Representation Learning on Visual-Symbolic Graphs for Video Understanding

Effrosyni Mavroudi$^{1[0000-0001-7552-8342]}$, Benjamín Béjar Haro$^{1[0000-0001-9705-4483]}$, and René Vidal^{1}

Mathematical Institute for Data Science, Johns Hopkins University, Baltimore, MD
{emavrou1,bbejar,rvidal}@jhu.edu

Abstract. Events in natural videos typically arise from spatio-temporal interactions between actors and objects and involve multiple co-occurring activities and object classes. To capture this rich visual and semantic context, we propose using two graphs: (1) an attributed spatio-temporal visual graph whose nodes correspond to actors and objects and whose edges encode different types of interactions, and (2) a symbolic graph that models semantic relationships. We further propose a graph neural network for refining the representations of actors, objects and their interactions on the resulting hybrid graph. Our framework goes beyond current approaches that assume nodes and edges of the same type, operate on a fixed graph structure and do not use a symbolic graph. In particular, our framework: a) has specialized attention-based aggregation functions for different node and edge types; b) uses visual edge features; c) integrates visual evidence with label relationships; and d) performs global reasoning in the semantic space. Experiments on challenging video understanding tasks, such as temporal action localization on the Charades dataset, show that the proposed method leads to state-of-the-art performance.

1 Introduction

The field of video understanding has been moving towards increasing levels of complexity, from classifying a single action in short trimmed videos to detecting multiple complex activities performed by multiple actors interacting with objects in untrimmed videos. Therefore, there is a need to develop algorithms that can effectively model spatio-temporal visual and semantic context. One way of capturing such context is to use graph-based modeling, which has a rich history in computer vision. Traditional graph-based approaches, e.g., using probabilistic graphical models [26,27,68,58], focused mainly on modeling context at the level of symbols rather than signals/visual representations. However, recent advances have enabled *representation learning on graph-structured data* using deep architectures called Graph Neural Networks (GNNs), which learn how to iteratively update node representations by aggregating messages from their neighbors [25].

Videos can be represented as visual spatio-temporal attributed graphs (visual st-graphs) whose nodes correspond to regions obtained by an object detector and whose edges capture interactions between such regions. GNNs have recently

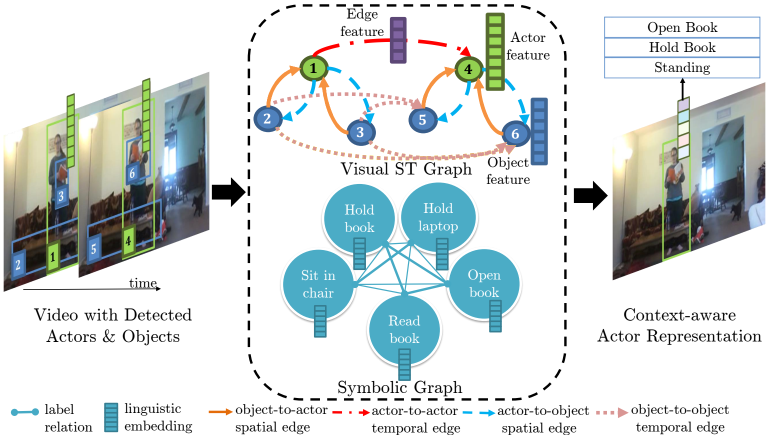

Fig. 1: **Cues for video understanding:** (1) *visual spatio-temporal interactions* between actors and objects and (2) commonsense *relationships between labels*, such as co-occurrences. These cues can be encoded in a hybrid spatio-temporal visual and symbolic attributed graph. In this work, we perform representation learning on this hybrid graph to obtain context-aware representations of detected semantic entities, such as actors and objects, that can be used to solve downstream video understanding tasks, such as multi-label action recognition.

been designed for refining the local node/edge features, typically extracted by a convolutional neural network, based on the spatio-temporal context captured by the graph. Although representation learning on visual st-graphs has lead to significant advances in video understanding [63,52,14,57,19,62,3], there are four key limitations of state-of-the-art approaches that prevent them from fully exploiting the rich structure of these graphs. First, most GNNs assume nodes/edges of the same type. In practice, the visual st-graph is a *heterogeneous* graph that has distinct node types (actor, object) and edge types (such as *object-to-actor spatial* and *actor-to-actor temporal*), with each type being associated with a feature of potentially different dimensionality and semantics, as shown in the example of Fig. 1. Because of this limitation, recent attempts at explicitly modeling actors and objects have resorted to applying separate GNNs for each node/edge type [63,12]. Second, most methods operate on a graph of fixed structure with dense connectivity between detected regions. In practice, only a few of the edges capture meaningful interactions. Third, current approaches do not incorporate edge features for updating the node representations. Finally, despite modeling local visual context, existing approaches do not reason at a global video level or exploit commonsense semantic label relationships, which have been shown to be beneficial in the image recognition domain [33,5].

In this work, in an effort to address these limitations, we propose a novel Graph Neural Network (GNN) model, called Visual Symbolic - Spatio Temporal - Message Passing Neural Network (VS-ST-MPNN), that performs representation learning on visual st-graphs to obtain context-aware representations of detected actors and objects (Fig. 1). Our model employs *learnable neighborhood aggregation mechanisms*, specialized for each node and edge type, to iteratively refine the actor and object representations. We also adapt the graph connectivity with an *attention mechanism* specialized for each type of interaction. For example, an actor node will separately attend to actor nodes at the previous frame and object nodes at the current frame. Furthermore, we initialize edge features with *geometric relations between regions*, refine them and use them to adapt the graph connectivity. Intuitively, nodes which are close to each other or are interacting should be strongly connected. Finally, one of our key contributions is incorporating an attributed *symbolic graph* whose nodes correspond to semantic labels, such as actions, described by word embeddings and whose edges capture label relationships, such as co-occurrence. We fuse the information of the two graphs with learnable association weights between their nodes and perform *global semantic reasoning* on the symbolic graph. Importantly, we do not require ground truth annotations of objects, tracks or semantic labels for each visual node.

In summary, the contributions of this work are three-fold. First, we model contextual cues for video understanding by combining a symbolic graph, capturing semantic label relationships, with a visual st-graph, encoding interactions between detected actors and objects. Second, we introduce a novel GNN that can perform joint representation learning on the hybrid visual-symbolic graph, in order to obtain visual and semantic context-aware representations of actors, objects and their interactions in a video, which can then be used to solve downstream recognition tasks. Finally, to demonstrate the effectiveness and generality of our method, we evaluate it on tasks such as multi-label temporal activity localization, object affordance detection and grounded video description on three challenging datasets and show that it achieves state-of-the-art performance.

