 3.1, when computing supervoxels, we incorporate high-level semantic segmentation maps as another feature in addition to the TXYRGB channels (spatiotemporal coordinates and color values). We use the semantic segmentation model UPerNet [74] trained on the ADE20K [85] dataset, which spans a diverse set of 150 stuff/object categories. We concatenate the activation channels of the network to the low-level features. When concatenating, similarly to [18], we scale each features with scaling hyperparameters $(\lambda_{RGB}, \lambda_{semantic}, \lambda_T, \lambda_{XY})$. The hyperparameter values and details can be found in the supplementary material.

Training & Inference. The datasets used in our experiments involve different tasks: while action recognition on Something^2 is single-label classification, Charades entails a multilabel classification. Thus, the two datasets require slightly different network architectures and experimental

[2] We shorten the name as Something^2 for brevity.

Model	Backbone	Pretrain	mAP
CoViAR [72]	R50	ImageNet	21.9
Asyn-TF [47]	VGG16	ImageNet	22.4
MultiScale TRN [84]	Inception	ImageNet	25.2
Non-Local Network [65]	R101-NL	Kinetics-400	37.5
Space-Time Region Graph [66]	R101-NL	Kinetics-400	39.7
Long-Term Feature Banks [71]	R101-NL	Kinetics-400	42.5
SlowFast [8]	R101-NL	Kinetics-400	42.5
SVAG (ours)	R101-NL	Kinetics-400	**44.1**

Table 1: **Comparison with the state-of-the-art on Charades (RGB modality).** Our proposed graph model (SVAG) outperforms all models considered in this comparison, including Non-Local Networks and Long-Term Feature Banks, which are long-range video models based on the same backbone and the same pretraining dataset.

setups, as also done in [66]. For Charades, we apply the linear classifier to our final output features $\{[\mathbf{z}_{r,i}, \mathbf{z}_{l,i}]\}$, perform max-pooling, and apply the sigmoid function. For Something^2, which has a single-label per sample, we average-pool all the final output features across supervoxels, and apply the linear classifier once followed by the softmax function.

Evaluation Metrics. For the Charades evaluation, we use the metric of mean Average Precision (mAP). Following prior work [66, 71], we sample 10 clips per video and use (left, center, right) 3-crop testing. We use max-pooling to aggregate class confidence scores across all 30 crops. For Something^2, we use the center crop and report the classification accuracy.

4.2. Comparison with the state-of-the-art.

Table 1 compares our approach with state-of-the-art methods on Charades dataset. We include methods that rely on the same backbone (R101-NL) and the same pretraining dataset (Kinetics-400) as our method, as well as some historical baselines. We refer the readers to [8, 56, 44, 43] for more results from leveraging multiple modality or pretraining with larger dataset. As shown in Table 1, our proposed supervoxel-based graph reasoning approach outperforms all recently introduced long-range models based on the same backbone and the same pretraining: the gains of our model are 6.6% over Non-local Networks [65], and 1.6% over Long-Term Feature Banks [71]. The accuracy of SVAG is also 4.4% higher than that produced by the Space-Time Region Graph [66]. As shown in Table 3, in addition to achieving higher accuracy, our model is considerably more efficient and less memory-intensive than Non-local Networks [65].

Table 2 compares our approach with other state-of-the-art methods on the Something^2 dataset. STDF [37] is the most recent state-of-the-art approach on this dataset. However its backbone model is not available at the time of submission of this work. Thus, we have built our approach on top of two other backbone models, TSM [34] and CSN [56]. As shown in Table 2, the proposed graph reasoning improves the performance of both these models. By leveraging the strong CSN backbone and the proposed supervoxel graph reasoning, our SVAG improves over the previous best reported result on this benchmark (53.8% vs 53.4%).

