Neural Message Passing on Hybrid Spatio-Temporal Visual and Symbolic Graphs for Video Understanding

Effrosyni Mavroudi, Benjamín Béjar Haro, René Vidal
Johns Hopkins University, Baltimore, MD, 21218, USA
{emavrou1,bbejar,rvidal1}@jhu.edu

Abstract

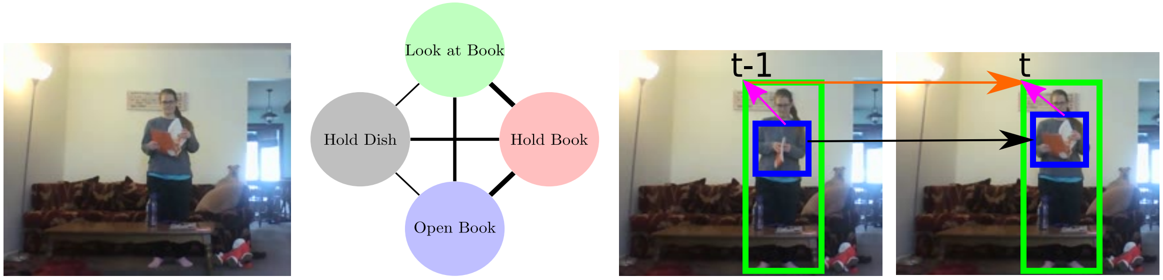
Many problems in video understanding require labeling multiple activities occurring concurrently in different parts of a video, including the objects and actors participating in such activities. However, state-of-the-art methods in computer vision focus primarily on tasks such as action classification, action detection, or action segmentation, where typically only one action label needs to be predicted. In this work, we propose a generic approach to classifying one or more nodes of a spatio-temporal graph grounded on spatially localized semantic entities in a video, such as actors and objects. In particular, we combine an attributed spatio-temporal visual graph, which captures visual context and interactions, with an attributed symbolic graph grounded on the semantic label space, which captures relationships between multiple labels. We further propose a neural message passing framework for jointly refining the representations of the nodes and edges of the hybrid visual-symbolic graph. Our framework features a) node-type and edge-type conditioned filters and adaptive graph connectivity, b) a soft-assignment module for connecting visual nodes to symbolic nodes and vice versa, c) a symbolic graph reasoning module that enforces semantic coherence and d) a pooling module for aggregating the refined node and edge representations for downstream classification tasks. We demonstrate the generality of our approach on a variety of tasks, such as temporal subactivity classification and object affordance classification on the CAD120 dataset and multilabel temporal action localization task on the large scale Charades dataset, where we outperform existing deep learning approaches, using only raw RGB frames.

1. Introduction

Consider the video frame shown in Figure 1a. It shows a person 'standing', 'holding a book', 'opening a book' and 'looking at a book' at the same time. What are the cues that convey these actions? In our example, recognizing the action 'look at a book' requires capturing the spatial interaction between the actor and the object, while 'open a book' requires temporal reasoning as well, considering the change of the pose of the actor (actor to actor temporal interaction) and the change of shape of the object (object to object temporal interaction). We therefore argue that *visual spatio-temporal interactions between semantic entities* are powerful contextual cues. Furthermore, we know that 'holding a book' is semantically similar to holding any other object, such as 'holding a dish'. We also know that the action 'hold a book' frequently co-occurs with 'open a book', hence commonsense label relationships constitute a second type of contextual cue.

In this work, we propose a visual spatio-temporal directed attributed graph (visual st-graph) grounded on semantic spatially localized entities, such as actors and objects, as a way of capturing visual contextual cues in videos. A simplified example of our visual st-graph instantiated on regions of two video frames is shown in Figure 1c. The nodes of this graph correspond to spatial regions and the edges encode spatio-temporal interactions. Both the nodes and the edges of this graph have types. Nodes (edges) of the same type are denoted with the same color.

We also introduce a novel Spatio-Temporal Message Passing Neural Network (ST-MPNN) for refining the representations of the visual st-graph nodes and edges. Our proposed model belongs to a class of deep learning models that can be directly applied to graphs, called Graph Neural Networks, which aim to extract high-level, discriminative, context-aware features from each node by taking into account its neighboring nodes and adjacent edges. Many of these approaches follow a "message passing" scheme [2, 12, 24, 23] by learning to iteratively update the hidden states of nodes by aggregating the hidden states of their neighbors, where each iteration is parameterized by shallow neural networks. Our proposed ST-MPNN follows a similar message passing approach, tailored to video understanding. First, inspired by the Dynamic Edge-Conditioned Filters [45] and the Structural-RNN [19], we learn *node- and edge-type-conditioned filtering weights*. Therefore, the net-



(a) Example of a video frame. (b) Symbolic Graph (c) Visual Spatio-Temporal Graph

Figure 1: How do we recognize the actions performed in the frame shown in (a)? We argue that there are two types of vital cues for video understanding: (1) external commonsense *semantic relationships of labels*, such as linguistic similarity or co-occurrence and (2) *visual spatio-temporal contextual cues*, such as human-object interactions. In this work, we perform representation learning on a hybrid symbolic (b) and visual (c) graph to leverage both types of cues. Visual node types are *actor* (shown in green) and *object* (shown in blue). Edge types include *object-to-actor spatial* (shown in magenta) and *actor-to-actor temporal* (shown in orange).

work learns distinct and specialized message passing mechanisms for each different type of interaction, such as *spatial object to actor interactions* or *temporal object to object interactions*. Second, the ST-MPNN *adapts the graph connectivity* during graph propagation, capturing the intuition that in any frame there are a few important interactions between actors and objects. Furthermore, *geometric relations and interactions between nodes* are utilized both for the node updates and the adaptive connectivity computation.

We further propose leveraging a *symbolic graph*, which encodes external commonsense knowledge, such as semantic label similarity, as shown in Figure 1b. Our goal is to integrate this prior knowledge about the inter-class relationships into the visual representation of the nodes of the visual st-graph. To achieve that, we connect the symbolic graph to the nodes of the visual st-graph, and apply a Graph Convolutional Network (GCN) [24] on the visual evidence combined with linguistic symbolic embeddings, to perform global semantic reasoning. The visual st-graph node representations are then enhanced by the globally learned representations. Finally, we use both context-enhanced node and edge features for final predictions.