2 Related work

Context and its role in vision has been studied for a long time [42]. There are two major, complementary types of context utilized in video understanding tasks: (a) *coarse-grained visual context* is captured by applying convolutional neural networks to short sequences of whole frames [51,54,56,4,65] followed by temporal models, such as recurrent neural networks [29,43,65] and (b) *fine-grained visual context* is captured by using mid-level representations of semantic parts, such as body parts [7,39], latent attributes [35], secondary regions [15], human-object interactions [45,67] and object-object interactions [36,3,65].

Graph neural networks for video understanding. The first approach applying a deep network on a visual graph for video understanding was the Structured Inference Machine [11], which introduced actor feature refinement with message passing, and trainable gating functions for filtering out spurious interactions, but only captured spatial relationships between actors. Another early approach was the S-RNN [21], which although related because it introduced the concept of weight-sharing between nodes or edges of the same type, it did not iteratively refine node representations. With the advent of GNNs, many researchers have explored using them for video understanding, by modeling whole frames [64], tracklets [63], feature map columns [52,14,41] or object proposals [57,19,62] as graph nodes and using off-the-shelf GNNs, such as MPNNs [13], GCNs [25] and Relation Networks [20,52,64,3] to refine the node or edge representations, obtaining significant performance gains. However, most of these GNNs are unable to handle edge features, directed edges, distinct node and edge types, and unknown underlying graph structure. Therefore, applying existing GNNs to visual st-graphs requires treating every node and edge in the same way [3,14], or focusing only on one edge type [52,64,36,19,20], or using separate GNNs for each type of interaction [57,63,12], hence completely ignoring or sub-optimally handling their rich graph structure. In contrast, our proposed method can be directly applied to any st-graph and supports message passing in heterogeneous graphs. The benefit of such fine-grained modeling has already been established in fields such as computational pharmacology and relational databases [69,16,48], but remains relatively unexplored in computer vision. Furthermore, similar to [46,14], our method utilizes an attention mechanism for adapting the visual edge weights over iterations, but in our case this mechanism is specialized for different edge types and takes advantage of edge features.

Symbolic graphs. There is a long line of work on exploiting external knowledge encoded in label relation graphs for visual recognition tasks. Semantic label hierarchies, such as co-occurrence, have been leveraged for improving object recognition [38,37,8,10], multi-label zero-shot learning [30] and other image-based visual tasks [31,47]. Much fewer papers utilize knowledge graphs for video understanding [1,23,24], possibly due to the limited number of semantic classes in traditional video datasets. A notable exception is the SINN [24], which performs graph-based inference in a hierarchical label space for action recognition. However, most of these methods directly perform inference on the symbolic graph. Rather, we aim to use the semantics of labels to integrate prior knowledge about the inter-class relationships as well as linguistic information, and facilitate semantic space reasoning. In a similar vein, Liang et al. [33] enhance convolutional feature maps in the coordinate space by using a symbolic graph, while [32,6] use a latent interaction graph. In contrast, we seek to improve the representation of visual st-graph nodes rather than enhance features in a grid structure. Fusing information from multiple graphs using GNNs is an exciting new line of research [2,59,53]. Similar to our approach, Chen et al. [5] combine a visual graph instantiated on objects with a symbolic graph and perform graph representation learning, while [22] enforce the scalar edge weights between visual regions to be consistent with the edges of the symbolic graph. However, both of them operate on simple spatial graphs and assume access to semantic labels of regions during training.

3 Method

In this section, we describe the overall architecture of our proposed VS-ST-MPNN model, shown in Fig. 2. Our goal is to refine the features of detected actors,

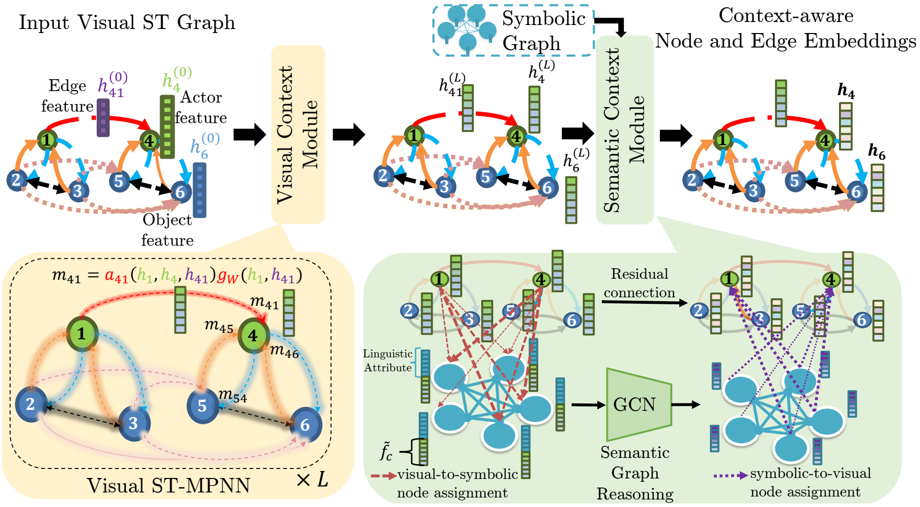

Fig. 2: **Overview of our VS-ST-MPNN model that performs representation learning on a hybrid visual-symbolic graph**. Given an input video that is represented as a visual st-graph, with nodes corresponding to detected actors and objects and edges capturing latent interactions, our framework has two modules that integrate context in the local representations of its nodes and edges: (a) a Visual Context Module (Sec. 3.1) that performs L rounds of node and edge updates on the visual graph, with specialized neighborhood aggregation functions that depend on the type of an edge and (b) a Semantic Context Module (Sec. 3.2) that integrates visual evidence with semantic knowledge encoded in an external symbolic graph and learns global semantic interaction-aware features.

objects and their interactions based on the contextual information captured in two graphs: a visual st-graph and a symbolic graph. The refinement is performed with a novel GNN, which a) is designed to exploit the rich structure of the visual st-graph by utilizing edge features and learning specialized attention-based neighborhood aggregation functions for different node and edge types, and b) enables the fusion with the symbolic graph, by incorporating a *semantic reasoning module* that learns semantic relation-aware features and a *soft-assignment module* that connects visual and symbolic graph nodes without requiring access to ground-truth semantic labels of regions during training. The context-aware features can then be used in downstream video understanding tasks.

3.1 Visual Context Module

Visual st-graph. Our input is a sequence of T frames with detected actor and object regions. Let $G^v = (V^v, E^v)$ be a spatio-temporal attributed directed graph, called the *visual st-graph*, where V^v is a finite set of vertices and $E^v \subseteq V^v \times V^v$

is a set of edges. Nodes correspond to actor and object detections, while edges model latent interactions. There are M actors and N objects per frame. Fig. 2 illustrates a toy example with $M = 1$, $N = 2$ and $T = 2$.