Method	Backbone	Pretrain	# Frame × # Crop	Top-1 val.	Top-5 val.	Top-1 test
S3D-G [76]	Inception	ImageNet	64 × 1	48.2	78.7	42.0
ECO [88]	Inception	Kinetics	16 × 1	41.4	–	–
ECO*Lite*			92 × 1	46.4	–	42.3
TRN [84]	Inception	ImageNet	8 × 1	34.4	–	33.6
MFNet [32]	R101	Scratch	10 × 1	43.9	73.1	37.5
STM [20]	R50	ImageNet	8 × 1	47.5	–	–
			8 × 3	49.2	79.3	–
			16 × 1	49.8	–	–
			16 × 3	50.7	80.4	43.1
STDF [37]	R50	ImageNet	$L \times 2$	50.1	79.5	–
	R152		$L \times 2$	**53.4**	**81.8**	–
TSM [34]	R101	Kinetics	16 × 1	48.3	77.2	–
SVAG (ours)	R101	Kinetics	16 × 1	**49.8**	**77.9**	–
CSN [56]	ip-CSN152	IG-65M	32 × 5	52.9†	83.0	–
SVAG (ours)	ip-CSN152	IG-65M	32 × 5	**53.8**	**83.4**	**49.4**

Table 2: **Comparison with the state-of-the-art on Something^2-v1 (RGB modality).** SVAG based on a R101 backbone outperforms TSM using the same backbone. By taking advantage of a stronger backbone (ip-CSN152), SVAG achieves the best reported number on this benchmark (53.8% on the validation set and 49.4% on the test set). † indicates the performance of our own implementation with 5-crop testing (52.9%) while the original paper [56] used 10-crop testing (53.3%). Our conclusion holds due to consistent relative improvement.

4.3. Qualitative Results.

In Figs. 1 and 4, we visualize the node attention scores as heat maps (mapped into the range of $[0, 1]$) computed by the graph pooling. As it can be observed, although the self-attention mechanism over nodes is learned by optimizing action recognition performance without direct spatiotemporal annotations, our model spontaneously learns to attend humans, hands and object regions near hands.

4.4. Ablation.

In this subsection, we study the accuracy of our model on Charades as we vary the different design choices.

Supervoxel size. As shown in Fig. 2, the supervoxels are initialized with regular grids. The temporal length τ and the spatial width ω of the initial grid cells are two hyperparameters. Fig. 5(a) shows the effect of the initial supervoxel scale (τ, ω). Based on this study, we use temporal length $\tau = 1s$ and spatial width $\omega = 64$pix. for the computation of supervoxels on the Charades dataset.

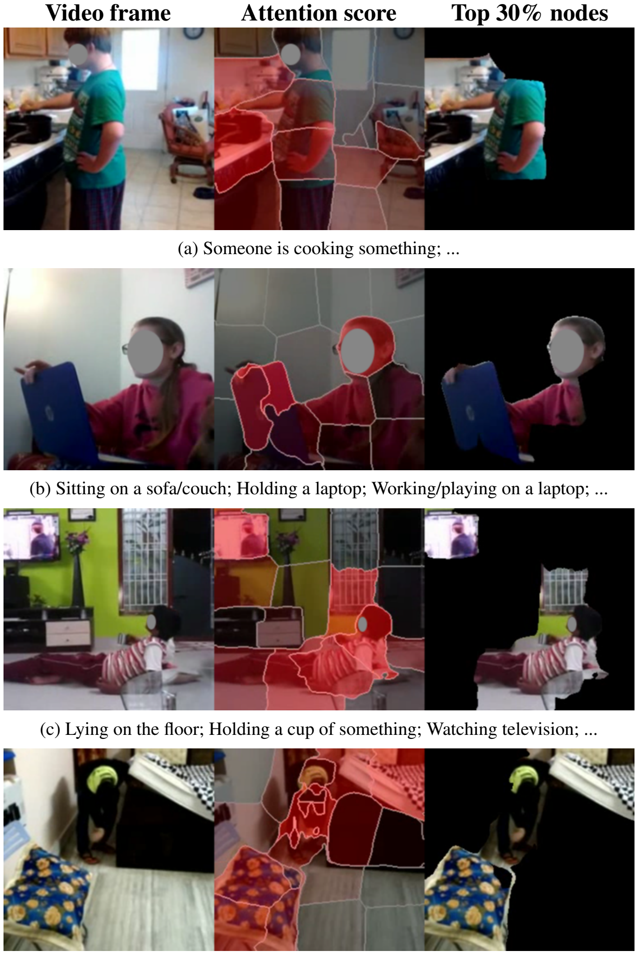
Graph convolutional blocks & pooling rate. Let B be the number of the graph pooling blocks, and η the pooling rate for each pooling block. The total effective pooling rate for the entire graph branch is η^B. Fig. 5(b) shows the effect of the graph convolutional depth by varying B from 1 to 4 while fixing η=100%. This plot suggests that using two layers gives us a good balance. In Fig. 5(c), we fix the number of graph pooling blocks B=2 and decrease η from 100% to 70%. The best performance is achieved for the total effective pooling rate of η^2=81%. Based on these results, we use B=2 and η=90%.

