Video Relation Detection with Spatio-Temporal Graph

Xufeng Qian[1] Yueting Zhuang*[1] Yimeng Li[1] Shaoning Xiao[1] Shiliang Pu[2] Jun Xiao[1]
Zhejiang University[1], Hikvision Research Institute[2]
{21821138,yzhuang,aquaird,shaoningx}@zju.edu.cn,pushiliang@hikvision.com,junx@cs.zju.edu.cn

ABSTRACT

What we perceive from visual content are not only collections of objects but the interactions between them. Visual relations, denoted by the triplet <*subject, predicate, object*>, could convey a wealth of information for visual understanding. Different from static images and because of the additional temporal channel, dynamic relations in videos are often correlated in both spatial and temporal dimensions, which make the relation detection in videos a more complex and challenging task. In this paper, we abstract videos into fully-connected spatial-temporal graphs. We pass message and conduct reasoning in these 3D graphs with a novel VidVRD model using graph convolution network. Our model can take advantage of spatial-temporal contextual cues to make better predictions on objects as well as their dynamic relationships. Furthermore, an online association method with a siamese network is proposed for accurate relation instances association. By combining our model (VRD-GCN) and the proposed association method, our framework for video relation detection achieves the best performance in the latest benchmarks. We validate our approach on benchmark ImageNet-VidVRD dataset. The experimental results show that our framework outperforms the state-of-the-art by a large margin and a series of ablation studies demonstrate our method's effectiveness.

CCS CONCEPTS

• **Computing methodologies** → *Activity recognition and understanding*;

KEYWORDS

video relation detection; visual relation detection; spatio-temporal graph convolutional network; siamese association network

ACM Reference Format:
Xufeng Qian[1] Yueting Zhuang*[1] Yimeng Li[1] Shaoning Xiao[1] Shiliang Pu[2] Jun Xiao[1]. 2019. Video Relation Detection with Spatio-Temporal Graph. In *Proceedings of the 27th ACM International Conference on Multimedia (MM '19), October 21–25, 2019, Nice, France.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3343031.3351058

1 INTRODUCTION

Understanding visual information is the central goal of computer vision. Relation detection in visual content is a challenging yet meaningful task, which requires to capture fine-grained visual cues including localizing where entities are and how do they interact with each other. While relations among objects in videos are significant ingredients for a deeper comprehension of dynamic visual content, relation detection and reasoning in videos (VidVRD) is rarely explored. The successful attempts of detecting video relation would not only help us build more effective models for some high-level visual understanding tasks such as visual question answering (VQA) and visual caption [36], but also facilitate the development of other fields in computer vision, e.g. video retrieval, video action detection, and video activity recognition [10, 32, 33, 43].

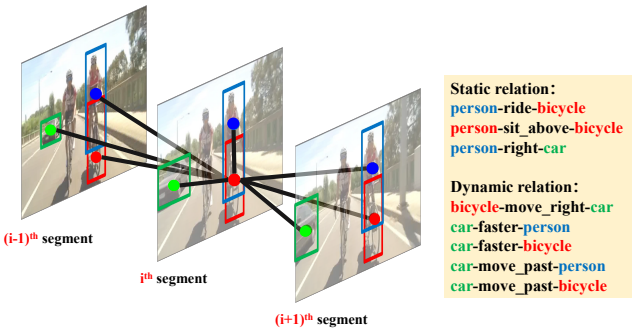

Figure 1: The left part visualizes the fully-connected spatio-temporal graph we construct for the bicycle node in i^{th} video segment. The right part shows the static relations and dynamic relations in the video to be detected.

Numerous recent researches have obtained exciting and significant results in static image relation detection [6, 16, 17, 20, 35, 37, 38, 40–42, 44]. A natural solution of relation detection in videos is to extend these methods to videos directly. However, unsatisfactory results were obtained due to the intrinsic differences between images and videos. Methods designed for ImgVRD tend to overlook dynamic interactions between entities, which yet always occur in videos. Considering the characteristics of videos, solutions for VidVRD are supposed to capture dynamic and time-varying relations between entities (see Figure1). As far as we know, [26] is the only attempt until now focusing on detecting relations in videos, while it shows limited performance partly for its deficiency in aggregating cues from the surrounding context. In Figure1, intuitively we can infer that message communication between entities would benefit video relation prediction, e.g., in spatial dimension, knowing <*bicycle, move_right, car*> would help to detect <*person, right, car*>, also in temporal dimension, knowing <*car, faster, bicycle*> in the current segment would increase the chance to predict <*car, move_past, bicycle*> in the next segment.

In this paper, we model the set of entities and their relationships in a video into a fully-connected spatio-temporal graph, which

MM '19, October 21–25, 2019, Nice, France

ACM ISBN 978-1-4503-6889-6/19/10…$15.00
https://doi.org/10.1145/3343031.3351058

*The corresponding author is Yueting Zhuang.

includes entity nodes in the neighborhood in the time and space dimensions, as the Figure1 shows. For relations detection, we propose a novel model named Video Relation Detection Graph Convolution Network (VRD-GCN) to aggregate information from context and conduct reasoning in this 3D graph. By capturing the relative variation in the geometry and appearance of entities in the space-time dimension, VRD-GCN is enabled to detect dynamic relationships between entities. Similarly, by passing message from neighbors and context in our spatio-temporal graph to the target entities, VRD-GCN is able to produce more accurate and complete detection results. Moreover, we utilize the geometric overlap and appearance relevance jointly to indicate how strongly are entities related in the graph. This design helps our model make full use of the relative characteristics of entities and facilitate the information communication between them in multi-channel.

After relation instances in segments detected, relation association will be applied to merge the short-term relation instances in entire videos. [26] proposed a greedy association method which repeatedly associates the short-term relation instances according to the geometric overlaps and the relation triplets. However, since the change of scene and the occurrence of the drifting problem (e.g. the proposals of an entity's trajectories in adjacent segments drift towards opposite directions), it is insufficient to determine whether two trajectories in consecutive segments belong to the same entity merely relying on the geometric overlap. To solve this issue, we propose a novel online association method with a siamese network[3] which jointly take appearance similarity and geometric overlaps of relation instance into account. Our association method is more accurate and robust with the help of the multi-aspects similarity comparison and the effectiveness of the siamese network. In addition, we upgrade our association algorithm with the online mechanism to facilitate the realization of online video relation detection.

We evaluate the effectiveness of our method on benchmark ImageNet-VidVRD dataset[26]. By combining VRD-GCN and online association with the siamese network, our architecture achieves the state-of-the-art performance on two standard tasks: video relation detection and video relation tagging (see Table5).