In summary, the contributions of this work are three-fold. First, we propose a new framework that combines an external knowledge symbolic graph combined with a visual st-graph and we propose a method for performing joint representation learning on that hybrid graph for video understanding. Second, we introduce a novel Spatio-Temporal Message Passing Neural Network for refining the node and edge features of the visual st-graph. Our ST-MPNN a) supports node-type- and edge-type-conditioned filters, b) features adaptive graph connectivity, and c) utilizes spatial relations and interactions. Finally, we evaluate our method on tasks such as multilabel temporal activity localization and object affordance detection on two challenging datasets and show that it achieves state-of-the-art performance. Importantly, we do not assume access to ground truth bounding boxes, tracks or semantic labels of regions in training.

2. Related work

Context for video understanding. Context and its role in vision and video understanding has been studied for a long time [37]. Context can be captured by coarse, global features, for instance by training deep networks [3, 46] on video clips or by explicitly considering scene cues [32]. Furthermore, long-term temporal context plays a significant role in action recognition [26, 38]. Another way of explicitly exploiting context is by using mid-level representations of semantic parts. Researchers have exploited body parts [5, 35], latent attributes [30], human-object interactions [40, 55], object-object interactions [31] and secondary regions [14] to help discriminate between actions.

Graph neural networks for video understanding. A natural representation of activities that encodes spatio-temporal context is by using visual graphs. The first approach of applying a deep network on a visual graph for group action recognition was the Structured Inference Machine (SIN) by Deng et al. [10], which featured actor feature refinement with message passing and trainable gating functions for filtering out spurious interactions. Although the SIN model is just spatial, our ST-MPNN shares the message passing philosophy and intuition of learnable edge weights (adaptive graph connectivity). Another related approach is S-RNN [19], which although it does not iteratively refine node and edge representations, it introduced the concept of weight-sharing between nodes or edges of the same type. Advances by the machine learning community in the design of Graph Neural Networks (GNNs), have recently lead to numerous applications on video understanding, where researchers have modeled whole frames [54], tracklets [53], feature map columns [47, 13] or frame object proposals [49, 17, 52] as graph nodes and have used off-the-shelf GNNs, such as MPNNs [12], GCNs [24] and Relation Networks [18, 47, 54, 1] to refine their node or edge representations, leading to significant performance gains. Except for a few works, such as [41], most of the GNNs are applied on graphs with pre-specified edge weights. In contrast, our model operates on a node- and edge-typed visual st-graph, features learnable adaptive connectivity [48, 51, 16] and refines the attributes of both nodes and edges.

Symbolic graphs. There is a long line of work on exploiting external knowledge encoded in label relation graphs for visual recognition tasks. For example, semantic label hierarchies, such as co-occurrence, exclusion, subsumption, hypernymy and meronymy, have been leveraged for improving visual object recognition [34, 33, 7, 9] or multi-label zero-shot learning [27]. Most of these approaches directly perform inference on the knowledge-graph. Rather, we aim to use the semantics of labels to integrate prior knowledge about the inter-class relationships into the visual representation of the nodes of the visual spatio-temporal graph. In a similar spirit, Chen et al. [4] combine a visual graph instantiated on objects with knowledge graphs and perform graph representation learning using GCNs. Similarly, [20] enforce the scalar edge weights between visual regions to be consistent with the edges of the symbolic graph and then refine the visual node representations. In addition to extending these approaches to the case of node- and edge-typed spatio-temporal visual attributed graphs, our method also does not assume access to semantic labels of regions during training. The closest work to ours is the Semantic Graph Reasoning layer [28], that can be injected between any convolutional layers and used to improve image recognition tasks, and which we extend for connecting the nodes of the visual and symbolic graphs. Much fewer papers utilize knowledge graphs for improving action recognition [42, 21]. A notable exception is the SINN [21], which performs graph-based inference in a hierarchical label space for action recognition, by iteratively refining the representations of concept layers using graph neural networks. Their symbolic graph is not linguistically grounded, whereas we use a symbolic graph grounded on individual semantic labels associated with linguistic attributes, such as distributed word embeddings.

3. Method

3.1. Neural Message Passing on Visual Spatio-Temporal Graph

Visual spatio-temporal graph. Our input is a sequence of T frames with spatially localized semantic entities, such as actors and objects. 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 $E^v \subseteq V \times V$ is a set of edges. The nodes of this graph represent spatial regions of the video, corresponding to semantic entities, such as actors (people) and objects. If we assume M actors and N objects per timestep, the number of nodes is $|V^v| = (M+N)T$.

We assume the graph is both node- and edge-typed, i.e. there exists function $n : V^v \mapsto \{0, \mathcal{N}-1\}$ assigning types to each node and $e : E^v \mapsto \{0, \ldots, \mathcal{E}-1\}$ assigning types to each edge, where $\mathcal{N}$ is the number of node types and $\mathcal{E}$ is the number of edge types. For example, the node types can be *actor* and *object* ($\mathcal{N} = 2$) and the edge types can be: object-to-actor spatial (*obj-act-sp*), actor-to-object spatial (*act-obj-sp*), object-to-actor spatial (*obj-act-sp*), actor-to-actor temporal (*act-act-t*) and obj-to-obj temporal (*obj-obj-t*) ($\mathcal{E} = 5$). Furthermore, nodes and edges of the visual st-graph are associated with attributes.