The graph is both node- and edge-typed with $\mathcal{N}$ node types and $\mathcal{E}$ edge types. For example, the node types are *actor* and *object* ($\mathcal{N} = 2$) and the edge types can be: object-to-actor spatial (*obj-act-s*), actor-to-object spatial (*act-obj-s*), actor-to-actor temporal (*act-act-t*) and object-to-object temporal (*obj-obj-t*) ($\mathcal{E} = 5$). Each node and edge is associated with an initial attribute vector, whose dimensionality may vary depending on the node/edge type. An actor/object appearance feature can be used as the initial attribute of node i ($\mathbf{h}_i^{(0)}$), while the relative spatial location of regions i and j can be used as the initial attribute of the edge from j to i ($\mathbf{h}_{ij}^{(0)}$). The allowed spatio-temporal connections between nodes of the visual st-graph are specified a-priori by a binary adjacency matrix $L^v \in \{0,1\}^{|V^v|\times|V^v|}$. For instance, we can constrain temporal edges to connect a node at frame t with another node of the same type at time $t-1$. L^v defines the neighborhood of each node and, thus, encodes the family of spatio-temporal interactions captured by the model.

Visual ST-MPNN. Given the input visual st-graph G^v with initial node and edge attributes/features, $\{\mathbf{h}_i^{(0)}\}_{i\in V^v}$ and $\{\mathbf{h}_{ij}^{(0)}\}_{(i,j)\in E^v}$, respectively, we introduce novel GNN propagation rules to perform representation learning on the visual st-graph with the goal of refining local node and edge attributes using spatio-temporal contextual cues. In each iteration of node and edge refinement our model: (1) adapts the connectivity of the visual st-graph by refining scalar edge weights using attention coefficients; (2) computes a message along each edge that depends on the edge type, the attention-based scalar edge weight, the attributes of the connected nodes and the edge attribute; (3) updates the attribute of every node by aggregating messages from incoming edges; and (4) updates the attribute of every edge by using the message that was computed alongside it. Next, we describe each one of these steps in more detail.

– *Adapting graph connectivity:* At each iteration l of the MPNN, we first refine the graph connectivity by computing *attention coefficients*, a_{ij} that capture the relevance of node j for the update of node i. In particular, in contrast to GAT [55], our model learns an attention mechanism specialized for each type of interaction and it utilizes edge features for its computation. The attention coefficients for the l-th iteration are computed as follows:

$$a_{ij}^{(l)} = \exp\left(\gamma_{ij}^{(l)}\right) / \left(\sum_{k\in N_\epsilon(i)} \exp\left(\gamma_{ik}^{(l)}\right)\right), \tag{1}$$

$$\gamma_{ij}^{(l)} = \rho\left((\mathbf{v}_a^\epsilon)^T \left[W_r^{\nu_r}\mathbf{h}_i^{(l-1)}; W_s^{\nu_s}\mathbf{h}_j^{(l-1)}; \lambda_{ea} W_{rs}^\epsilon \mathbf{h}_{ij}^{(l-1)}\right]\right). \tag{2}$$

Here, ϵ is the type of the edge from j to i, $N_\epsilon(i)$ denotes the set of nodes connected with i via an incoming edge of type ϵ, $\mathbf{h}_{ij}^{(l-1)}$ is the previous state of the edge from j to i, ν_r is the node type of the receiver node i, ν_s is the node type of the sender node j, ρ is a non-linearity, such as Leaky-ReLU [18]. The λ_{ea} is binary scalar hyperparameter denotes whether edge features will be used for computing the attention coefficients. $W_r^{\nu_r}$, $W_s^{\nu_s}$ and W_{rs}^{ϵ} are learnable projection weights and are shared between nodes (edges) of the same type. All projection matrices linearly transform the current node (edge) attribute to a refined feature of fixed dimensionality d_l. $\mathbf{v}_a^{\epsilon}$ is a learnable attention vector. For improved readability we have dropped the layer index (l) from the attention and projection weights.

- *Message computation:* After computing the attention coefficients, we compute a message along each edge. The message from node j to node i is:

$$\mathbf{m}_{ij}^{(l)} = a_{ij}^{(l)} \left(\lambda_v W_s^{\nu_s} \mathbf{h}_j^{(l-1)} + \lambda_e W_{rs}^{\epsilon} \mathbf{h}_{ij}^{(l-1)} \right), \tag{3}$$

where λ_e is a binary scalar hyperparameter, denoting whether edge features will used in the messages, λ_v is a binary scalar hyperparameter, denoting whether the sending node feature will be used in the message and the learnable weight matrices are the same as the ones used in the attention computation.

- *Node and edge update:* Following the message computation, the node attribute is updated using an aggregation of incoming messages from different edge types and a residual connection, while the edge attribute is set to be equal to the message:

$$\mathbf{h}_i^{(l)} = \mathbf{h}_i^{(l-1)} + \sigma \left(\sum_{\epsilon=0}^{\mathcal{E}-1} \sum_{j \in N_\epsilon(i)} \mathbf{m}_{ij}^{(l)} \right), \mathbf{h}_{ij}^{(l)} = \mathbf{m}_{ij}^{(l)},$$

where $\sigma(\cdot)$ is a non-linearity, such as ReLU. After L layers of the spatio-temporal MPNN (or equivalently L rounds of node and edge updates), we obtain refined, visual context-aware node and edge attributes: $\mathbf{h}_i^{(L)} \in \mathbb{R}^{d_L}$, $\mathbf{h}_{ij}^{(L)} \in \mathbb{R}^{d_L}$.

3.2 Semantic Context Module

Symbolic graph. Let $G^s = (V^s, E^s)$, be the input *symbolic* graph, where V^s and E^s denote the symbol set and edge set, respectively. The nodes of this graph correspond to semantic labels, such as action labels or object labels. Each symbolic node c is associated with a semantic attribute, such as the linguistic embedding of the label ($\mathbf{s}_c \in \mathbb{R}^K$). Edges in the symbolic graph are associated with scalar weights, which encode label relationships, such as co-occurrence. These edge weights are summarized in the fixed adjacency matrix $L^s \in \mathbb{R}^{|V^s| \times |V^s|}$.

- *Integration of visual evidence with the symbolic graph:* As a first step, we update the attributes of the symbolic graph using visual evidence, i.e., the visual context-aware representations of the nodes of the visual st-graph. To achieve this, without requiring access to the ground-truth semantic labels of regions, we learn associations between the nodes of the visual and of the symbolic graph. The association weight $\phi_{c,i}^{vs}$ represents the confidence of assigning the feature from visual node i to the symbolic node c:

$$\phi_{c,i}^{vs} = \frac{\omega_{c,i} \exp\left(\left(\mathbf{w}_c^{vs}\right)^T \mathbf{h}_i^{(L)} \right)}{\sum_{c' \in V^s} \omega_{c',i} \exp\left(\left(\mathbf{w}_{c'}^{vs}\right)^T \mathbf{h}_i^{(L)} \right)}, \tag{4}$$

where $\mathbf{w}_c^{vs} \in \mathbb{R}^{d_L}$ is a trainable weight vector and $\Omega \in \{0,1\}^{|V_s|\times|V_v|}$ is an input binary mask that defines allowed visual-to-symbolic node connections. For example, when our symbolic nodes correspond to action classes, we can disable connections between object and symbolic nodes.