Our main contributions are threefold:

(1) we abstract videos into fully-connected spatio-temporal graphs and propose a novel video relation detection model named VRD-GCN to pass message and conduct reasoning in these graphs.
(2) we propose an online association method with a siamese network to accurately and effectively associate short-term relation instances.
(3) we conduct a range of relation detection experiments on ImageNet-VidVRD dataset. Our method outperforms the VidVRD baseline and other competing models by a large margin and achieves the state-of-the-art performance under all metrics in both detection and tagging tasks.

2 RELATED WORK

Visual Relation Detection. Plenty of recent works have been proposed for relation detection on static images. [17] trained visual models for objects and predicates respectively and combined them to jointly predict relationships. They also introduced a language module to cast relationships into a semantic space using pre-trained word embeddings to improve the performance. [42] interpreted the relation triplet as a vector translation by mapping the features of objects and predicates into a low-dimensional space. [35] argued that separated predictions of objects and relationships can benefit from the surrounding context and generated scene graph by iteratively passing message between nodes and edges. [20] combined appearance and spatial feature of a pair of objects and trained a set of weakly-supervised classifiers to detect relations. [41] further employed two bidirectional LSTM, one to encode the global context across bounding regions and another to compute and propagate information for predicting edges in the condition of previously predicted object labels and all other computed context. [37] used relatedness to prune unlikely connections in a fully connected dense graph and applied an attentional graph convolutional network (aGCN) to propagate higher-order context throughout the graph. While there have been numerous works on images, video relation detection is less explored mainly because of the lack of annotated video dataset. [26] contributed the first video relation detection dataset which contains rich labeled relations. They also proposed a VidVRD method on this dataset which proposed a effective framework to solve video relation detection and utilized motion feature of entities to predict dynamic relationships.

Graph Neural Networks. Graph neural networks (GNNs) have been widely applied in structural scenarios where the data are naturally represented in graph structure [7, 11, 15, 21, 30, 39] and also non-structural data like images and texts [1, 9, 14, 18, 19, 25, 28]. Due to the ability of reasoning in graphs, GNNs are good at dealing with data containing rich interactions among elements. Recently, the Graph Convolutional Network (GCN) [15] was proposed for natural language processing and revealed the effectiveness for learning on graph-structured data. [37] applied an attentional graph convolutional network (aGCN) to propagate higher-order context throughout the scene graph for relation inference. To get a better understanding of videos, [33] represented videos as space-time region graphs and employed graph convolutions to capture spatial-temporal relations between objects.

Object Tracking. Visual object tracking aims to estimate the location of one target (Single Object Tracking, SOT) or several targets (Multiple Object Tracking, MOT) in each frame of videos. Object tracking can be modeled as a similarity learning problem. [5] learned a discriminative correlation filter to localize the target in a new frame. [8] proposed a fast scale estimation approach by learning separate filters for translation and scale. In order to compare with [26] fairly, our architecture adopts [8] for object tracking. As for similarity matching in object tracking, [4] only took the bounding box position and size into consideration. Furthermore, [34] integrated appearance information to improve the performance of [4]. A fully-convolutional siamese architecture [3] for SOT used a convolutional embedding to map the sampler image and searched image into a new space and then obtained the similarity using a cross-relation layer. Also, siamese architectures can be applied to address the similarity learning problem [3, 12, 29]. We adopt a siamese network to perform relation association and achieve better performance compared to the association method in [26].

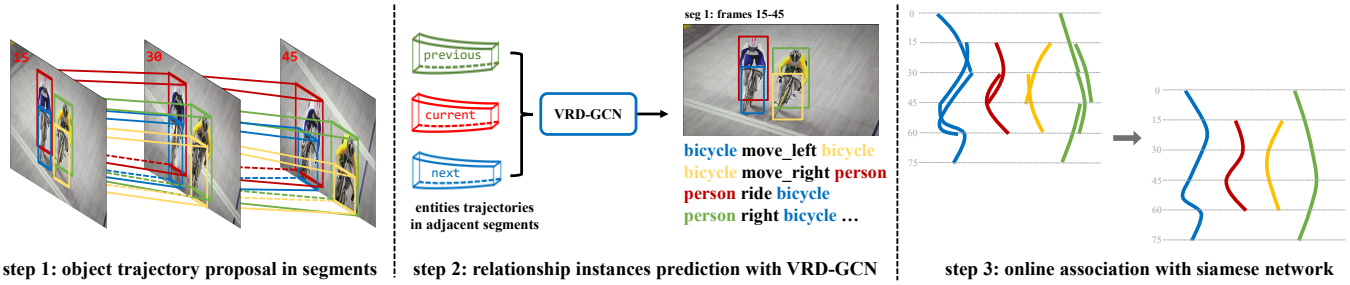

Figure 2: An overview of our method. VidVRD task can naturally be decomposed into three independent parts: multi-object tracking, relation prediction, and relation instances association.

3 VIDEO RELATION DETECTION

The definition of VidVRD task[26] is: given a object set O, a predicate set $\mathcal{P}$, and videos with arbitrary lengths, we are required to detect all the visual relation instances of interest represented by *<subject, predicate, object>* $\in O \times \mathcal{P} \times O$ with the subject trajectory $\mathcal{T}_s$ and object trajectory $\mathcal{T}_o$, where $\mathcal{T}_s$ and $\mathcal{T}_o$ are two sequences of bounding boxes of subject and object respectively. As Figure2 shows, we can naturally decompose the VidVRD task into three independent parts: multi-object tracking, relation prediction, and relation instances association. We first split videos into segments and extract trajectory proposals from each segment. Then, we apply our VRD-GCN to predict relationships of all pairs of entities. Finally, we use the online association with a siamese network to associate short-term relation instances detected in the second step. We adopt the object trajectory results from [26] as well as their preprocessing pipeline in step 1 for a fair comparison.

3.1 VRD-GCN for Segment Relation Prediction

Intuitively, not only the entities close to each other in space keep strong interconnection, but the entities in the adjacent time periods also keep strong relations. Therefore, we abstract a video into a fully-connected spatio-temporal graph, where each entity is regarded as a trajectory node connected to all other nodes in previous, current and next segments(see Figure1). Graph Convolutional Network (GCN)[15], which is a non-Euclidean connectionist data struct, becomes our natural choice to aggregate information and perform reasoning in this spatio-temporal graph.