Effect of graph pooling. We also study the effect of graph pooling by removing SAGPool from the proposed network. As shown in Fig. 5(c), removing SAGPool (red diamond marker in the Figure) hurts the recognition performance compared to the top accuracy achieved with SAGPool and a pooling rate $\eta^2 = (90\%)^2$. This demonstrates the effectiveness and benefit of applying graph pooling after every graph convolution, which forces the model to identify the most salient graph nodes.

Enforcing sparsity on $\mathbf{G}^{\texttt{sim}}$. We study the effect of sparsification during the construction of $\mathbf{G}^{\texttt{sim}}$. In Fig. 5(d), we vary the sparsity percentage χ from 10% to 100%, while other hyperparameters are kept the same ($B = 2$ and $\eta = 90\%$). Using top 30% of the graph edges in $\mathbf{G}^{\texttt{sim}}$ yields the best accuracy.

Temporal length T. Fig. 5(e) shows the accuracy of our model as we vary the temporal span T, compared to the R101-NL model and LFB [71]. Setting T =10s yields the best result. Note that here we use 30-crop testing, as described in Sec. 4.1. Following prior works [66, 71], we sample 10 clips per video and use the (left, center, right) crops of each clip for testing. We use max-pooling to aggregate class confidence scores across all 30 crops.

Graph inference over the entire video. While most prior models have high computational costs and large memory footprints which force them to operate on short temporal windows, SVAG yields a compact representation that makes it possible to perform our graph inference efficiently over the entire video. Compared to 30-crop testing using $T = 10$ seconds, graph inference over the entire video achieves slightly lower accuracy (43.6% vs 44.1%). The videos in the Charades dataset on average last around 30 seconds. This suggests that connecting nodes that are too far away may obscure the true context and negatively affect the performance. Therefore, for full video testing, we propose to enforce a temporal constraint on the similarity graph that no edge exists between two supervoxels that are more than $\mathbb{T}$ seconds away. With $\mathbb{T}$ set to 30, 20 and 10, the full video testing achieves 43.8%, 44.0% and 44.2% respectively. While for the case of Charades the improvement of graph inference over the entire video compared to crop-based testing is small (44.2% vs 44.1%), we expect that the compactness and efficiency of our SVAG will be even more beneficial for analysis of videos with contextual dependencies exhibited over longer temporal ranges, where most prior models become computationally unpractical.



(a) Someone is cooking something; ...