After computing the voting weights, each symbolic node is associated with a weighted sum of projected visual node features: $\tilde{\mathbf{f}}_c = \sigma(\sum_i \phi_{c,i}^{vs} W_p^{vs} \mathbf{h}_i^{(L)})$, where $W_p^{vs} \in \mathbb{R}^{D_s \times d_L}$ is a learnable projection weight matrix. The new representation of each symbolic graph node c is computed as the concatenation of the linguistic embedding, $\mathbf{s}_c$, and the visual feature, $\tilde{\mathbf{f}}_c$, $\mathbf{s}_c^{(0)} = \left[\mathbf{s}_c; \tilde{\mathbf{f}}_c\right] \in \mathbb{R}^{K+D_s}$.

– *Semantic Graph Reasoning:* We learn semantic relation-aware features by applying a vanilla GCN [25] on the nodes of the symbolic graph. The GCN yields evolved symbolic node features $S^{(R)} \in \mathbb{R}^{|V^s|\times D_s}$, by iteratively applying the propagation rule: $S^{(r+1)} = \mathrm{GCN}(S^{(r)}, L^s)$, where $S^{(r)}$ denotes the matrix of symbolic node embeddings at the r-th iteration.

– *Update of visual st-graph:* The evolved symbolic node representations obtained after R iterations of graph convolutions on the symbolic graph can be mapped back to the visual st-graph, so that the representation of the visual nodes can be enriched by global semantic context. To achieve this we compute mapping weights (attention coefficients) from symbolic nodes to visual nodes:

$$\phi_{i,c}^{sv} = \frac{\omega_{c,i} \exp\left(e_{i,c}^{sv}\right)}{\sum_{c' \in V^s} \omega_{c',i} \exp\left(e_{i,c'}^{sv}\right)}, \tag{5}$$

where $e_{i,c}^{sv} = (\mathbf{v}_a^{sv})^T \left[\mathbf{s}_c^{(R)}; \mathbf{h}_i^{(L)}\right]$ and $\mathbf{v}_a^{sv} \in \mathbb{R}^{d_L+D_s}$ is a learnable attention vector. The final visual node feature representation is then obtained using a residual connection: $\mathbf{h}_i = \mathbf{h}_i^{(L)} + \sigma\left(\sum_{c' \in V^s} \phi_{i,c'}^{sv} W_p^{sv} \mathbf{s}_{c'}^{(R)}\right)$. These context-aware representations can be used for downstream video understanding tasks.

4 Experiments

To demonstrate the effectiveness and generality of our method, we conduct experiments on three challenging video understanding tasks that require reasoning about interactions between semantic entities and relationships between classes: a) sub-activity and object affordance classification (Sec. 4.1), b) multi-label temporal action localization (Sec. 4.2) and c) grounded video description (Sec. 4.3).

CAD-120. This dataset provides 120 RGB-D video sequences, with each video showing a daily activity comprised of a sequence of sub-activities (e.g., *moving*, *drinking*) and object affordances (e.g., *reachable*, *drinkable*) [27]. Given temporal segments, the task is to classify each actor in each segment into one of 10 sub-activity classes and each object into one of 12 affordance classes. Evaluation is performed with 4-fold, leave-one-subject-out, cross-validation using F1-scores averaged over all classes as an evaluation metric. With a visual st-graph provided

Table 1: **Results on CAD-120 [27]** for sub-activity and object affordance detection, measured via F1-score. Our results are *averaged across 5 runs*.

Method	Detection F1-score (%)	
	Sub-activity	Object affordance
ATCRF [27]	80.4	81.5
S-RNN [21]	83.2	88.7
S-RNN [21] (multitask)	82.4	**91.1**
GPNN [46]	88.9	88.8
STGCN [12]	88.5	-
VS-ST-MPNN (ours)	**90.4** (±0.8)	*89.2* (±0.3)
only visual graph (ours)	89.6 (±1.1)	88.6 (±0.6)

by the dataset (including hand-crafted features of actors and objects and geometric relations), it is a particularly good test-bed for comparing different GNNs.

Charades. Charades [50] is a large dataset with 9848 RGB videos and annotations for 157 action classes, many of which involve human-object interactions. Each video contains an average of 6.8 activity instances, many of which are co-occurring. Following [50], multi-label action temporal localization performance is measured in terms of mean Average Precision (mAP), evaluating per-frame predictions for 25 equidistant frames in each one of the 1.8k validation videos.

ActivityNet Entities. The task in the recently released ActivityNet Entities [65] dataset, containing 15k videos and more than 158k annotated bounding boxes, is to generate a sentence describing the event in a ground-truth video segment, and to spatially localize all the generated nouns that belong to a vocabulary of 432 object classes. Following Zhou et al. [65], the quality of generated captions is measured using standard metrics, such as Bleu (B@1, B@4), METEOR (M), CIDEr (C), and SPICE (S), whereas the quality of object localization is evaluated on generated sentences using the $F1_{all}$, $F1_{loc}$ metrics. Object localization results on the test set were obtained using the evaluation server [1].

4.1 Experiments on CAD-120

Implementation details. We use the visual st-graph provided with the dataset, which is instantiated on the actor and objects of each temporal segment of the input video and contains 5 edge types: *obj-obj-s*, *obj-act-s*, *act-obj-s*, *act-act-t* and *obj-obj-t*. We construct a symbolic graph that has nodes corresponding to the 10 sub-activity and 12 affordance classes, with edge weights capturing per-frame class co-occurrences in training data. The attribute of each symbolic node is obtained by using off-the-shelf word2vec [40] class embeddings of size $K = 300$. Actor (object) nodes are connected to sub-activity (affordance) symbolic nodes (see suppl. for details). The following hyperparameters are used in the VS-ST-MPNN model: $L = 4$ graph update rounds, $R = 1$ GCN layer and

[1] `https://competitions.codalab.org/competitions/20537`

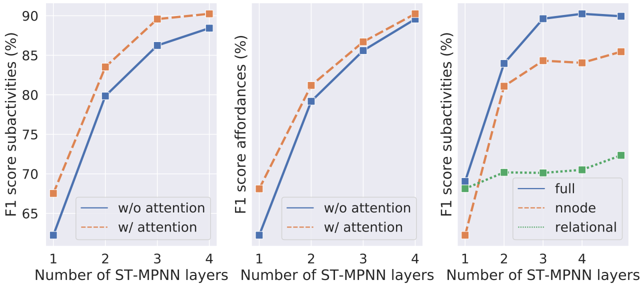

Fig. 3: **Effect of adaptive graph connectivity and node update type on CAD-120 detection performance**. Using an attention mechanism to adapt the graph connectivity outperforms using a fixed visual adjacency matrix. Updating nodes based on both neighboring node and incoming edge attributes (*full*) is superior to updating them using just the nodes (*nnode*) or edges (*relational*).

messages of size 256. Attention and nodes are updated based on node and edge attributes ($\lambda_v = 1, \lambda_e = 1, \lambda_{ea} = 1$). We train our model using the sum of cross-entropy losses computed at each node of the st-graph for 100 epochs, with a batch size of 5 sequences. We use the Adam learning rate scheduler with an initial learning rate of 0.001. Dropout is applied with a rate of 0.5 on all fully connected layers.