Input of spatio-temporal GCN. Inputting all segments of video into the net is obviously unnecessary because of information-redundancy and resource-intension. We did experiments by taking more segments as input while them showed worse performance, which indicated that the information from far-away segments may disturb the relation prediction. Therefore, we only utilize features of trajectories from three adjacent segments (including the previous, current, and next one). To further reduce the computational complexity and noticing that the communication of messages between entities in the previous and next segment contributes little to the relation prediction in the current segment, we decompose the affinity matrix to two parts: one for entities in the previous segment and the current one, and another for entities in the next segment and the current one. Size of each affinity matrix is $(2N)^2$ (N is the number of trajectories in one segment after NMS) and computation can be performed parallelly on these two branches, as Figure3 shows. We concatenate features from adjacent segments and feed them into full-connected spatio-temporal graph convolutional module (ST-GCN).

Spatio-Temporal GCN. Graph convolutional network (GCN)[15] is effective for reasoning in graph, not only for GCN's shared weight mechanism which reduces the computational complexity but also for that multi-layer struct can be performed with GCN to pass information to nodes from its neighbors with an arbitrary depth in graph. One layer of basic GCN can be formulated as

$$\mathbf{X}^{'} = \sigma(\mathbf{AXW}) \tag{1}$$

where $\mathbf{W} \in \mathbb{R}^{d \times d}$ is the adaptive parameters and $\mathbf{X} \in \mathbb{R}^{2N \times d}$ is the features of entities input to ST-GCNs. $\mathbf{X}^{'}$ is the output in the same size as $\mathbf{X}$. σ is a non-linear activation function. $\mathbf{A} \in \mathbb{R}^{2N \times 2N}$ is the affinity matrix. Generally, $\mathbf{A}$ is an adjacent matrix or similarity matrix of the graph.

In our spatio-temporal graph, the hidden state of a target entity is supposed to absorb more information from the entities correlated with it, for example, the entities having large spatial-temporal geometric overlap or the entities having great appearance relevance. To this end, we set our affinity matrix into two forms, one is the vIoU matrix of trajectories (vIoU is the voluminal intersection over union of two trajectories), and another is the appearance relevance matrix of entities, as visualized in Figure3(b). We integrate the geometry GCN and the appearance GCN into our ST-GCN which correspond to the two forms of affinity matrixes respectively. Furthermore, we add the residual learning block to alleviate degradation problem when stacking layers of ST-GCN and make the model more robust. The output of appearance GCN $\mathbf{X^a}$ and geometry GCN $\mathbf{X^g}$ would be added together with the original features $\mathbf{X}$ by element-wise as Eq.2. Then, ReLU activation and normalization are applied before the output of this ST-GCN layer be fed into the next one to keep the scale of the input and output consistent.

$$\mathbf{X}^{'} = norm(\sigma(\mathbf{X^a} + \mathbf{X} + \mathbf{X^g})) \tag{2}$$

In geometry GCN, we use vIoU as values in affinity matrix indicating how strongly two objects are related in geometry. If two trajectories are in different segments, only the part overlapped in temporal dimension are put into the calculation. vIoU matrix are then normalized by Manhattan norm in each row of it in Eq.4. Also, after acquiring the output of geometry GCN $\mathbf{X^g}$, we apply first

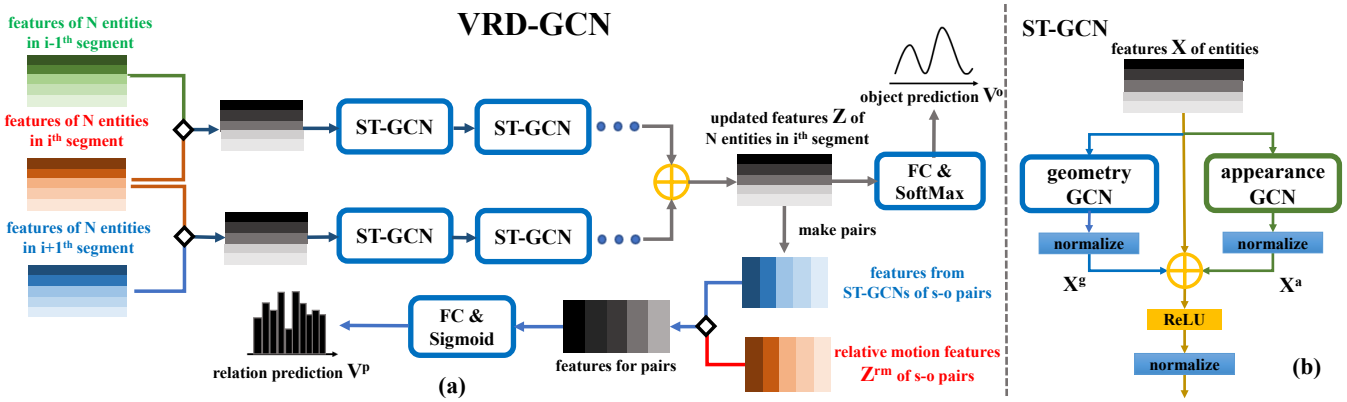

Figure 3: (a) Overview of our VRD-GCN model. ⋄ donates concatenation and ⊕ donates element-wise addition (b) the spatial-temporal graph convolutional net module (ST-GCN)

ReLU activation and then Layer-Norm[2] to $\mathbf{X^g}$ to keep the scale of the input and output of geometric GCN consistent as Eq.3.

$$\mathbf{X^g} = norm(\sigma(\mathbf{A^g X W^g})) \tag{3}$$

$$A^g_{ij} = \frac{vIoU(\mathcal{T}_i, \mathcal{T}_j)}{\sum_{j=0}^{N-1}(vIoU(\mathcal{T}_i, \mathcal{T}_j))} \tag{4}$$

In appearance GCN, we set the appearance relevance matrix as Eq.6. We first apply two different linear transformations to entities' features to map them into a new space, then multiply them to obtain the appearance relevance. The learned appearance relevance matrix is able to connect the related objects in the same or adjacent segments even there is no geometric overlap between them. Visualized appearance relevance matrix before normalization in Figure4 proves that our design works well. We normalize the affinity matrix by applying a softmax layer to each row to rescale the affinity value to be more reasonable as Eq.6. As the same as geometry GCN, after obtaining the output features map $\mathbf{X^a}$ from appearance GCN, we apply ReLU activation and Layer-Norm to it as Eq.5.

$$\mathbf{X^a} = norm(\sigma(\mathbf{A^a X W^a})) \tag{5}$$

$$A^a_{ij} = \frac{exp(\phi(\mathbf{X_i})^T \phi'(\mathbf{X_j}))}{\sum_{j=0}^{N-1}(exp(\phi(\mathbf{X_i})^T \phi'(\mathbf{X_j})))} \tag{6}$$

By stacking ST-GCN layers, we promote information exchange between entity nodes in large distance, and the performance of model improved as response (e.g. entity A has strong relevance with entity B and entity C, while B is correlated weakly with C, multi-layer ST-GCNs would help promote the message communication between B and C).