Each node type and edge type can be associated with an attribute of dimensions F_ν and F_ϵ, respectively, where $\nu = 0, \ldots, \mathcal{N}-1$ denotes the node type and $\epsilon = 0, \ldots, \mathcal{E}-1$ denotes the edge type. For example, the i-th node corresponding to the j-th region at frame t has an attribute $\mathbf{h}_i^{(0)} = \mathbf{f}_{j,t} \in \mathbb{R}^{F_\nu}$, where $\nu = n(i)$ is the type of the node i and $\mathbf{f}_{j,t}$ is the appearance feature extracted from the region j at time t. Similarly, the edge connecting nodes i and j has an attribute $\mathbf{h}_{ij}^{(0)} \in \mathbb{R}^{F_\epsilon}$, where $\epsilon = e(i,j)$ is the type of the edge from j to i. This attribute corresponds to the relative spatial location of nodes. To finalize the construction of our input visual st-graph, we need to specify a binary adjacency matrix $L^v \in \{0,1\}^{|V^v|\times|V^v|}$, which specifies the allowed spatio-temporal connections between nodes. For instance, we can constrain temporal edges to connect a node at frame t with another node of the same type at time $t' = t-1$. This adjacency matrix defines the neighborhood of each node and therefore encodes the family of spatio-temporal interactions captured by the model.

Visual graph refinement using message passing neural network. We now describe how to perform representation learning on the visual st-graph in order to refine node and edge attributes. Specifically, given a spatio-temporal visual st-graph G^v with initial node and edge attributes, $\{\mathbf{h}_i^{(0)}\}_{i \in V^v}$ and $\{\mathbf{h}_{ij}^{(0)}\}_{(i,j) \in E^v}$, that capture local appearance and geometric relations, respectively, we introduce a novel Graph Neural Network model to enhance these local features with spatio-temporal contextual cues. In particular, we introduce a layer-wise propagation rule for a Spatio-Temporal Message Passing Neural Network (ST-MPNN), which operates directly on the spatio-temporal graph. Each iteration of neural message passing in our ST-MPNN consists of two steps: (1) refine the adjacency matrix (graph structure), which captures the connectivity between nodes, by estimating the importance of node j for updating the representation of node i; and (2) update the states/attributes of a node as a weighted sum of the states of neighboring nodes and the states of an edge using the message that was computed along that edge.

– *Adaptive graph connectivity:* Formally, at each iteration l of the ST-MPNN, we first refine the graph connectivity by computing *attention coefficients* a_{ij}, capturing the significance of node j for the update of node i. Our proposed attention computation is inspired by Graph Attention Networks [48], which we extend by supporting directed edges, discrete node and edge types, as well as by exploiting the edge state (geometric relation or interaction information) as:

$$a_{ij}^{(l)} = \underset{j}{\text{softmax}}(\gamma_{ij}^{(l)}) = \frac{\exp\left(\gamma_{ij}^{(l)}\right)}{\sum_{k \in N_\epsilon(i)} \exp\left(\gamma_{ik}^{(l)}\right)}, \quad (1)$$

where

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

Here, $N_\epsilon(i)$ denotes the set of neighbors of node i connected with i via an incoming edge of type ϵ, $N_\epsilon(i) = \{ j \mid L_{ij}^v = 1, e(i,j) = \epsilon \}$, $\mathbf{h}_{ij}^{(l-1)}$ is the state of the edge from node j to node i as computed in the previous iteration, $\epsilon = e(i,j)$ is the type of the edge from j to i, $\nu_r = n(i)$ is the node type of the receiver node i, $\nu_s = n(j)$ is the node type of the sender node j, ρ is a non- linearity, such as Leaky-ReLU, λ_{ea} is binary scalar, denoting whether the state of the edge will be used in the computation of attention coefficients. $W_r^{\nu_r}$, $W_s^{\nu_s}$ and W_{rs}^ϵ are learnable receiver node-, sender node- and edge-projection weights, respectively. All projection matrices linearly transform the current node (edge) state to a higher level feature of fixed dimensionality d_l. $\mathbf{v}_a^\epsilon$ is a learnable attention vector. For improved readability we have dropped the layer indices from the attention and projection weights. Note that we share the linear transformations, parametrized by projection weight matrices, between nodes (edges) of the same type. In this way, we allow our model to learn an attention mechanism specialized for each type of interaction, e.g., for computing the attention coefficients of an actor node over the object nodes at the same frame.

– *Message computation:* After computing the attention coefficients, we compute a message along each edge, and use these messages to update the node and edge states. In particular, we compute the message from node j to node i as:

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

where λ_e is a binary scalar, denoting whether the edge state will be used in the message, λ_v is a binary scalar, denoting whether the sending node state will be used in the message and the learnable weight matrices are the same as the ones used in the attention computation. We, therefore, have three architecture choices for the message computation: (1) compute message using both the node and the edge info (*full* message), (2) compute message using only the node info (*nnode* message), or (3) compute message using only the edge state info (*relational* message).

– *Node and edge state update:* Following the message computation, the node state is updated using an aggregation of incoming messages and a residual connection (applying an additional linear transformation if needed), while the edge state is set to be equal to the message. Formally,

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

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

3.2. Representation Learning on Hybrid Visual Spatio-Temporal and Symbolic Graph

The actor and object node representations obtained by graph propagation on the visual st-graph are context- and interaction-aware, but they do not explicitly capture symbolic semantics. Thus, in this work we also propose using a symbolic graph, which encodes commonsense external knowledge, for enhancing intermediate feature representations. To perform joint representation learning on the hybrid graph, the symbolic graph is connected to the visual st-graph, thus it integrates visual evidence, which is used along linguistic symbolic embeddings for performing global semantic reasoning. The visual st-graph node representations are finally obtained by integrating the globally learned representations.

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 is associated with a semantic attribute, such as the linguistic embedding of the label. Let $\mathbf{s}_c \in \mathbb{R}^K$ be the linguistic feature for symbolic node c, where K is the dimension of the linguistic embedding. Edges in the symbolic graph are associated with scalar weights, which could encode co-occurence probabilities, semantic similarity between classes or class hierarchies. These edge weights are summarized in the fixed input 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 states of the symbolic graph using visual evidence, i.e., the representations of the nodes of the visual st-graph, which is grounded on regions of the input video. To achieve this, we learn associations between nodes of the visual st-graph and nodes of the symbolic graph. Specifically, the linguistic representation of each symbolic node is enriched with a visual feature computed by summing up all weighted transformed visual node features, where the weights are:

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

where $W^{vs} = \{W^{vs}_c\} \in \mathbb{R}^{d_L\times|V_s|}$ is a trainable weight matrix for calculating assignment weights and $\Omega \in \{0,1\}^{|V_s|\times|V_v|}$ is a pre-specified binary mask that defines allowed visual-to-symbolic node connections. For example, in the case that our symbolic nodes correspond to action classes, we would like to disable connections between object nodes with the symbolic nodes. The voting weight $\phi^{vs}_{c,i}$ represents the confidence of assigning the feature from visual node i to the symbolic node c. These voting weights can be intuitively thought of as the probability of each visual st-graph node being assigned to the semantic label of the symbolic graph node, although the ground truth semantic labels of visual nodes are in general unknown during training.