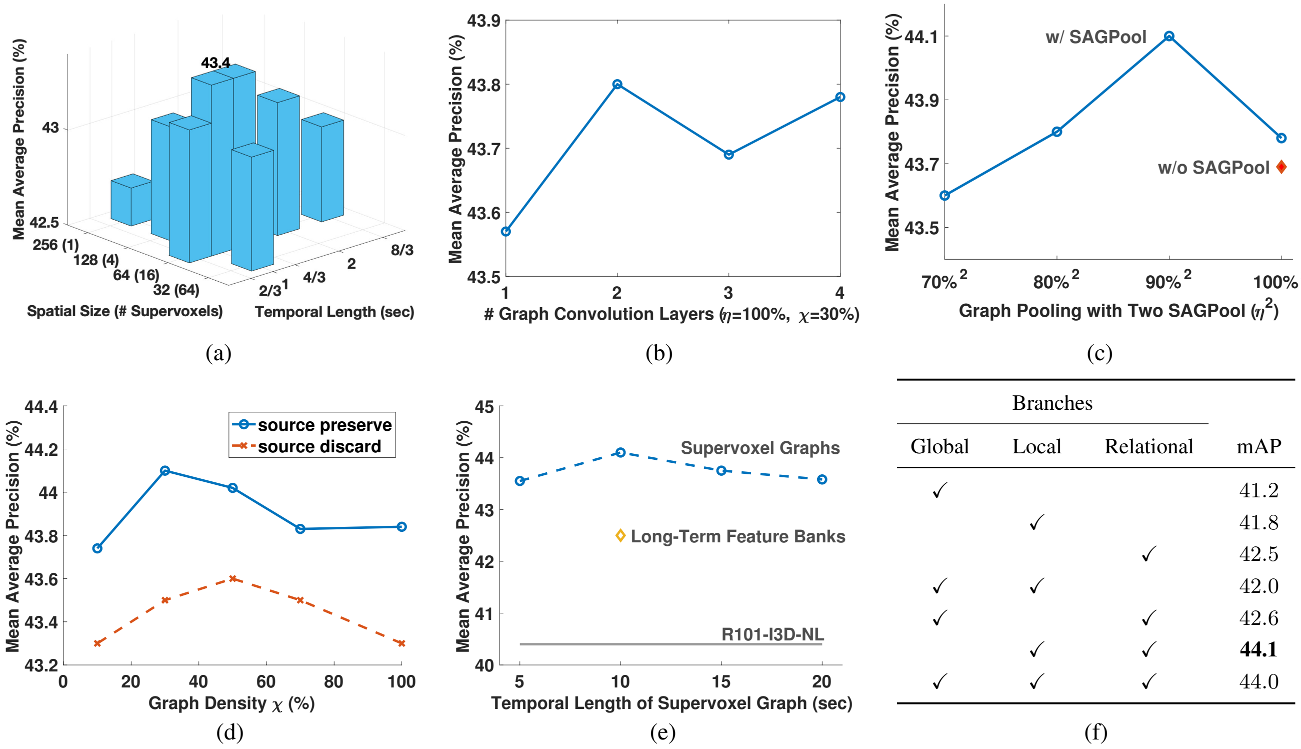
(b) Sitting on a sofa/couch; Holding a laptop; Working/playing on a laptop; ...

(c) Lying on the floor; Holding a cup of something; Watching television; ...

(d) Tidying up a blanket/s; ...

Figure 4: **Visualization of node attention.** Supervoxel attention scores computed for a few sample frames are shown in the 2^{nd} column. The 3^{rd} column shows the top 30% supervoxels according to attention scores. We observe that highly attended supervoxels are those most relevant to the action performed in the video. This demonstrates that our model spontaneously learns to attend regions relating to humans and objects without any spatiotemporal annotation.

Combining global, local, and relational branches. Fig. 5(f) compares different combinations of graph branches (global, local, and relational). We define the global branch as a global average pooling operation performed over all nodes of the graph, *i.e.*, $\mathbf{z}_g = \texttt{avg\_pool}(\mathbf{X})$. This operation allows us to capture global context of a given video. We additionally concatenate $\mathbf{z}_g$ as an input to the fusion layer for global branch tests.