Prediction of object and predicate. As Figure3(a) shows, we add the outputs of two branches by element-wise and extract out the feature map $\mathbf{Z} \in \mathbb{R}^{N \times d}$ belonging to entities in the current segment. Features $\mathbf{Z}$ from ST-GCNs of N entity nodes are then used to predict the categories of objects and the distribution of predicates. In one way, $\mathbf{Z}$ would be fed into a linear transformation layer ϕ^o and a softmax layer to get the prediction vector $\mathbf{V^o}$ for object classification as Eq.7. In the other, as Eq.8, each two feature vectors in $\mathbf{Z}$ would be paired to compose a new feature map of *<subject, object>* pair with dimension $(N \times (N-1), 2d)$. Then the feature map of s-o pairs would be concatenated with the relative motion feature $\mathbf{Z^{rm}} \in \mathbb{R}^{(N \times (N-1)) \times d'}$ [26] to form the final feature map of s-o pairs in shape $((N \times (N-1)), 2d + d')$. Finally, the final feature map would be fed into a linear transformation layer ϕ^p and a sigmoid layer to predict predicates distributions vector $\mathbf{V^p}$.

Because the number of combinations of *<subject, predicate, object>* is tremendous making triplet prediction unrealistic, and triplet would not be predicted if it did not appear in the train set, we predict subject, predicate, and object separately. In prediction, we multiply the confidence score in $\mathbf{V^o}$ and $\mathbf{V^p}$ of the triplet *<subject, predicate, object>* for each relation instance. After it, we keep top-n relation instances in one segment and then feed them into the association module.

$$\mathbf{V^o_i} = softmax(\phi^o(\mathbf{Z_i})) \quad (i \in [1, N]) \tag{7}$$

$$\mathbf{V^p_{ij}} = sigmoid(\phi^p(\mathbf{Z_i}||\mathbf{Z_j}||\mathbf{Z^{rm}_{ij}})) \quad (i, j \in [1, N] \; and \; i \neq j) \tag{8}$$

Loss function of VRD-GCN. In training, we use single classification and cross entropy loss function (CE) to calculate the loss between object prediction vector V^o and object category label C^o_{gt} in object classification and use multilabel classification and binary cross entropy loss function (BCE) to calculate the loss between predicate prediction vector V^p and relation vector in gound-truth V^p_{gt} for predicate prediction. To balance the scale of two losses for object classification and predicate prediction, we multiply two losses by weight W_o and W_p respectively and then add them together according to Eq.9.

$$Loss_{rel} = W_o * CE(\mathbf{V^o}, \mathbf{C^o_{gt}}) + W_p * BCE(\mathbf{V^p}, \mathbf{V^p_{gt}}) \tag{9}$$

3.2 Online Association with Siamese Network.

After obtaining the triplet prediction of all the entity pairs in t^{th} segment, we adopt an online association algorithmwith a siamese

Algorithm 1 Online Relational Association via Siamese Network

Input:
the set of detected long-term relation instances before the current segment $\mathcal{S} = \{(c, < s, p, o >, (\mathcal{T}_s, \mathcal{T}_o))\}$;
the set of detected short-term relation instances in the current segment (t^{th}) $\mathcal{A} = \{(\hat{c}, < \hat{s}, \hat{p}, \hat{o} >, (\hat{\mathcal{T}_s}, \hat{\mathcal{T}_o}))\}$;
Output:
the set of merged instances $\mathcal{L}$;
Initialize:
$\mathcal{L} = \mathcal{S}$, y = threshold;
$\mathcal{B}$ = instances in $\mathcal{S}$ that end at the $(t-1)^{th}$ segment;
Descending sort $\mathcal{A}$ according to $\hat{c}$;
Descending sort $\mathcal{B}$ according to c;
for $(c, < s, p, o >, (\mathcal{T}_s, \mathcal{T}_o))$ in $\mathcal{B}$ **do**
  for $(\hat{c}, < \hat{s}, \hat{p}, \hat{o} >, (\hat{\mathcal{T}_s}, \hat{\mathcal{T}_o}))$ in $\mathcal{A}$ **do**
    Compute the confidence score $S_{asso}(\mathcal{T}_s, \hat{\mathcal{T}_s})$ and $S_{asso}(\mathcal{T}_o, \hat{\mathcal{T}_o})$ using Eq.11;
    if $< s, p, o > = < \hat{s}, \hat{p}, \hat{o} >$ AND $S_{asso}(\mathcal{T}_s, \hat{\mathcal{T}_s}) > y$ AND $S_{asso}(\mathcal{T}_o, \hat{\mathcal{T}_o}) > y$ **then**
      Append $(\hat{\mathcal{T}_s}, \hat{\mathcal{T}_o})$ to $(\mathcal{T}_s, \mathcal{T}_o)$
      Recompute c using Eq.12
      Update $(c, < s, p, o >, (\mathcal{T}_s, \mathcal{T}_o))$ in $\mathcal{L}$
      Remove $(\hat{c}, < \hat{s}, \hat{p}, \hat{o} >, (\hat{\mathcal{T}_s}, \hat{\mathcal{T}_o}))$ from $\mathcal{A}$
      Break
    end if
  end for
end for
for $(\hat{c}, < \hat{s}, \hat{p}, \hat{o} >, (\hat{\mathcal{T}_s}, \hat{\mathcal{T}_o}))$ in $\mathcal{A}$ **do**
  Add $(\hat{c}, < \hat{s}, \hat{p}, \hat{o} >, (\hat{\mathcal{T}_s}, \hat{\mathcal{T}_o}))$ to $\mathcal{L}$;
end for

network (as shown in Algorithm1) to associate the short-term relation instances $\mathcal{A} = (\hat{c}, < \hat{s}, \hat{p}, \hat{o} >, (\hat{\mathcal{T}_s}, \hat{\mathcal{T}_o}))$ in current segment and the long-term relation instances $\mathcal{S} = (c, < s, p, o >, (\mathcal{T}_s, \mathcal{T}_o))$ in the video clip before the current segment. (where $< s, p, o >$ is the triplet prediction of a relation instance, c is the confidence score, and $(\mathcal{T}_s, \mathcal{T}_o)$ is the trajectories of subject and object respectively.)