After computing the voting weights, each symbolic node is associated with a visual feature obtained as follows: $\tilde{\mathbf{f}}_c = \sum_i \phi^{vs}_{c,i}W^{vs}_p\mathbf{h}^{(L)}_i$, where $W^{vs}_p \in \mathbb{R}^{D_s\times d_L}$ is a learnable projection weight matrix. The new representation of each symbolic graph node c is computed as the concatenation of a) the linguistic embedding $\mathbf{s}_c$ and b) the visual feature $\tilde{\mathbf{f}}_c$ passed through a non-linearity, $\mathbf{s}_c(0) = \left[\mathbf{s}_c;\sigma(\tilde{\mathbf{f}}_c)\right]$, therefore it is a vector of dimension $K + D_s$.

– *Graph reasoning on symbolic graph:* The symbolic graph reasoning module performs graph propagation over representations of all symbolic nodes using a vanilla Graph Convolutional Network [24], resulting in evolved symbolic features $S^{(R)} \in \mathbb{R}^{|V^s|\times D_s}$. The layer-wise propagation rule is:

$$S^{(r+1)} = \sigma\left(\tilde{D}^{-\frac{1}{2}}\tilde{L}^s\tilde{D}^{-\frac{1}{2}}S^{(r)}W^{(r)}\right), \qquad (5)$$

where $S^{(r+1)}$ denotes the matrix of activation in the $r+1$-th layer of the GCN (or round of symbolic node update), $\tilde{L^s} = L^s + I_{|V_s|}$ is the adjacency matrix of the undirected symbolic graph G^s with added self- loops, $I_{|V_s|}$ is the identity matrix, $\tilde{D}$ is the diagonal degree matrix, $W^{(r)}$ is a layer-specific trainable weight matrix and $\sigma(\cdot)$ is a non-linearity, such as ReLU.

– *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 commonsense external knowledge. To achieve this we compute mapping weights (attention coefficients) from symbolic nodes to visual nodes:

$$\phi^{sv}_{i,c} = \frac{\omega_{c,i}\exp\left(e^{sv}_{i,c}\right)}{\sum_{c'\in V^s}\omega_{c,i}\exp\left(e^{sv}_{i,c'}\right)}, \qquad (6)$$

where $e^{sv}_{i,c} = \left(\mathbf{v}^{sv}_a\right)^T\left[\mathbf{s}^{(R)}_c;\mathbf{h}^{(L)}_i\right]$ and $\mathbf{v}^{sv}_a \in \mathbb{R}^{d_L+D_s}$ is a learnable attention vector. The final visual node feature representation is then given by:

$$\mathbf{h}_i = \mathbf{h}^{(L)}_i + \sigma\left(\sum_{c'\in V^s}\phi^{sv}_{i,c'}W^{sv}_p\mathbf{s}^{(R)}_{c'}\right). \qquad (7)$$

3.3. Node and edge feature aggregation

For various video understanding tasks, we are interested in classifying either a single node of the visual st-graph or a subset of nodes. For example, for subactivity and object affordance temporal segmentation, we are asked to predict a subactivity/affordance label per actor/object node. However, for multilabel temporal action localization, we have to predict multiple labels for the set of actor nodes at each frame. Formally, let $V^v_1, V^v_2, \ldots, V^v_P$ be P pre-defined subsets of the set of visual nodes V^v for which we would like to predict labels. To classify subset V^v_p, we need a single feature vector describing the subset. We therefore propose aggregating the refined representations of the nodes belonging to the subset (and optionally the refined representations of adjacent edges) into the following feature vector describing subset p:

$$\mathbf{f}_p = \frac{1}{|V^v_p|}\sum_{i\in V^v_p}\mathbf{h}_i + \frac{1}{|N_e|}\sum_{i\in V^v_p}\sum_{(i,j)\in E^v}\mathbf{h}_{ij} \qquad (8)$$

4. Experiments

4.1. CAD-120

Dataset, tasks and metrics. We consider the CAD-120 dataset, which provides RGBD data for 120 videos corresponding to 4 subjects. Each video consists of a sequence of sub-activities (e.g. moving, drinking, etc.) and object affordances (e.g. reachable, drinkable, etc.), which evolve over time. Since the activities of this dataset involve multiple human-object and object-object interactions, this dataset is a particularly good test-bed for analyzing our proposed method. Since in this dataset the number of semantic labels is small, we only evaluate the ST-MPNN module, without using a symbolic graph. The features of the humans and objects as well as features describing their relative geometric relations are provided by the dataset [25]. For an analytic description of the available hand-crafted features see [25].

Method	Detection F1-score (%)	
	Sub-activity	Object affordance
ATCRF [25]	80.4	81.5
S-RNN [19]	83.2	88.7
S-RNN [19] (multitask)	82.4	**91.1**
GPNN [41]	88.9	88.8
STGCN [11]	87.2	-
ST-MPNN (Ours)	**91.7**	*89.4*

Table 1: Results on CAD-120 [25] dataset for sub-activity and object affordance detection, measured via F1-score.

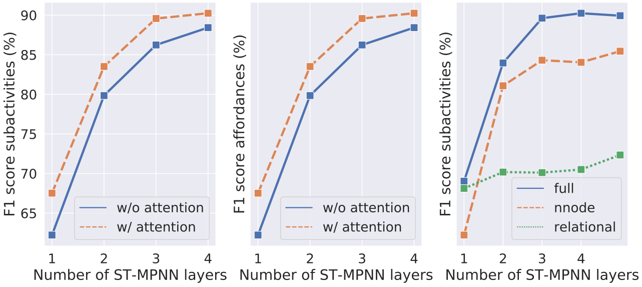


Figure 2: Effect of adaptive graph connectivity and node update type on CAD-120 [25] detection performance. "w/o attention": fixed visual adjacency matrix, "w/ attention": adaptive graph connectivity, "full", "nnode" and "relational": node update types.