Branches			
Global	Local	Relational	mAP
✓			41.2
	✓		41.8
		✓	42.5
✓	✓		42.0
✓		✓	42.6
	✓	✓	**44.1**
✓	✓	✓	44.0

(f)

Figure 5: **Ablation study.** Classification accuracy variation of SVAG on Charades as a function of (a) supervoxel size, (b) the number of graph convolutional layers B, (c) graph pooling rate η^2, (d) density of similarity graph χ, (e) temporal window length T, and (f) the combination of the used branches.

	Classification Head		Other components	Backbone
	(GFLOPS)	(Mem@{train, test})	(GFLOPS)	(GFLOPS)
Non-Local [65]	107.8	{1830, 680} Mb	–	{R101-NL: 645.9, R50-NL: 452.1, ip-CSN101: 166.1, ip-CSN50: 124.3}
SVAG (Ours)	1.2	{90, 60} Mb	Supervoxel computation: 40.1	

Table 3: **Comparison of required resources for a 10 second clip.** We compare GFLOPS and memory between our method and the Non-Local Network of Wang *et al.* [65]. The input video has a shape of ($64\texttt{f} \times 256\text{p} \times 256\text{p}$). Supervoxel computation includes both the UperNet feature extraction and the supervoxel iteration.

Among the single-branch configurations, the relational branch achieves the highest accuracy (42.5%). Furthermore, the relational branch provides complementary information to that captured by the local branch: when paired together (local, relational) achieve an accuracy of 44.1%. Adding the global branch to the pair of (local, relational) does not provide additional gain. It suggests that the local and the relational branches already contain sufficient global information. From this study we conclude that the global branch is not necessary when both the local and the relational branches are used. Yet, it is an important baseline, so we consider it for our ablation study.

5. Discussion and Conclusion

In this work, we propose to capture long-range and contextual dependencies of video through relational inference over a graph-based model. The nodes of the graph are semantic supervoxels that conform to the spatiotemporal structure of objects and motion cues in the video, while the edges capture spatiotemporal relations and feature similarity between nodes. Using supervoxels as graph nodes meets several desiderata: compactness, coverage, and semantic expressivity. It not only provides the full coverage of the entire video, but also effectively reduces the dimensionality of the convolutional tensors computed by 3D CNNs. The compactness and coverage of supervoxels facilitate efficient semantic relational reasoning over long-range spatiotemporal volumes. Our experiments demonstrate that 16 supervoxels per second are sufficient to provide a rich semantic description of the video, and the proposed hierarchical graph convolution and attentional pooling on this compact representation can capture long-range relations and yield state-of-the-art accuracy on challenging action recognition datasets.

References

[1] J. L. Ba, J. R. Kiros, and G. E. Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[2] Y. Bian, C. Gan, X. Liu, F. Li, X. Long, Y. Li, H. Qi, J. Zhou, S. Wen, and Y. Lin. Revisiting the effectiveness of off-the-shelf temporal modeling approaches for large-scale video classification. *arXiv preprint arXiv:1708.03805*, 2017.

[3] J. Carreira and A. Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[4] Y. Chen, Y. Kalantidis, J. Li, S. Yan, and J. Feng. Aˆ 2-nets: Double attention networks. In *Advances in Neural Information Processing Systems*, 2018.

[5] B. Dai, Y. Zhang, and D. Lin. Detecting visual relationships with deep relational networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[6] N. Dalal, B. Triggs, and C. Schmid. Human detection using oriented histograms of flow and appearance. In *Proceedings of the European Conference on Computer Vision*, 2006.

[7] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[8] C. Feichtenhofer, H. Fan, J. Malik, and K. He. Slowfast networks for video recognition. In *Proceedings of the International Conference on Computer Vision*, 2019.

[9] C. Feichtenhofer, A. Pinz, and R. Wildes. Spatiotemporal residual networks for video action recognition. In *Advances in Neural Information Processing Systems*, 2016.