The greedy association proposed by [26] take the whole set of relation instances in videos as input and associate two trajectories only when they have a higher vIoU than the vIoU threshold. The greedy association merely utilized the geometry information which works well only when trajectory proposals are very accurate. However, the drifting problem always occurs among segments ascribed to the deficiency of object tracking algorithm (see Figure5), which badly influences the performance of the greedy association algorithm. To overcome this issue, we advocate an alternative approach in which a siamese network are trained to implement online association. Specifically, we feed the feature vectors of two subjects or two objects in adjacent segments from VRD-GCN into an embedding net. Then, we evaluate the confidence score α of appearance similarity of these two entities by applying a similarity function (we use cosine similarity in experiments) to the embedded features as Eq.10. Limited by the very few cases for training and to prevent model overfitting, we merely use 3 layers of linear transformations as the embedding net to reduce the number of parameters. To take both the geometry information and the appearance information into consideration, vIoU and the confidence score α are added together after multiplied by the corresponding weight W_g and W_a to produce the final confidence score $S_{asso}(\mathcal{T}, \mathcal{T}^{'})$ for association as Eq.11. ($\mathcal{T}$ and $\mathcal{T}^{'}$ are arbitrary two trajectories in consecutive segments)

According to Algorithm1, we associate $\mathcal{A}$ with the sorted long-term relation instances $\mathcal{S}$ with the online mechanism. Only when the triplets of two relation instances are identical and the confidence score for the association is larger than the threshold will them be associated. In addition, The bounding-boxes in the overlapping part of two associated trajectories are averaged to obtain more accurate trajectories. The score c_p of long-term relation instance $p = \{(c^t, < s, p, o >, (\mathcal{T}_s^t, \mathcal{T}_o^t))\}$ ($t \in [m, n]$) from m^{th} to n^{th} segments is then updated to the highest score of all short-term relation instances in p as Eq.12.

$$\alpha(\mathcal{T}, \mathcal{T}^{'}) = \varphi(emb(f_{\mathcal{T}}), emb(f_{\mathcal{T}^{'}})) \tag{10}$$

$$S_{asso}(\mathcal{T}, \mathcal{T}^{'}) = W_g * vIoU(\mathcal{T}, \mathcal{T}^{'}) + W_a * \alpha(\mathcal{T}, \mathcal{T}^{'}) \tag{11}$$

$$c_p = max(c^t) \quad (t \in [m, n]) \tag{12}$$

In real applications, our algorithm would take relation instances $\mathcal{A}$ in one segment of a video as input at a time and process the whole video iteratively. We would extract out associated long-term relation instances in $\mathcal{L}$ which contain relation instances in the current segment and store other instances to disk. For the instances extracted, we sort them in descending order according to score c and retain the top-n instances for the next iteration.

Loss function of siamese network. To train siamese network, we use loss function as Eq.13 where $\alpha(\mathcal{T}, \mathcal{T}^{'})$ is as Eq.10 and $\delta(\mathcal{T}, \mathcal{T}^{'})$ is 1.0 when $\mathcal{T}$ and $\mathcal{T}^{'}$ is labeled the same trajectory in adjacent segments otherwise 0.0.

$$Loss_{sia} = log(1.0 + exp(\delta(\mathcal{T}, \mathcal{T}^{'}) * \alpha(\mathcal{T}, \mathcal{T}^{'}))) \tag{13}$$

4 EXPERIMENTS

Dataset. We evaluated our method on the newly released benchmark: ImageNet-VidVRD dataset[26]. This dataset consists of 1000 videos collected from ILSVRC2016-VID[23], which are well-labeled with object categories and corresponding trajectories. The visual relations are tagged under 35 categories of objects and 132 categories of predicates, noted as *<subject, predicate, object>*. The categories of predicates can be divided into relative spatial positions, actions and actions' adjectives and relation instances with different kinds of predicates could be tagged overlapping with each other both spatially and temporally. Above all, it has 4835 instances from 3219 visual relation triplets types and 9.5 relation instances per segments on average. The released split uses 80% of videos for training and 20% for testing.

Setting. As the conventions in [26], we evaluate our method on two standard tasks: **Relation detection** and **Relation tagging**. The detection task takes a video as input to output a set of relation triplets with localized objects. A relation triplet is considered to be correct if there is the same relation triplet tagged in the ground truth and both trajectories of its subject and object must have sufficient

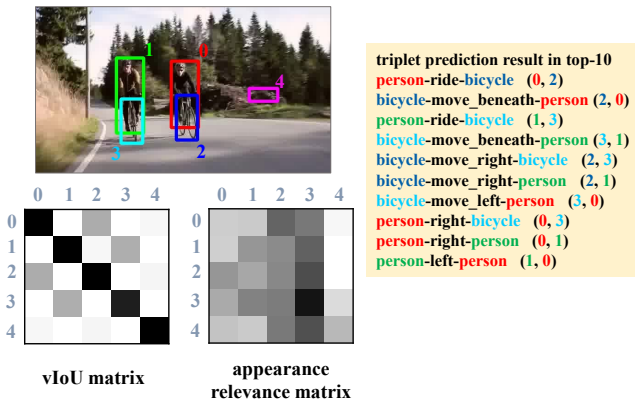

Figure 4: Visualization of the fixed vIoU matrix and the learned appearance relevance matrix of one segment extracted out before softmax in ST-GCN. A darker grid represents a larger matrix value. The value of a grid (m, n) represent the vIoU or appearance relevance of m^{th} and n^{th} entities.

vIoU. In experiments, the overlapping threshold of vIoU is set to 0.5. The tagging task reduces the influence of objects localization, the output of which is a set of video relation triplets annotated to the whole video without the localization of object. As many detection tasks, we use mean average precision (mAP) and Recall@K (K is set to 50 and 100) as our VidVRD detection evaluation metrics. We can measure the fraction of the positive detection in the top K results from Recall@K and the overall precision performance at different recall value from mAP metric. For tagging task, we use Precision@K (K is set to 1, 5 and 10) as the evaluation metric following [26]'s evaluation setting to measure the accuracy of the tagging results.