Comparison with the state of the art. Table 1 compares the subactivity and affordance detection performance of our method with prior work. Our method obtains state-of-the-art results for sub-activity detection, with an average performance of **90.4**% and a best of **91.3**%, and the second best result on affordance detection (89.2%) - being only second to the S-RNN (multi-task) [21]. The S-RNN was trained on the joint task of detection and anticipation and we outperform it by 8% in the subactivity classification task. Even without using the symbolic graph, our method improves upon recent GNNs, which were applied on the same attributed visual st-graph, validating our novel layer propagation rules.

Ablation analysis. In Fig. 3, we show the effect of attention, edge features and number of visual node updates on the recognition performance. First, we compare the performance of a model trained with a fixed binary adjacency matrix with a model trained using attention. Clearly, adaptive graph connectivity benefits performance in both tasks. Second, we conclude that using the attributes of both the neighboring nodes and adjacent edges is better than using only those of the neighboring nodes, validating the usefulness of edge features. We also observe that increasing the number of ST-MPNN layers improves performance, which saturates after 4-5 layers.

4.2 Experiments on Charades

Implementation details. To tackle the challenging problem of multi-label temporal action localization, we perform a late fusion of a global model, operating

Table 2: **Multi-label temporal action localization results on Charades [50]**. Performance is measured via per-frame mAP. R: RGB, F: optical flow. Our method yields a relative improvement of 6% over the state-of-the-art method by using **only raw RGB frames**.

Method	Feat	Input	mAP (%)
Predictive-corrective [9]	VGG	R	8.9
Two-stream [49]	VGG	R+F	8.94
Two-stream + LSTM [49]	VGG	R+F	9.6
R-C3D [60]	VGG	R+F	12.7
ATF [49]	VGG	R+F	12.8
RGB I3D [43]	I3D	R	15.63
I3D [43]	I3D	R+F	17.22
I3D + LSTM [43]	I3D	R+F	18.12
RGB I3D + super-events [43]	I3D	R	18.64
I3D + super-events [43]	I3D	R+F	19.41
STGCN [12]	I3D	R+F	19.09
I3D + 3TGMs + super-events [44]	I3D	R+F	22.3
I3D + biGRU + VS-ST-MPNN (Ours)	I3D	**R**	**23.7**(±0.2)

on whole frames, and a local model, operating on actors and objects. The global model is an I3D RGB model [4] fine-tuned on Charades [43], combined with a two-layer biGRU of size 256, similar to existing baselines on this dataset. The proposed VS-ST-MPNN is used as the local model. To build the visual st-graph we detect actors and objects using a Faster-RCNN [17] trained on the MS-COCO [34] dataset. We rank detections based on their score and we keep the top-2 human detections and top-10 object detections per frame. Zero-padding is applied to handle frames with fewer actors and objects. We pool features from the `Mixed_4f` 3D feature map of the I3D for each detected region using RoIAlign [17] and max-pooling in space. This yields an attribute of size 832 for the actor/object regions for the frames of the original video sampled at 1.5 FPS. We use 3 types of edges: *obj-act-s*, *act-obj-s* and *act-act-t* and describe each edge with the relative position of the connected regions. Our symbolic graph has nodes corresponding to the 157 action classes and edge weights corresponding to per-frame label co-occurrences in training data. Obtaining a linguistic attribute for each symbolic node in Charades is not trivial, since action names often contain multiple words. To circumvent that, each action class is separated into a verb and an object and the average of these two word embeddings is used as the initial node attribute. The hyperparameters are: $L = 3$, $d_L = 512$, $R = 1$, $D_s = 256$, $\lambda_v = 1, \lambda_e = 1$ and $\lambda_{ea} = 1$. For performing per-frame multi-label action classification, we average the learned actor representations at each frame, we input them to a two-layer biGRU of size 256, and we feed the resulting hidden states to binary action classifiers. We train for 40 epochs with a binary cross-entropy loss applied per frame, using a batch size of 16 sequences. We also apply dropout with a rate of 0.5 on all fully connected layers and use the Adam scheduler, with an initial learning rate of $1e^{-4}$.

Table 3: **Ablation analysis on the Charades [50] dataset**. *Visual*: Visual Context Module. *Semantic*: Semantic Context Module. *Long Term*: long-term temporal modeling (biGRU).

ID	Visual	Semantic	Long Term	mAP (%)	mAP (%)
					+ Global model
1	✓	✓	✓	**18.6**	**23.4**
2	-	-	✓	15.2	22.2
3	✓	✓	-	15.3	22.0
4	✓	-	-	13.7	21.8
5	-	✓	-	11.7	21.8
6	-	-	-	10.7	20.9

Comparison with prior work. As shown in Table 2, our framework outperforms all other methods on temporal action localization, with a mAP of **23.7**%, using only raw RGB frames. It yields a relative improvement of 24% over the alternative graph-based approach [12], which is using both RGB and optical flow inputs, as well as additional actor embeddings trained at the ImSitu dataset [61].
Impact of each graph. In Table 3, we report the baseline result (10.7%) obtained by classifying activities based on local actor features (ID: 6). Refining these features by using our Visual Context Module improves performance by 3%. As shown quantitatively in the supplementary material, both our specialized attention mechanism and the usage of edge features improve the performance, outperforming a vanilla GNN. Representation learning on the hybrid graph yields a significant absolute improvement of **5% over the baseline**. Additionally, modeling long-term temporal context and global context leads to the final state-of-the-art performance, indicating that the representations learned by our model are complementary to holistic scene cues and temporal dynamics.
Per-class improvement analysis. To gain a better understanding of the benefits of representation learning on the visual graph, we highlight in Fig. 4 the activity classes with the highest positive and negative difference in performance when adding *obj-to-act-s* messages. By harnessing visual human-object interaction cues, our model is able to better recognize actions such as *Watching television.*
Impact of semantic graph reasoning. Comparing the models with IDs 3 and 4 in Table 3, we observe that adding the semantic context module improves mAP by 2%. Notably, updating the visual nodes by attending over the initial symbolic node features (linguistic) instead of the evolved features did not improve performance in our experiments, showing the importance of semantic graph reasoning. The semantic module seems to particularly help with rare classes, such as *Holding a vacuum*, which has only 213 training examples (3% of available annotated segments), and classes with strong co-occurrences (Fig. 5). The t-SNE visualization shows that, although the visual context-aware actor embeddings are already capturing meaningful label relationships (e.g., *open* and *hold book*), the integration of semantic relationships via the symbolic graph results in more tightly clustered embeddings and well-defined groups, facilitating action recognition.
Model complexity. Since our visual st-graph is designed to capture only local spatio-temporal interactions, we can compute messages in parallel and process