[10] C. Gan, T. Yao, K. Yang, Y. Yang, and T. Mei. You lead, we exceed: Labor-free video concept learning by jointly exploiting web videos and images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[11] G. Gkioxari, R. Girshick, P. Dollár, and K. He. Detecting and recognizing human-object interactions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[12] R. Goyal, S. E. Kahou, V. Michalski, J. Materzynska, S. Westphal, H. Kim, V. Haenel, I. Fruend, P. Yianilos, M. Mueller-Freitag, et al. The" something something" video database for learning and evaluating visual common sense. In *Proceedings of the International Conference on Computer Vision*, 2017.

[13] J. Gu, H. Zhao, Z. Lin, S. Li, J. Cai, and M. Ling. Scene graph generation with external knowledge and image reconstruction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[14] A. Gupta, A. Kembhavi, and L. S. Davis. Observing human-object interactions: Using spatial and functional compatibility for recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(10):1775–1789, 2009.

[15] M. Hoai and A. Zisserman. Improving human action recognition using score distribution and ranking. In *Proceedings of the Asian Conference on Computer Vision*, 2014.

[16] R. Hu, M. Rohrbach, J. Andreas, T. Darrell, and K. Saenko. Modeling relationships in referential expressions with compositional modular networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[17] N. Hussein, E. Gavves, and A. W. Smeulders. Timeception for complex action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[18] V. Jampani, D. Sun, M.-Y. Liu, M.-H. Yang, and J. Kautz. Superpixel sampling networks. In *Proceedings of the European Conference on Computer Vision*, 2018.

[19] S. Ji, W. Xu, M. Yang, and K. Yu. 3d convolutional neural networks for human action recognition. *IEEE transactions on pattern analysis and machine intelligence*, 35(1):221–231, 2012.

[20] B. Jiang, M. Wang, W. Gan, W. Wu, and J. Yan. Stm: Spatiotemporal and motion encoding for action recognition. In *Proceedings of the International Conference on Computer Vision*, 2019.

[21] J. Johnson, R. Krishna, M. Stark, L.-J. Li, D. Shamma, M. Bernstein, and L. Fei-Fei. Image retrieval using scene graphs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[22] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[23] D.-J. Kim, J. Choi, T.-H. Oh, and I. S. Kweon. Dense relational captioning: Triple-stream networks for relationship-based captioning. In *Proceedings of*

the IEEE Conference on Computer Vision and Pattern Recognition, 2019.

[24] T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations (ICLR)*, 2017.

[25] A. Klaser, M. Marszalek, and C. Schmid. A spatiotemporal descriptor based on 3d-gradients. In *19th British Machine Vision Conference, September 2008*, pages 995–1004, 2008.

[26] M. P. Kumar and D. Koller. Efficiently selecting regions for scene understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2010.

[27] Z. Lan, M. Lin, X. Li, A. G. Hauptmann, and B. Raj. Beyond gaussian pyramid: Multi-skip feature stacking for action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[28] I. Laptev. On space-time interest points. *International journal of computer vision*, 64(2-3):107–123, 2005.

[29] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[30] Q. V. Le, W. Y. Zou, S. Y. Yeung, and A. Y. Ng. Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2011.

[31] J. Lee, I. Lee, and J. Kang. Self-attention graph pooling. In *Proceedings of the International Conference on Machine Learning*, 2019.

[32] M. Lee, S. Lee, S. Son, G. Park, and N. Kwak. Motion feature network: Fixed motion filter for action recognition. In *Proceedings of the European Conference on Computer Vision*, 2018.

[33] F. Li, C. Gan, X. Liu, Y. Bian, X. Long, Y. Li, Z. Li, J. Zhou, and S. Wen. Temporal modeling approaches for large-scale youtube-8m video understanding. *arXiv preprint arXiv:1707.04555*, 2017.

[34] J. Lin, C. Gan, and S. Han. Tsm: Temporal shift module for efficient video understanding. In *Proceedings of the International Conference on Computer Vision*, 2019.

[35] C. Lu, R. Krishna, M. Bernstein, and L. Fei-Fei. Visual relationship detection with language priors. In *Proceedings of the European Conference on Computer Vision*, 2016.