4.1 Implementation Details.

Video segmentation, trajectory proposals, and features. Since our main contributions lie in VRD-GCN module and online association algorithm, we adopt the object trajectory results from [26] as well as their preprocessing pipeline for a fair comparison. We treat videos as consecutive overlapping (15 frames overlapped) segments with 30-frames without sampling. The objects trajectories are also adopted into these segments. A fine-tuned Faster-RCNN [22] with ResNet101 [13] with tagged 35 object categories from MS-COCO and ILSVRC2016 datasets is used as object detector to produce bounding boxes at each frame. Then the implementation of [8] in Dlib linked frame-level bounding boxes across the segment. Besides, it is found that the number of trajectories per segment accords with long-tailed distributions, including the trajectories in dataset and detected by our tracker (e.g. the number of trajectories in 97.02% segments in ImageNet-VidVRD is no more than 5). Therefore, we use non-maximum suppression (NMS) with $vIoUthreshold = 0.5$ to reduce the similar trajectories and use top N (N is set to 5) as the objects trajectories input for VRD-GCN modules. Following Sec.3 notation, we express the input of VRD-GCN model X as a matrix consisting of $2 * 2N$ trajectory features. Each trajectory feature is chosen to be a concatenation of the improved dense trajectory (iDT)[31] feature and the classeme feature as [26] did. The relative spatial-temporal positions between the trajectories are also calculated to predict predicate categories. It is noteworthy that these kinds of features were chosen for the fair-comparison while our model is compatible with any other features.

Table 1: Evaluation for our method using geometry or appearance GCN branches on ImageNet-VidVRD dataset

branch	relation detection			relation tagging		
	R@50	R@100	mAP	P@1	P@5	P@10
appearance	6.14	7.18	12.78	52.00	37.90	26.05
IoU	7.84	9.27	15.51	55.00	39.30	27.00
IoU + appearance	**8.07**	**9.33**	**16.26**	**57.50**	**41.00**	**28.50**

Train details. Our framework utilized a two-stage training strategy. We first trained our VRD-GCN modules to achieve expected performance on relation detection task in segments without association. We use loss function mentioned in Sec.3 to optimize the performance in predictions of object and predicate and we set the weight of object classification loss W_o to 1.0 and the weight of predicate prediction loss W_p to 25.0 respectively. The batch size and initial learning rate are set to 5 and 10^{-3}. After VRD-GCN module had converged, we used trajectory GCN features produced as the inputs to train siamese network. Since the online association algorithm only links bounding boxes across consecutive segments, we sample positive training pairs from training video segments if the two trajectories in consecutive segments both have a higher vIoU with the same ground truth trajectory over 0.5, if don't then negative pairs. 2311 positive pairs and 3441 negative pairs are sampled from the dataset. As many MOT[4, 34] pipelines designed, we adopt batches with a balanced ratio of 1:3 (number of positive pairs : number of negative pairs) and online hard negative data mining[27] in training process. The batch size and the initial learning rate are set to 256 and 10^{-4}.

4.2 Ablation Studies

We run a number of ablations to figure the effectiveness of our VRD-GCN modules and online association algorithm. For VRD-GCN modules, we focus on how different kinds of affinity matrix in ST-GCN unit and the number of GCN layers can influence the performance on both relation detection and tagging tasks. For the online association algorithm, we validate its performance for successful linking trajectories of the same object in two consecutive segments in detail.

VRD-GCN modules. In our proposed spatio-temporal GCN unit, we designed two kinds of affinity matrix: vIoU matrix and appearance relevance matrix. Literally, vIoU matrix indicates how two objects are correlated in geometric position and appearance relevance matrix indicate the intrinsic relevance of two objects. We investigated how strong these two GCN branches can influence the performance and how they cooperated with the other. The results are shown in Table1. We can observe that the geometric branch performed better than appearance branch since relative geometric information is much more important in describing relations about relative motions and positions. However, as shown in Figure4, appearance branch is also important since while the fixed vIoU matrix

Table 2: Evaluation for our method with different number of ST-GCN layers on ImageNet-VidVRD dataset

number of layers	relation detection			relation tagging		
	R@50	R@100	mAP	P@1	P@5	P@10
0	4.30	5.36	8.58	34.50	21.80	15.65
1	6.23	7.65	13.22	46.50	32.40	23.40
2	7.01	8.09	14.58	**61.00**	38.70	26.90
3	**8.07**	**9.33**	**16.26**	57.50	**41.00**	**28.50**
4	6.97	8.17	14.87	59.00	39.60	28.35

passes message between overlapped objects, the learned appearance relevance matrix is enabled to connect the related objects even without overlap. And model shows the best result when appearance GCN and geometry GCN are joined, which has 0.75% improvements in detection task (mAP) than only using geometry GCN.

Table2 shows the results of experiments on our multi-layer architecture. It shows when the number of layers is smaller than 4, our architecture can be benefited with the complexity growth and achieves the best performance generally on both detection and tagging task at 3 layers. When the number of layers added up to 4, our architecture suffers from overfitting problem with performance dropping. In other comparison experiments, we set our VRD-GCN architecture to 3 layers by default.

Association algorithm. Our proposed association algorithm using vIoU strategy with similarity confidence scores obtained via a trained siamese network. We explored the effectiveness of our association algorithm by experiments on separated association task, which measures the accuracy of linking the object trajectories in consecutive segments. We also compared our trained siamese network confidence score with naive cosine similarity with GCN trajectory features. The results are shown in Table3. We can observe that the accuracy improve 34.44% after using siamese instead of cosine similarity, which indicates our motivation that the feature trained from GCN with the object and predicate losses is not capable for association. This is because the direct GCN features were trained for semantic classification tasks, which needed to be transformed into a new space for similarity comparison. We also observed that after NMS approach, our vIoU strategy can achieve a good association accuracy of 93.74%, and achieve 95.32% with the help of the siamese network. It proves our motivation that by taking similarity confidence scores into the algorithm, the influence of geometric drifting problem can be reduced, as Figure5.