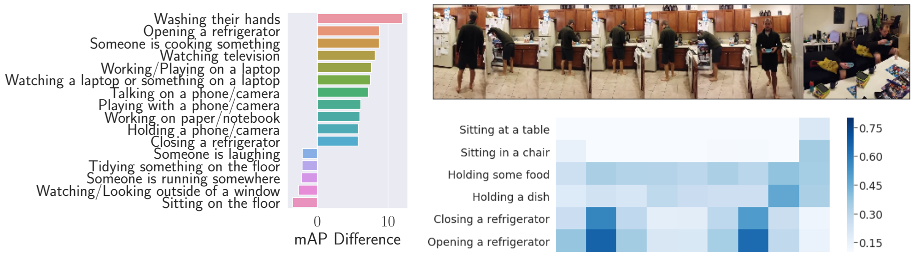

Fig. 4: **Qualitative results on Charades**. (*left*) The classes with the highest positive and negative performance difference after adding *object-to-actor spatial* messages. Incorporating spatial structure benefits actions that involve interactions with distant objects, such as *watching television* or *cooking*. (*right*) Action predictions of our model (ID: 3) for 9 frames of a sample Charades video.

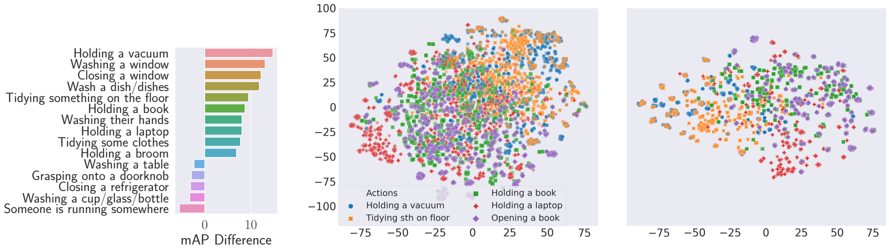

Fig. 5: **Qualitative evaluation of the Semantic Context Module**. (*left*) Classes with the highest positive and negative performance difference when adding the semantic module. (*right*) t-SNE visualization of actor node embeddings from Charades validation set obtained before and after adding the SCM. We show 1121 random samples per class for 5 selected action classes. (*Best viewed zoomed in and in color.*)

the entire Charades validation set (around 2K videos at 1.5FPS) in 2 minutes on a single Titan XP GPU, given initial features pooled from actor/object regions. For additional implementation details, ablations, qualitative results, failure cases and limitations we refer the reader to the the supplementary material.

4.3 Experiments on ActivityNet Entities

The current state-of-the-art grounded video description model (GVD) [65] uses a hierarchical LSTM decoder that generates a sentence describing a video segment, given global video features as well as local region features of 100 region proposals from 10 equidistant frames. The region features are refined using a *multi-head self-attention* (MHA) mechanism. To validate the effectiveness of our model, we experiment with three variants of the GVD: (a) replace the MHA with our VS-ST-MPNN; (b) use the MHA along with our Semantic Context Module; (c)

Table 4: **Grounded video description results on ActivityNet Entities [65]**. MHA: multi-head self-attention. SCM-VG: our semantic context module with visual-to-symbolic node correspondences pre-trained on Visual Genome.

	$B@1$	$B@4$	M	C	S	$F1_{all}$	$F1_{loc}$
Validation set							
GVD (MHA) [65]	**23.9**	2.59	11.2	47.5	15.1	7.11	24.1
GVD (VCM + SCM) (ours)	23.4	2.41	11.1	47.3	14.8	7.28	25.2
GVD (MHA + SCM) (ours)	23.8	2.67	**11.3**	48.6	**15.2**	**7.35**	**25.3**
GVD (MHA + SCM-VG) (ours)	**23.9**	**2.78**	**11.3**	**49.1**	15.1	7.15	24.0
Test set							
Masked Transformer [66]	22.9	2.41	10.6	46.1	13.7	-	-
Bi-LSTM+TempoAttn [66]	22.8	2.17	10.2	42.2	11.8	-	-
GVD (MHA) [65]	23.6	2.35	11.0	45.5	14.9	7.59	25.0
GVD (VCM + SCM) (ours)	23.1	2.34	10.9	46.1	14.5	-	-
GVD (MHA + SCM) (ours)	23.6	2.54	11.2	47.7	15.0	7.30	24.4
GVD (MHA + SCM-VG) (ours)	**24.1**	**2.63**	**11.4**	**49.0**	**15.1**	**7.81**	**27.1**

the same as before but with visual-to-symbolic node assignment weights initialized based on knowledge transfer from the Visual Genome dataset [28] (more details can be found in the supplementary material). We use a symbolic graph whose nodes correspond to object classes. As shown in Table 4, replacing MHA with our visual module does not improve captioning, but it improves localization accuracy with a relative improvement of 4% (24.1 → 25.2). Adding our Semantic Context Module to GVD leads to an improvement across all captioning and localization metrics, which is even more pronounced in the test set (improving CIDEr from 45.5 to 47.7). Note that the initial region features already captured semantic information by including object class probabilities. Therefore, the improvement in captioning cannot be attributed solely to the inclusion of semantic context, but rather to our semantic reasoning framework. Finally, from the superior captioning performance of our third variant, we conclude that prior knowledge about correspondences between visual and symbolic nodes, if available, can possibly facilitate representation learning on the hybrid graph.

5 Conclusions

In this paper, we proposed a novel deep learning framework for video understanding that performs joint representation learning on a hybrid graph, composed of a symbolic graph and a visual st-graph, for obtaining context-aware visual node and edge features. We obtained state-of-the-art performance on three challenging datasets, demonstrating the effectiveness and generality of our framework.

Acknowledgements: The authors would like to thank Carolina Pacheco Oñate, Paris Giampouras and the anonymous reviewers for their valuable comments. This research was supported by the IARPA DIVA program via contract number D17PC00345.