Different types of features have different dimensionalities, which can be naturally handled by our model. Evaluation is performed using 4-fold cross-validation and averaging performance across the folds. In each fold we test our model on the activities performed by one subject and train on the sequences of the other three subjects. We report the F1-score for sub-activities and affordances averaged over all classes.

Implementation. We instantiate a visual st-graph on the actor and objects of each temporal segment of an input sequence. We also experiment with 5 edge types: edges connecting objects in the same temporal segment (*obj-obj-sp*), edges connecting objects with the actor within a temporal segment (*obj-act-sp*), edges connecting the actor with objects within a temporal segment (*act-obj-sp*), edges connecting actors between two consecutive temporal segments (*act-act-t*) and edges connecting objects between two consecutive temporal segments (*obj-obj-t*). After instantiating the visual st-graph, our task is to classify each actor node to one of the 10 subactivity classes and each object node to one of the 12 affordance classes. We refine the visual node representations using our proposed ST-MPNN model. We set the number of layers of our ST-MPNN to 4 and the size of all messages to $d = 256$. We use a *full* node update

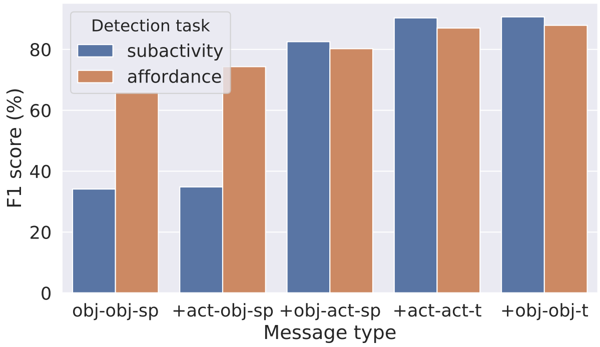


Figure 3: Ablation on CAD-120 [25] subactivity and affordance detection performance by incrementally adding edge types, starting from using only *object-to-object spatial* edges.

(Eq. 3 with $\lambda_v = 1$ and $\lambda_e = 1$) and we include the edge state in the attention computation (Eq. 2 with $\lambda_{ea} = 1$). We train our model using the sum of cross-entropy losses computed at each node of the st-graph. Our model is trained for 100 epochs, with a batch size of 5 sequences, using the *Adam* [22] learning rate scheduler, with an initial learning rate of 0.001, which we reduce by a factor of 0.1 once learning stagnates. We apply Dropout with a rate of 0.5 on all fully connected layers.

Results and comparison with the state of the art. Table 1 shows the subactivity and affordance detection F1-scores averaged over all the classes. ST-MPNN obtains state-of-the-art results for sub-activity detection ($\mathbf{91.7}\%$) and the second best result on affordance detection (89.3%), being only second to S-RNN (multi-task) [19], which is trained on the joint task of affordance detection and anticipation, and is, thus, not directly comparable to the rest of reported methods. Our model outperforms by more than 8.0% on both tasks the approach by Koppula et al. [25] (ATCRF), which models the same visual st-graph with the same features in a probabilistic framework. More importantly, our ST-MPNN improves upon other, recently proposed, Graph Neural Network architectures, such as the GPNN [41], validating our novel layer propagation rules.

Ablation analysis. In Fig. 2, we show the effect of different components of our model on the recognition performance. First, we compare the performance between an ST-MPNN trained with a fixed binary adjacency matrix with an ST-MPNN trained using attention. Clearly, adaptive graph connectivity benefits performance in both tasks and over different number of layers (message passing rounds). Second, we experiment with various node update types, and we conclude that using the states of both the neighboring nodes and adjacent edges is better than using only the neighboring node states. Ignoring the neighboring node states signifi-

Method	Feat	Stream	mAP (%)
Predictive-corrective [8]	VGG	R	8.9
Two-stream [43]	VGG	R+F	8.94
Two-stream + LSTM [43]	VGG	R+F	9.6
R-C3D [50]	VGG	R+F	12.7
ATF [43]	VGG	R+F	12.8
RGB I3D [38]	I3D	R	15.63
I3D [38]	I3D	R+F	17.22
I3D + LSTM [38]	I3D	R+F	18.12
RGB I3D + super-events [38]	I3D	R	18.64
I3D + super-events [38]	I3D	R+F	19.41
STGCN [11]	I3D	R+F	19.09
I3D + 3TGMs + super-events [39]	I3D	R+F	22.3
ST-MPNN + Symb (Ours)	I3D	R	15.3
ST-MPNN + Symb + biGRU (Ours)	I3D	R	18.6
ST-MPNN + Symb + I3D + biGRU (Ours)	I3D	R	**23.4**

Table 2: Comparison with the state of the art on Charades [44] dataset for multilabel temporal action localization. Performance is measured via per-frame mAP, evaluated with the official evaluation script `Charades_v1_localize`. Feat: type of feature, R: RGB input, F: optical flow input, biGRU: bidirectional Gated Recurrent Unit.

Table 3: Ablation analysis on Charades dataset.

Approach	mAP (%)	Approach	mAP (%)
Actor FC	10.69	+ act-obj-sp	12.70
+ obj-act-sp	11.15	+ attention	13.08
+ act-act-t	11.92	+ full node update	13.52
+ 2 layer	12.44	+ attention w/ edge	13.68
+ 3 layer	12.65	+ symbolic graph	**15.29**
+ pool edges	12.73		

cantly hurts performance. In Figure 3 we show the contribution of each edge type on the final performance.

4.2. Charades

Dataset, tasks and metrics. Charades [44] is a large scale dataset consisting of 9848 videos acros 157 activities. Each video contains an average of 6.8 activity instances, often with complex co-occurring activities and fine-grained human-object interactions, making it a suitable dataset to test our model and leverage both rich spatio-temporal contextual cues as well as external commonsense knowledge. Our goal is to temporally localize action instances. Therefore, we are interested in a multi-label classification of the subset of actor nodes at each frame. Performance is measured in terms of mean Average Precision by comparing per- frame predictions from 25 equidistant frames with ground-truth annotations in the official testing split.