4.3 Comparisons with State-of-The-Arts

Comparison methods. We compare the performance of our method with other five state-of-the-art methods: Visual Phrase (VP) [24], Lu's-V [17], Lu's [17], VTransE [42], and VidVRD [26]. Since the first four methods focus on image relation detection, they tend to neglect dynamic relationship in videos. For the sake of fairness, first, we use Faster R-CNN in bounding-boxes extraction and DSST in trajectories extraction for all these methods. Also, features extracted in our method are used in all these methods for relation prediction. Second, we keep 20 relation triplets predictions for each entity pair with top confidence scores in all these methods. Third, traditional greedy association method is used in all these methods

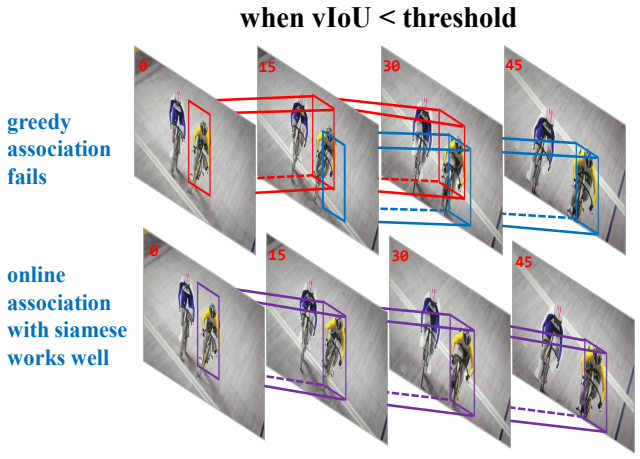

Figure 5: When two trajectories of one entity in adjacent segments have insufficient overlap, our online association with siamese still works well while the greedy association fails

Table 3: Evaluation for different association methods on ImageNet-VidVRD dataset

	IoU	gcn feature		
		cosine similarity	siamese	siamese + IoU
accuracy	93.74	47.51	81.95	**95.32**

for association. Particularly, we use greedy association or online association with siamese net in VRD-GCN to demonstrate the superiority of our online association method.

Because the main improvement made by our architecture lies in the short-term relation instance detection in single segment, we make additional comparisons on relation detection modules without association between baseline VidVRD and our model VRD-GCN in a fine-grained manner. Besides, since we utilize a two-stage training strategy, we use the metrics in this experiment as the sign of whether VRD-GCN has converged in the first stage.

Comparison of relation detection module. The quantitative results of VidVRD and our VRD-GCN in segment without association are reported in Table4. We can observe that compared to VidVRD our model improves the mAP in relation detection by 2.6%. The reason why VRD-GCN can significantly improve the performance of relation instance detection is that VRD-GCN is superior in both respects of object classification and relation prediction. VRD-GCN outperforms VidVRD in recall@20 of predicate prediction by 17.91% and simultaneously improve the object-precision to 81.03%.

Comparison of the entire model. The quantitative results are reported in Table5. We can observe that our architecture (VRD-GCN + online association with siamese network) outperforms all the competing methods by a large margin under all evaluation metrics.

First, comparing VidVRD baseline and our method with others, we find that the model which can capture dynamic relationships between entities show a better performance in video relation detection. Second, our VRD-GCN give better relationships recognition results than VidVRD even with the same association module, which

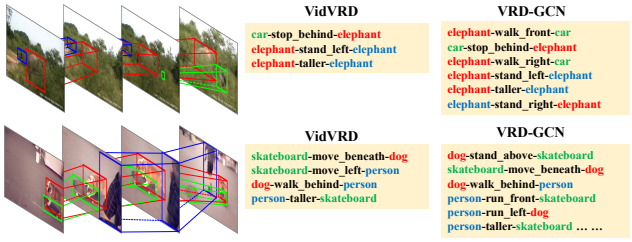

Figure 6: Visualization of video relation detection results using VidVRD baseline and our model VRD-GCN. The correct relation instances in the top-20 results are shown.

Table 4: Evaluation using VidVRD baseline and VRD-GCN in segments (without association) on benchmark ImageNet-VidVRD dataset.

Method	relation detection			relation tagging		
	R@50	R@100	mAP	P@1	P@5	P@10
VidVRD	19.80	23.18	16.50	28.15	20.06	14.86
VRD-GCN (Ours)	**21.90**	**25.84**	**19.10**	**35.70**	**24.89**	**17.82**
Method	subject and object			predicate		
	Acc@1			R@20	R@50	R@100
VidVRD	70.67			18.98	27.28	34.43
VRD-GCN (Ours)	**81.03**			**36.89**	**46.72**	**53.13**

Table 5: Evaluation of different methods on standard video relation detection and video relation tagging. "siamese" donates using online association with siamese network while the greedy association is applied to other models. "gt" donates using trajectories of objects in ground truth.

Standard	relation detection			relation tagging		
Methods	R@50	R@100	mAP	P@1	P@5	P@10
VP	0.89	1.41	1.01	36.50	25.55	19.20
Lu's-V	0.99	1.80	2.37	20.00	12.60	9.55
Lu's	1.10	2.23	2.40	20.50	16.30	14.05
VTransE	0.72	1.45	1.23	15.00	10.00	7.65
VidVRD	5.54	6.37	8.58	43.00	28.90	20.80
VRD-GCN (Ours)	7.43	8.75	14.23	**59.50**	40.50	27.85
VRD-GCN+siamese (Ours)	**8.07**	**9.33**	**16.26**	57.50	**41.00**	**28.50**
VidVRD gt	12.51	16.55	15.53	43.50	29.70	23.20
VRD-GCN gt (Ours)	17.50	21.80	26.52	62.50	44.20	31.10
VRD-GCN+siamese gt (Ours)	**18.28**	**22.39**	**27.87**	**63.00**	**45.80**	**32.70**

indicates that the information passed from context and neighbors entities helps a lot. Third, after adding the new association module, our method gains 2.03% absolute improvement in mAP, which demonstrates the effectiveness of our association module. Finally, given trajectories in ground-truth (see the last rows in Table5), our model also shows great superiority compared with other methods.

Besides the standard evaluation, we also evaluate methods on the zero-shot setting. It can be observed from Table6 that our methods also achieve the state-of-the-art performance here. For a trajectory, the variance of triplet score distribution from our model's prediction is smaller than that from other models'. In zero-shot setting, when trajectories' correspondence is not required, especially when the

Figure 7: Cases of the top-5 result of video relation tagging via VidVRD baseline and our model VRD-GCN.

Table 6: Evaluation of different methods on zero-shot video relation detection and video relation tagging.

Zero-shot	relation detection			relation tagging		
Method	R@50	R@100	mAP	P@1	P@5	P@10
Lu's-V [22]	0.93	0.93	0.40	2.74	0.82	0.82
Lu's [22]	0.69	1.16	0.47	1.37	1.37	1.23
VTransE [42]	0.69	0.69	0.03	1.37	1.37	0.96
VidVRD	1.62	2.08	0.40	**4.11**	**1.92**	**1.92**
VRD-GCN (Ours)	3.94	7.18	0.67	**4.11**	1.11	1.37
VRD-GCN + siamese (Ours)	**4.63**	**7.64**	**0.74**	**4.11**	1.11	1.23

number of zero-shot triplets are few in test set and the values under all metrics in zero-shot are small, other baselines are more likely to hit the gound-truth by chance. But our model performs better when the trajectories' correspondence is required.