References

1. Assari, S.M., Zamir, A.R., Shah, M.: Video classification using semantic concept co-occurrences. In: IEEE Conference on Computer Vision and Pattern Recognition (2014)
2. Bajaj, M., Wang, L., Sigal, L.: G3raphground: Graph-based language grounding. In: IEEE International Conference on Computer Vision. pp. 4281–4290 (2019)
3. Baradel, F., Neverova, N., Wolf, C., Mille, J., Mori, G.: Object level visual reasoning in videos. In: European Conference on Computer Vision. pp. 106–122 (2018)
4. Carreira, J., Zisserman, A.: Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 4724–4733 (2017). https://doi.org/10.1109/CVPR.2017.502
5. Chen, X., Li, L., Fei-Fei, L., Gupta, A.: Iterative Visual Reasoning Beyond Convolutions. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 7239–7248 (2018). https://doi.org/10.1109/CVPR.2018.00756
6. Chen, Y., Rohrbach, M., Yan, Z., Shuicheng, Y., Feng, J., Kalantidis, Y.: Graph-based global reasoning networks. In: IEEE Conference on Computer Vision and Pattern Recognition (2019)
7. Chéron, G., Laptev, I., Schmid, C.: P-CNN: Pose-Based CNN Features for Action Recognition. In: IEEE International Conference on Computer Vision. pp. 3218–3226 (2015). https://doi.org/10.1109/ICCV.2015.368
8. Choi, M.J., Lim, J.J., Torralba, A., Willsky, A.S.: Exploiting hierarchical context on a large database of object categories. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 129–136 (2010). https://doi.org/10.1109/CVPR.2010.5540221
9. Dave, A., Russakovsky, O., Ramanan, D.: Predictive-Corrective Networks for Action Detection. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 2067–2076 (2017). https://doi.org/10.1109/CVPR.2017.223
10. Deng, J., Ding, N., Jia, Y., Frome, A., Murphy, K., Bengio, S., Li, Y., Neven, H., Adam, H.: Large-Scale Object Classification Using Label Relation Graphs. In: European Conference on Computer Vision. pp. 48–64. Lecture Notes in Computer Science (2014)
11. Deng, Z., Vahdat, A., Hu, H., Mori, G.: Structure Inference Machines: Recurrent Neural Networks for Analyzing Relations in Group Activity Recognition. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 4772–4781 (2016). https://doi.org/10.1109/CVPR.2016.516
12. Ghosh, P., Yao, Y., Davis, L., Divakaran, A.: Stacked spatio-temporal graph convolutional networks for action segmentation. In: IEEE Winter Applications of Computer Vision Conference (2020)
13. Gilmer, J., Schoenholz, S.S., Riley, P.F., Vinyals, O., Dahl, G.E.: Neural Message Passing for Quantum Chemistry. In: International Conference on Machine learning. pp. 1263–1272 (2017)
14. Girdhar, R., Carreira, J., Doersch, C., Zisserman, A.: Video action transformer network. In: IEEE Conference on Computer Vision and Pattern Recognition (2019)
15. Gkioxari, G., Girshick, R., Malik, J.: Contextual Action Recognition with R*CNN. In: IEEE International Conference on Computer Vision. pp. 1080–1088 (2015). https://doi.org/10.1109/ICCV.2015.129
16. Gong, L., Cheng, Q.: Exploiting edge features for graph neural networks. In: IEEE Conference on Computer Vision and Pattern Recognition (2019)

17. He, K., Gkioxari, G., Dollar, P., Girshick, R.: Mask R-CNN. IEEE Transactions on Pattern Analysis and Machine Intelligence pp. 1–1 (2018). https://doi.org/10.1109/TPAMI.2018.2844175
18. He, K., Zhang, X., Ren, S., Sun, J.: Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In: IEEE International Conference on Computer Vision (2015)
19. Huang, H., Zhou, L., Zhang, W., Xu, C.: Dynamic Graph Modules for Modeling Higher-Order Interactions in Activity Recognition. In: British Machine Vision Conference (2019)
20. Ibrahim, M.S., Mori, G.: Hierarchical relational networks for group activity recognition and retrieval. In: European Conference on Computer Vision. pp. 721–736 (2018)
21. Jain, A., Zamir, A.R., Savarese, S., Saxena, A.: Structural-RNN: Deep Learning on Spatio-Temporal Graphs. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 5308–5317 (2016)
22. Jiang, C., Xu, H., Liang, X., Lin, L.: Hybrid Knowledge Routed Modules for Large-scale Object Detection. In: Neural Information Processing Systems, pp. 1552–1563 (2018)
23. Jiang, Y.G., Wu, Z., Wang, J., Xue, X., Chang, S.F.: Exploiting feature and class relationships in video categorization with regularized deep neural networks. IEEE Transactions on Pattern Analysis and Machine Intelligence **40**(2), 352–364 (2018). https://doi.org/10.1109/TPAMI.2017.2670560
24. Junior, N.I.N., Hu, H., Zhou, G., Deng, Z., Liao, Z., Mori, G.: Structured Label Inference for Visual Understanding. IEEE Transactions on Pattern Analysis and Machine Intelligence pp. 1–1 (2019). https://doi.org/10.1109/TPAMI.2019.2893215
25. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. In: International Conference on Learning Representations (2017)
26. Koller, D., Weber, J., Huang, T., Malik, J., Ogasawara, G., Rao, B., Russell, S.: Towards robust automatic traffic scene analysis in real-time. In: IEEE Conference on Computer Vision and Pattern Recognition (1994)
27. Koppula, H.S., Gupta, R., Saxena, A.: Learning human activities and object affordances from rgb-d videos. International Journal Robotics Research **32**(8), 951–970 (2013). https://doi.org/10.1177/0278364913478446
28. Krishna, R., Zhu, Y., Groth, O., Johnson, J., Hata, K., Kravitz, J., Chen, S., Kalantidis, Y., Li, L.J., Shamma, D.A., Bernstein, M.S., Fei-Fei, L.: Visual genome: Connecting language and vision using crowdsourced dense image annotations. International Journal of Computer Vision **123**(1), 3273 (2017). https://doi.org/10.1007/s11263-016-0981-7, `https://doi.org/10.1007/s11263-016-0981-7`
29. Lea, C., Flynn, M.D., Vidal, R., Reiter, A., Hager, G.D.: Temporal Convolutional Networks for Action Segmentation and Detection. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 1003–1012 (2017). https://doi.org/10.1109/CVPR.2017.113
30. Lee, C., Fang, W., Yeh, C., Wang, Y.F.: Multi-label Zero-Shot Learning with Structured Knowledge Graphs. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 1576–1585 (2018). https://doi.org/10.1109/CVPR.2018.00170
31. Li, R., Tapaswi, M., Liao, R., Jia, J., Urtasun, R., Fidler, S.: Situation recognition with graph neural networks. In: IEEE International Conference on Computer Vision (2017)