[36] C.-Y. Ma, A. Kadav, I. Melvin, Z. Kira, G. AlRegib, and H. Peter Graf. Attend and interact: Higher-order object interactions for video understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[37] B. Martinez, D. Modolo, Y. Xiong, and J. Tighe. Action recognition with spatial-temporal discriminative filter banks. In *Proceedings of the International Conference on Computer Vision*, 2019.

[38] B. Ni, X. Yang, and S. Gao. Progressively parsing interactional objects for fine grained action detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[39] P. Pan, Z. Xu, Y. Yang, F. Wu, and Y. Zhuang. Hierarchical recurrent neural encoder for video representation with application to captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[40] X. Peng, C. Zou, Y. Qiao, and Q. Peng. Action recognition with stacked fisher vectors. In *Proceedings of the European Conference on Computer Vision*, 2014.

[41] M. Qi, W. Li, Z. Yang, Y. Wang, and J. Luo. Attentive relational networks for mapping images to scene graphs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[42] Z. Qiu, T. Yao, and T. Mei. Learning spatio-temporal representation with pseudo-3d residual networks. In *Proceedings of the International Conference on Computer Vision*, 2017.

[43] M. S. Ryoo, A. Piergiovanni, J. Kangaspunta, and A. Angelova. Assemblenet++: Assembling modality representations via attention connections. In *ECCV*, 2020.

[44] M. S. Ryoo, A. Piergiovanni, T. Mingxing, and A. Angelova. Assemblenet: Searching for multi-stream neural connectivity in video architectures. In *ICLR*, 2020.

[45] S. Sadanand and J. J. Corso. Action bank: A high-level representation of activity in video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

[46] A. Santoro, D. Raposo, D. G. Barrett, M. Malinowski, R. Pascanu, P. Battaglia, and T. Lillicrap. A simple neural network module for relational reasoning. In *Advances in Neural Information Processing Systems*, 2017.

[47] G. A. Sigurdsson, S. Divvala, A. Farhadi, and A. Gupta. Asynchronous temporal fields for action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[48] G. A. Sigurdsson, G. Varol, X. Wang, A. Farhadi, I. Laptev, and A. Gupta. Hollywood in homes: Crowdsourcing data collection for activity understanding. In *Proceedings of the European Conference on Computer Vision*, 2016.

[49] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in Neural Information Processing Systems*, 2014.

[50] N. Srivastava, E. Mansimov, and R. Salakhudinov. Unsupervised learning of video representations using lstms. In *Proceedings of the International Conference on Machine Learning*, 2015.

[51] C. Sun, S. Shetty, R. Sukthankar, and R. Nevatia. Temporal localization of fine-grained actions in videos by domain transfer from web images. In *Proceedings of the 23rd ACM international conference on Multimedia*, pages 371–380. ACM, 2015.

[52] C. Sun, A. Shrivastava, C. Vondrick, R. Sukthankar, K. Murphy, and C. Schmid. Relational action forecasting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[53] G. W. Taylor, R. Fergus, Y. LeCun, and C. Bregler. Convolutional learning of spatio-temporal features. In *Proceedings of the European Conference on Computer Vision*, 2010.

[54] Y. Tian, P. Luo, X. Wang, and X. Tang. Deep learning strong parts for pedestrian detection. In *Proceedings of the International Conference on Computer Vision*, 2015.

[55] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the International Conference on Computer Vision*, 2015.

[56] D. Tran, H. Wang, L. Torresani, and M. Feiszli. Video classification with channel-separated convolutional networks. In *Proceedings of the International Conference on Computer Vision*, 2019.

[57] 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*, 2018.

[58] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, **L**. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, 2017.

[59] H. Wang and C. Schmid. Action recognition with improved trajectories. In *Proceedings of the International Conference on Computer Vision*, 2013.

[60] L. Wang, Y. Qiao, and X. Tang. Action recognition with trajectory-pooled deep-convolutional descriptors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[61] 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 *Proceedings of the European Conference on Computer Vision*, 2016.