Implementation details. For extracting visual features, we use the I3D model [3] fine-tuned on Charades, publicly shared by the authors of [38]. For obtaining bounding boxes of people and other objects, which are the nodes of our visual graph, we use the Faster-RCNN [15] pretrained on the MSCOCO [29] dataset. We keep the two highest scoring human detections ($M = 2$) and 10 object detections per frame ($N = 10$). We apply masking for handling frames with varying number of actors and objects. Rather than using the object detector features for describing the actors and objects, we exploit the rich spatio-temporal feature maps of the I3D action recognition model, by pooling features from the `Mixed_4f` feature map of the I3D, which has a spatial output stride of 16 pixels, a temporal output stride of 4 frames and 832 channels. In particular, we first temporally downsample the spatio-temporal feature map to obtain an effective temporal downsampling by a factor of 16 frames (1.5FPS) and then we apply RoIAlign [15] to pool features from each region at each downsampled frame. This leads to a feature map of $832 \times 7 \times 7$ per region per frame. To obtain a single feature vector for each actor and object node, we max-pool this feature map. In this dataset, we use 3 types of edges: *obj-act-sp*, *act-obj-sp* and *act-act-t*. The attribute associated with each edge is obtained by computing the relative position between the two bounding boxes.

Our symbolic graph has nodes corresponding to the 157 action classes. To obtain the linguistic embedding of each action class, we map its name to a verb and object pair, use off-the-self word2vec [36] embeddings of size $K = 300$ to represent the verb and the object and then we average them. The edge weights of the symbolic graph are obtained by computing frequencies of per-frame action label co-occurences in training data.

The architecture we use in this dataset is the following: we set the number of layers of our ST-MPNN to 3 and the size of all ST-MPNN messages to $d = 512$. We set the number of layers of the Symbolic Graph Reasoning module to 1 and the size of its messages to $D_c = 256$. Since Charades is a multi-label, multi-class dataset, we use the binary cross-entropy loss. Our model is trained for 40 epochs, with a batch size of 16 sequences, using the *Adam* learning rate scheduler, with an initial learning rate of 0.0001, which we reduce by a factor of 0.1 once learning stagnates. We apply Dropout with a rate of 0.5 on all fully connected layers.

Results. We compare our results with the state of the art in Table 2. Our proposed method reaches the same performance as the end-to-end trained I3D on RGB frames (15.3%), starting from a baseline of 10.69% corresponding to classifying local actor features extracted from an early feature map of the I3D. This result is obtained without using any scene context or long-term temporal context. By adding a bidirectional Gated Recurrent Unit [6] (biGRU) on top of our model, we obtain a performance of 18.6%. Fusing the predictions of this model with a biGRU trained on top of

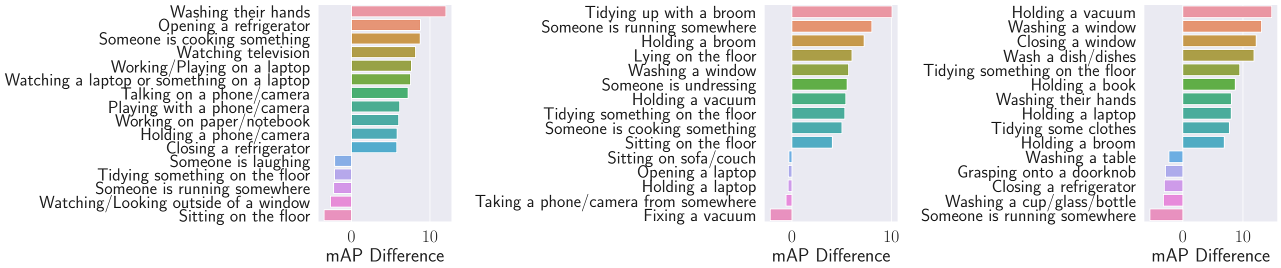


Figure 4: The classes with the highest positive and negative performance difference between different instantiations of our model. (a) Actor region classification (no context) vs our ST-MPNN with *object-to-actor spatial* messages. Incorporating structure benefits actions that involve interactions with objects far away from the actor, such as *watching television* or *cooking*. (b) Adding *actor-to-actor temporal* messages helps with long actions, such as *running*, and actions involving objects that are hard to detect (*Holding a broom*). (c) Adding symbolic graph benefits actions that have a few training examples, such as *Holding a vacuum* or have strong co-occurences, such as *Holding a book*.

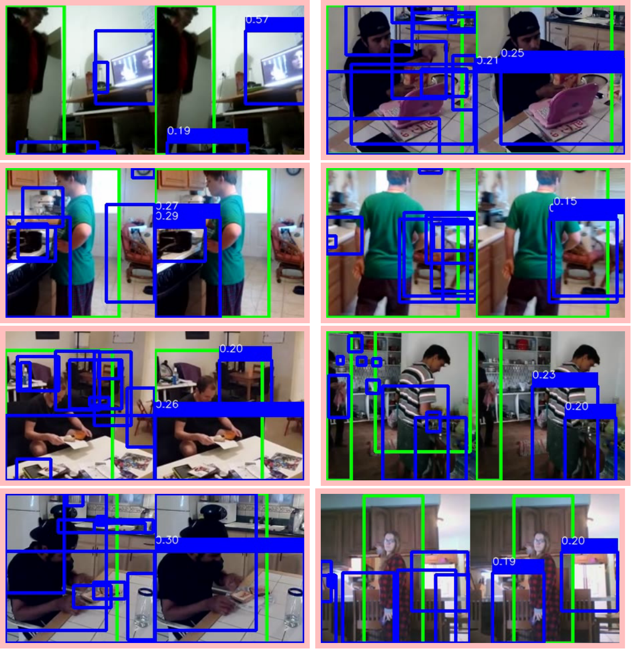


Figure 5: Visualization of attention over objects for updating the actor feature on sample frames from Charades dataset. Each pair of images shows: the original frame with the actor detection in green and object detections in blue (*left*) and the actor and the two objects with largest attention coefficients (*right*).