32. Li, Y., Gupta, A.: Beyond grids: Learning graph representations for visual recognition. In: Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., Garnett, R. (eds.) Neural Information Processing Systems. pp. 9225–9235 (2018)
33. Liang, X., Hu, Z., Zhang, H., Lin, L., Xing, E.P.: Symbolic Graph Reasoning Meets Convolutions. In: Neural Information Processing Systems. pp. 1853–1863. Curran Associates, Inc. (2018)
34. 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. pp. 740–755. Springer International Publishing, Cham (2014)
35. Liu, J., Kuipers, B., Savarese, S.: Recognizing human actions by attributes. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 3337–3344 (2011). https://doi.org/10.1109/CVPR.2011.5995353
36. Ma, C., Kadav, A., Melvin, I., Kira, Z., AlRegib, G., Graf, H.P.: Attend and Interact: Higher-Order Object Interactions for Video Understanding. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 6790–6800 (2018). https://doi.org/10.1109/CVPR.2018.00710
37. Marszalek, M., Schmid, C.: Semantic Hierarchies for Visual Object Recognition. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 1–7 (2007). https://doi.org/10.1109/CVPR.2007.383272
38. Marszalek, M., Schmid, C.: Constructing Category Hierarchies for Visual Recognition. In: European Conference on Computer Vision. pp. 479–491. ECCV '08, Springer-Verlag, Berlin, Heidelberg (2008)
39. Mavroudi, E., Tao, L., Vidal, R.: Deep Moving Poselets for Video Based Action Recognition. In: IEEE Winter Applications of Computer Vision Conference. pp. 111–120 (2017). https://doi.org/10.1109/WACV.2017.20
40. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed Representations of Words and Phrases and their Compositionality. In: Neural Information Processing Systems, pp. 3111–3119 (2013)
41. Nicolicioiu, A., Duta, I., Leordeanu, M.: Recurrent space-time graph neural networks. In: Neural Information Processing Systems (2019)
42. Oliva, A., Torralba, A.: The role of context in object recognition. Trends in Cognitive Sciences **11**(12), 520 – 527 (2007). https://doi.org/https://doi.org/10.1016/j.tics.2007.09.009
43. Piergiovanni, A., Ryoo, M.S.: Learning Latent Super-Events to Detect Multiple Activities in Videos. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 5304–5313 (2018). https://doi.org/10.1109/CVPR.2018.00556
44. Piergiovanni, A.J., Ryoo, M.S.: Temporal gaussian mixture layer for videos. In: International Conference on Machine learning (2019)
45. Prest, A., Ferrari, V., Schmid, C.: Explicit modeling of human-object interactions in realistic videos. IEEE Transactions on Pattern Analysis and Machine Intelligence (2013)
46. Qi, S., Wang, W., Jia, B., Shen, J., Zhu, S.C.: Learning human-object interactions by graph parsing neural networks. In: European Conference on Computer Vision. pp. 401–417 (2018)
47. Ramanathan, V., Li, C., Deng, J., Han, W., Li, Z., Gu, K., Song, Y., Bengio, S., Rossenberg, C., Fei-Fei, L.: Learning semantic relationships for better action retrieval in images. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 1100–1109 (2015). https://doi.org/10.1109/CVPR.2015.7298713
48. Schlichtkrull, M., Kipf, T.N., Bloem, P., Van Den Berg, R., Titov, I., Welling, M.: Modeling relational data with graph convolutional networks. In: European Semantic Web Conference. pp. 593–607. Springer (2018)

49. Sigurdsson, G.A., Divvala, S., Farhadi, A., Gupta, A.: Asynchronous Temporal Fields for Action Recognition. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 5650–5659 (2017). https://doi.org/10.1109/CVPR.2017.599
50. 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. pp. 510–526 (2016)
51. Simonyan, K., Zisserman, A.: Two-stream convolutional networks for action recognition in videos. In: Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N.D., Weinberger, K.Q. (eds.) Neural Information Processing Systems. pp. 568–576. Curran Associates, Inc. (2014)
52. Sun, C., Shrivastava, A., Vondrick, C., Murphy, K., Sukthankar, R., Schmid, C.: Actor-centric relation network. In: European Conference on Computer Vision. pp. 318–334 (2018)
53. Teney, D., Liu, L., van den Hengel, A.: Graph-structured representations for visual question answering. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 1–9 (2017)
54. Tran, D., Bourdev, L., Fergus, R., Torresani, L., Paluri, M.: Learning spatiotemporal features with 3d convolutional networks. In: IEEE International Conference on Computer Vision (2015)
55. Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., Bengio, Y.: Graph Attention Networks. International Conference on Learning Representations (2018)
56. Wang, L., Xiong, Y., Wang, Z., Qiao, Y., Lin, D., Tang, X., Val Gool, L.: Temporal segment networks: Towards good practices for deep action recognition. In: European Conference on Computer Vision (2016)
57. Wang, X., Gupta, A.: Videos as space-time region graphs. In: European Conference on Computer Vision. pp. 413–431 (2018)
58. Wang, X., Ji, Q.: Video event recognition with deep hierarchical context model. In: IEEE Conference on Computer Vision and Pattern Recognition (2015)
59. Xiong, Y., Huang, Q., Guo, L., Zhou, H., Zhou, B., Lin, D.: A graph-based framework to bridge movies and synopses. In: IEEE International Conference on Computer Vision (2019)
60. Xu, H., Das, A., Saenko, K.: R-C3d: Region Convolutional 3d Network for Temporal Activity Detection. In: IEEE International Conference on Computer Vision. pp. 5794–5803 (2017). https://doi.org/10.1109/ICCV.2017.617
61. Yatskar, M., Zettlemoyer, L., Farhadi, A.: Situation recognition: Visual semantic role labeling for image understanding. In: IEEE Conference on Computer Vision and Pattern Recognition (2016)
62. Yuan, Y., Liang, X., Wang, X., Yeung, D., Gupta, A.: Temporal Dynamic Graph LSTM for Action-Driven Video Object Detection. In: IEEE International Conference on Computer Vision. pp. 1819–1828 (2017). https://doi.org/10.1109/ICCV.2017.200
63. Zhang, Y., Tokmakov, P., Hebert, M., Schmid, C.: A structured model for action detection. In: IEEE Conference on Computer Vision and Pattern Recognition (2019)
64. Zhou, B., Andonian, A., Oliva, A., Torralba, A.: Temporal relational reasoning in videos. In: Computer Vision – ECCV 2018. pp. 831–846 (2018)
65. Zhou, L., Kalantidis, Y., Chen, X., Corso, J.J., Rohrbach, M.: Grounded video description. In: IEEE Conference on Computer Vision and Pattern Recognition (2019)
66. Zhou, L., Zhou, Y., Corso, J.J., Socher, R., Xiong, C.: End-to-end dense video captioning with masked transformer. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 8739–8748 (2018)

67. Zhou, Y., Ni, B., and, Tian, Q.: Interaction part mining: A mid-level approach for fine-grained action recognition. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 3323–3331 (2015). https://doi.org/10.1109/CVPR.2015.7298953
68. Zhu, Y., Nayak, N.M., Roy-Chowdhury, A.K.: Context-aware modeling and recognition of activities in video. In: IEEE Conference on Computer Vision and Pattern Recognition (2013)
69. Zitnik, M., Agrawal, M., Leskovec, J.: Modeling polypharmacy side effects with graph convolutional networks. Bioinformatics p. 457466 (2018)