[62] 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 *Proceedings of the European Conference on Computer Vision*, 2016.

[63] W. Wang, R. Wang, S. Shan, and X. Chen. Exploring context and visual pattern of relationship for scene graph generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[64] X. Wang, A. Farhadi, and A. Gupta. Actions˜ transformations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[65] X. Wang, R. Girshick, A. Gupta, and K. He. Non-local neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[66] X. Wang and A. Gupta. Videos as space-time region graphs. In *Proceedings of the European Conference on Computer Vision*, 2018.

[67] Y. Wang and M. Hoai. Pulling actions out of context: Explicit separation for effective combination. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[68] Y. Wang and G. Mori. Hidden part models for human action recognition: Probabilistic versus max margin. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(7):1310–1323, 2011.

[69] Y. Wang, V. Tran, G. Bertasius, L. Torresani, and M. Hoai. Attentive action and context factorization. In *Proceedings of the British Machine Vision Conference*, 2020.

[70] Y. Wang, V. Tran, and M. Hoai. Eigen-evolution dense trajectory descriptors. In *Proceedings of the International Conference on Automatic Face and Gesture Recognition*, 2018.

[71] C.-Y. Wu, C. Feichtenhofer, H. Fan, K. He, P. Krähenbühl, and R. Girshick. Long-term feature banks for detailed video understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[72] C.-Y. Wu, M. Zaheer, H. Hu, R. Manmatha, A. J. Smola, and P. Krähenbühl. Compressed video action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[73] Z. Wu, X. Wang, Y.-G. Jiang, H. Ye, and X. Xue. Modeling spatial-temporal clues in a hybrid deep learning framework for video classification. In *Proceedings of the 23rd ACM international conference on Multimedia*, pages 461–470. ACM, 2015.

[74] T. Xiao, Y. Liu, B. Zhou, Y. Jiang, and J. Sun. Unified perceptual parsing for scene understanding. In *Proceedings of the European Conference on Computer Vision*, 2018.

[75] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[76] S. Xie, C. Sun, J. Huang, Z. Tu, and K. Murphy. Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In *Proceedings of the European Conference on Computer Vision*, 2018.

[77] J. Yang, J. Lu, S. Lee, D. Batra, and D. Parikh. Graph r-cnn for scene graph generation. In *Proceedings of the European Conference on Computer Vision*, 2018.

[78] B. Yao and L. Fei-Fei. Modeling mutual context of object and human pose in human-object interaction activities. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2010.

[79] J. Yao, S. Fidler, and R. Urtasun. Describing the scene as a whole: Joint object detection, scene classification and semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

[80] T. Yao, Y. Pan, Y. Li, and T. Mei. Exploring visual relationship for image captioning. In *Proceedings of the European Conference on Computer Vision*, 2018.

[81] C.-P. Yu, H. Le, G. Zelinsky, and D. Samaras. Efficient video segmentation using parametric graph partitioning. In *Proceedings of the International Conference on Computer Vision*, 2015.

[82] J. Yue-Hei Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici. Beyond short snippets: Deep networks for video classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[83] J. Zhang, M. Elhoseiny, S. Cohen, W. Chang, and A. Elgammal. Relationship proposal networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[84] B. Zhou, A. Andonian, A. Oliva, and A. Torralba. Temporal relational reasoning in videos. In *Proceedings of the European Conference on Computer Vision*, 2018.

[85] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba. Scene parsing through ade20k dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[86] J. Zhu, B. Wang, X. Yang, W. Zhang, and Z. Tu. Action recognition with actons. In *Proceedings of the International Conference on Computer Vision*, 2013.

[87] B. Zhuang, L. Liu, C. Shen, and I. Reid. Towards context-aware interaction recognition for visual relationship detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[88] M. Zolfaghari, K. Singh, and T. Brox. Eco: Efficient convolutional network for online video understanding. In *Proceedings of the European Conference on Computer Vision*, 2018.