the I3D scene features, we obtain the best known performance in the localization setting of the Charades dataset, using only raw RGB frames. This result shows that the representations learned by our model, which capture visual spatio-temporal interactions between actors and objects, as well as semantic relationships, are complementary to representations that capture holistic scene cues and temporal dynamics. To gain a better understanding of the benefits gained by performing representation learning on the hybrid visual-symbolic graph, we highlight in Figure 4, the activity classes with the highest positive and negative difference between various versions of our model. By passing messages along appropriate edges of the st-graph, our model harnesses human-object interaction cues and local temporal context and by adding reasoning on the symbolic graph it incorporates external knowledge and helps recognizing rare classes, such as *Holding a vacuum*, which has only 213 training examples (3% of available annotated segments).

Table 3 validates the design of our model, by showing how each component helps in improving the overall model performance. The addition of the symbolic graph leads to a significant 2% improvement in mAP.

5. Conclusions

In this paper, we have proposed a novel deep learning framework for video understanding that performs joint representation learning on a hybrid graph composed of a symbolic graph and visual spatio-temporal graph. We also introduced a novel Neural Message Passing network (*ST-MPNN*) with adaptive graph connectivity and node-type- and edge-type-conditioned filters. We obtained state-of-the-art performance on two challenging datasets, demonstrating the effectiveness of our framework. We also presented an ablation analysis showing how our model benefits by capturing human-object interactions and semantic label relationships. Promising future directions include using symbolic graphs for modeling richer class hierarchies, applying our method to additional tasks such as weakly supervised object detection and extending our ST-MPNN model to perform hierarchical representation learning.

Acknowledgements: Authors would like to thank Carolina Pacheco Oñate and Ambar Pal for helpful comments. This research is based upon work supported by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Interior/ Interior Business Center (DOI/IBC) contract number D17PC00345. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DOI/IBC, or the U.S. Government.

References

[1] F. Baradel, N. Neverova, C. Wolf, J. Mille, and G. Mori. Object level visual reasoning in videos. In *European Conference on Computer Vision*, pages 106–122, Cham, 2018. Springer International Publishing. 3

[2] M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst. Geometric Deep Learning: Going beyond Euclidean data. *IEEE Signal Processing Magazine*, 34(4):18–42, July 2017. 1

[3] J. Carreira and A. Zisserman. Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4724–4733, July 2017. 2, 7

[4] X. Chen, L. Li, L. Fei-Fei, and A. Gupta. Iterative Visual Reasoning Beyond Convolutions. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 7239–7248, June 2018. 3

[5] G. Chéron, I. Laptev, and C. Schmid. P-CNN: Pose-Based CNN Features for Action Recognition. In *IEEE International Conference on Computer Vision*, pages 3218–3226, Dec. 2015. 2

[6] K. Cho, B. van Merrienboer, D. Bahdanau, and Y. Bengio. On the properties of neural machine translation: Encoder–decoder approaches. In *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, pages 103–111, Doha, Qatar, Oct. 2014. Association for Computational Linguistics. 7

[7] M. J. Choi, J. J. Lim, A. Torralba, and A. S. Willsky. Exploiting hierarchical context on a large database of object categories. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 129–136, June 2010. 3

[8] A. Dave, O. Russakovsky, and D. Ramanan. Predictive-Corrective Networks for Action Detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2067–2076, July 2017. 7

[9] J. Deng, N. Ding, Y. Jia, A. Frome, K. Murphy, S. Bengio, Y. Li, H. Neven, and H. Adam. Large-Scale Object Classification Using Label Relation Graphs. In *European Conference on Computer Vision*, Lecture Notes in Computer Science, pages 48–64, 2014. 3

[10] Z. Deng, A. Vahdat, H. Hu, and G. Mori. Structure Inference Machines: Recurrent Neural Networks for Analyzing Relations in Group Activity Recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4772–4781, June 2016. 2

[11] P. Ghosh, Y. Yao, L. S. Davis, and A. Divakaran. Stacked spatio-temporal graph convolutional networks for action segmentation. *arXiv preprint arXiv:1811.10575*, 2018. 6, 7

[12] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl. Neural Message Passing for Quantum Chemistry. In *International Conference on Machine learning*, pages 1263–1272, July 2017. 1, 3

[13] R. Girdhar, J. Carreira, C. Doersch, and A. Zisserman. Video Action Transformer Network. *CoRR*, Dec. 2018. arXiv: 1812.02707. 2

[14] G. Gkioxari, R. Girshick, and J. Malik. Contextual Action Recognition with R*CNN. In *IEEE International Conference on Computer Vision*, pages 1080–1088, 2015. 2

[15] K. He, G. Gkioxari, P. Dollar, and R. Girshick. Mask R-CNN. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2018. 7

[16] H. Hu, J. Gu, Z. Zhang, J. Dai, and Y. Wei. Relation Networks for Object Detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3588–3597, June 2018. 3

[17] H. Huang, L. Zhou, W. Zhang, and C. Xu. Dynamic Graph Modules for Modeling Higher-Order Interactions in Activity Recognition. *CoRR*, Dec. 2018. arXiv: 1812.05637. 2

[18] M. S. Ibrahim and G. Mori. Hierarchical relational networks for group activity recognition and retrieval. In *European Conference on Computer Vision*, pages 721–736, 2018. 3

[19] A. Jain, A. R. Zamir, S. Savarese, and A. Saxena. Structural-RNN: Deep Learning on Spatio-Temporal Graphs. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 5308–5317, June 2016. 1, 2, 6

[20] C. Jiang, H. Xu, X. Liang, and L. Lin. Hybrid Knowledge Routed Modules for Large-scale Object Detection. In *Neural Information Processing Systems*, pages 1552–1563. 2018. 3

[21] N. I. N. Junior, H. Hu, G. Zhou, Z. Deng, Z. Liao, and G. Mori. Structured Label Inference for Visual Understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2019. 3

[22] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015. 6

[23] T. Kipf, E. Fetaya, K.-C. Wang, M. Welling, and R. Zemel. Neural Relational Inference for Interacting Systems. In *International Conference on Machine learning*, pages 2688–2697, July 2018. 1

[24] T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2017. 1, 2, 3, 5

[25] H. S. Koppula, R. Gupta, and A. Saxena. Learning human activities and object affordances from rgb-d videos. *International Journal Robotics Research*, 32(8):951–970, July 2013. 5, 6

[26] C. Lea, M. D. Flynn, R. Vidal, A. Reiter, and G. D. Hager. Temporal Convolutional Networks for Action Segmentation and Detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1003–1012, July 2017. 2

[27] C. Lee, W. Fang, C. Yeh, and Y. F. Wang. Multi-label Zero-Shot Learning with Structured Knowledge Graphs. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1576–1585, June 2018. 3

[28] X. Liang, Z. Hu, H. Zhang, L. Lin, and E. P. Xing. Symbolic Graph Reasoning Meets Convolutions. In *Neural Information Processing Systems*, pages 1853–1863. Curran Associates, Inc., 2018. 3

[29] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*, pages 740–755, Cham, 2014. Springer International Publishing. 7

[30] J. Liu, B. Kuipers, and S. Savarese. Recognizing human actions by attributes. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3337–3344, June 2011. 2

[31] C. Ma, A. Kadav, I. Melvin, Z. Kira, G. AlRegib, and H. P. Graf. Attend and Interact: Higher-Order Object Interactions for Video Understanding. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 6790–6800, June 2018. 2

[32] M. Marszalek, I. Laptev, and C. Schmid. Actions in context. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2929–2936, June 2009. 2

[33] M. Marszalek and C. Schmid. Semantic Hierarchies for Visual Object Recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–7, June 2007. 3

[34] M. Marszalek and C. Schmid. Constructing Category Hierarchies for Visual Recognition. In *European Conference on Computer Vision*, ECCV '08, pages 479–491, Berlin, Heidelberg, 2008. Springer-Verlag. event-place: Marseille, France. 3

[35] E. Mavroudi, L. Tao, and R. Vidal. Deep Moving Poselets for Video Based Action Recognition. In *IEEE Winter Applications of Computer Vision Conference*, pages 111–120, Mar. 2017. 2

[36] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed Representations of Words and Phrases and their Compositionality. In *Neural Information Processing Systems*, pages 3111–3119. Curran Associates, Inc., 2013. 7

[37] A. Oliva and A. Torralba. The role of context in object recognition. *Trends in Cognitive Sciences*, 11(12):520 – 527, 2007. 2

[38] A. Piergiovanni and M. S. Ryoo. Learning Latent Super-Events to Detect Multiple Activities in Videos. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 5304–5313, June 2018. 2, 7

[39] A. J. Piergiovanni and M. S. Ryoo. Temporal Gaussian Mixture Layer for Videos. *CoRR*, Mar. 2018. arXiv: 1803.06316. 7

[40] A. Prest, V. Ferrari, and C. Schmid. Explicit Modeling of Human-Object Interactions in Realistic Videos. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(4):835–848, Apr. 2013. 2

[41] S. Qi, W. Wang, B. Jia, J. Shen, and S.-C. Zhu. Learning human-object interactions by graph parsing neural networks. In *European Conference on Computer Vision*, pages 401–417, 2018. 3, 6

[42] V. Ramanathan, C. Li, J. Deng, W. Han, Z. Li, K. Gu, Y. Song, S. Bengio, C. Rossenberg, and L. Fei-Fei. Learning semantic relationships for better action retrieval in images. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1100–1109, June 2015. 3

[43] G. A. Sigurdsson, S. Divvala, A. Farhadi, and A. Gupta. Asynchronous Temporal Fields for Action Recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 5650–5659, July 2017. 7

[44] G. A. Sigurdsson, G. Varol, X. Wang, A. Farhadi, I. Laptev, and A. Gupta. Hollywood in homes: Crowdsourcing data collection for activity understanding. In *European Conference on Computer Vision*, pages 510–526, 2016. 7

[45] M. Simonovsky and N. Komodakis. Dynamic Edge-Conditioned Filters in Convolutional Neural Networks on Graphs. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 29–38, July 2017. 1

[46] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Neural Information Processing Systems*, pages 568–576. Curran Associates, Inc., 2014. 2

[47] C. Sun, A. Shrivastava, C. Vondrick, K. Murphy, R. Sukthankar, and C. Schmid. Actor-centric relation network. In *European Conference on Computer Vision*, pages 318–334, 2018. 2, 3

[48] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio. Graph Attention Networks. *International Conference on Learning Representations*, 2018. 3, 4

[49] X. Wang and A. Gupta. Videos as space-time region graphs. In *European Conference on Computer Vision*, pages 413–431, Cham, 2018. Springer International Publishing. 2

[50] H. Xu, A. Das, and K. Saenko. R-C3d: Region Convolutional 3d Network for Temporal Activity Detection. In *IEEE International Conference on Computer Vision*, pages 5794–5803, Oct. 2017. 7

[51] J. Yang, J. Lu, S. Lee, D. Batra, and D. Parikh. Graph r-cnn for scene graph generation. In *European Conference on Computer Vision*, pages 670–685, 2018. 3

[52] Y. Yuan, X. Liang, X. Wang, D. Yeung, and A. Gupta. Temporal Dynamic Graph LSTM for Action-Driven Video Object Detection. In *IEEE International Conference on Computer Vision*, pages 1819–1828, Oct. 2017. 2

[53] Y. Zhang, P. Tokmakov, M. Hebert, and C. Schmid. A Structured Model For Action Detection. Dec. 2018. 2

[54] B. Zhou, A. Andonian, A. Oliva, and A. Torralba. Temporal relational reasoning in videos. In *Computer Vision – ECCV 2018*, pages 831–846, Cham, 2018. Springer International Publishing. 2, 3

[55] Y. Zhou, B. Ni, and, and Q. Tian. Interaction part mining: A mid-level approach for fine-grained action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3323–3331, June 2015. 2