Qualitative results. From the qualitative results compared with baseline VidVRD of relation detection in Figure6 and relation tagging in Figure7 respectively, we can see that benefited from the information passed from the neighbors, our method produces better results. E.g. in the last row in Figure6, by knowing *<skateboard, move_beneath, dog>* and *<dog, walk_behind, person>*, our method detects instance *<person, run_front, skateboard>* successfully while VidVRD baseline fails here.

5 CONCLUSIONS

We proposed a novel video relation detection model VRD-GCN and an online association method with siamese net. VRD-GCN improves the accuracy of relationship detection significantly and online association with siamese net addresses the inherent drifting problems among trajectories in adjacent segments. By combining these two methods together, our architecture outperforms the state-of-the-art by a large margin and promote the development of this field greatly. Moving forward, we are going to 1) design a more effective end-to-end model which integrates these modules above together. 2) improve the efficiency of the model in inference to realize real-time video relation detection.

6 ACKNOWLEDGMENTS

This work was supported by National Key Research and Development Program of China (SQ2018AAA010010), Zhejiang Natural Science Foundation (LR19F020002, LZ17F020001), National Natural Science Foundation of China (6197020369, 61572431), the Fundamental Research Funds for the Central Universities and Chinese Knowledge Center for Engineering Sciences and Technology.

REFERENCES

[1] Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. 2015. Deep Compositional Question Answering with Neural Module Networks. *CoRR* abs/1511.02799 (2015).

[2] Lei Jimmy Ba, Ryan Kiros, and Geoffrey E. Hinton. 2016. Layer Normalization. *CoRR* abs/1607.06450 (2016).

[3] Luca Bertinetto, Jack Valmadre, João F. Henriques, Andrea Vedaldi, and Philip H. S. Torr. 2016. Fully-Convolutional Siamese Networks for Object Tracking. In *Computer Vision - ECCV 2016 Workshops - Amsterdam, The Netherlands, October 8-10 and 15-16, 2016, Proceedings, Part II*. 850–865.

[4] Alex Bewley, ZongYuan Ge, Lionel Ott, Fabio Tozeto Ramos, and Ben Upcroft. 2016. Simple online and realtime tracking. In *2016 IEEE International Conference on Image Processing, ICIP 2016, Phoenix, AZ, USA, September 25-28, 2016*. 3464–3468.

[5] David S. Bolme, J. Ross Beveridge, Bruce A. Draper, and Yui Man Lui. 2010. Visual object tracking using adaptive correlation filters. In *The Twenty-Third IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2010, San Francisco, CA, USA, 13-18 June 2010*. 2544–2550.

[6] Long Chen, Hanwang Zhang, Jun Xiao, Xiangnan He, Shiliang Pu, and Shih-Fu Chang. 2019. Counterfactual Critic Multi-Agent Training for Scene Graph Generation. In *ICCV*.

[7] Zhiyong Cui, Kristian Henrickson, Ruimin Ke, and Yinhai Wang. 2018. High-Order Graph Convolutional Recurrent Neural Network: A Deep Learning Framework for Network-Scale Traffic Learning and Forecasting. *CoRR* abs/1802.07007 (2018).

[8] Martin Danelljan, Gustav Häger, Fahad Shahbaz Khan, and Michael Felsberg. 2014. Accurate Scale Estimation for Robust Visual Tracking. In *British Machine Vision Conference, BMVC 2014, Nottingham, UK, September 1-5, 2014*.

[9] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. 2016. Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*. 3837–3845.

[10] Rohit Girdhar, Deva Ramanan, Abhinav Gupta, Josef Sivic, and Bryan Russell. 2017. ActionVLAD: Learning Spatio-Temporal Aggregation for Action Classification. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*. 3165–3174.

[11] William L. Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive Representation Learning on Large Graphs. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*. 1025–1035.

[12] Anfeng He, Chong Luo, Xinmei Tian, and Wenjun Zeng. 2018. A Twofold Siamese Network for Real-Time Object Tracking. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*. 4834–4843.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*. 770–778.

[14] Mikael Henaff, Joan Bruna, and Yann LeCun. 2015. Deep Convolutional Networks on Graph-Structured Data. *CoRR* abs/1506.05163 (2015).

[15] Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*.

[16] Yikang Li, Wanli Ouyang, Bolei Zhou, Kun Wang, and Xiaogang Wang. 2017. Scene Graph Generation from Objects, Phrases and Region Captions. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*. 1270–1279.

[17] Cewu Lu, Ranjay Krishna, Michael S. Bernstein, and Fei-Fei Li. 2016. Visual Relationship Detection with Language Priors. In *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part I*. 852–869.

[18] Kenneth Marino, Ruslan Salakhutdinov, and Abhinav Gupta. 2017. The More You Know: Using Knowledge Graphs for Image Classification. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*. 20–28.

[19] Federico Monti, Davide Boscaini, Jonathan Masci, Emanuele Rodolà, Jan Svoboda, and Michael M. Bronstein. 2017. Geometric Deep Learning on Graphs and Manifolds Using Mixture Model CNNs. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*. 5425–5434.

[20] Julia Peyre, Ivan Laptev, Cordelia Schmid, and Josef Sivic. 2017. Weakly-Supervised Learning of Visual Relations. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*. 5189–5198.

[21] Afshin Rahimi, Trevor Cohn, and Timothy Baldwin. 2018. Semi-supervised User Geolocation via Graph Convolutional Networks. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*. 2009–2019.

[22] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. 2015. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*. 91–99.

[23] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael S. Bernstein, Alexander C. Berg, and Fei-Fei Li. 2015. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision* 115, 3 (2015), 211–252.

[24] Mohammad Amin Sadeghi and Ali Farhadi. 2011. Recognition using visual phrases. In *The 24th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2011, Colorado Springs, CO, USA, 20-25 June 2011*. 1745–1752.

[25] Victor Garcia Satorras and Joan Bruna Estrach. 2018. Few-Shot Learning with Graph Neural Networks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*.

[26] Xindi Shang, Tongwei Ren, Jingfan Guo, Hanwang Zhang, and Tat-Seng Chua. 2017. Video Visual Relation Detection. In *Proceedings of the 2017 ACM on Multimedia Conference, MM 2017, Mountain View, CA, USA, October 23-27, 2017*. 1300–1308.

[27] Abhinav Shrivastava, Abhinav Gupta, and Ross B. Girshick. 2016. Training Region-Based Object Detectors with Online Hard Example Mining. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*. 761–769.

[28] Damien Teney, Lingqiao Liu, and Anton van den Hengel. 2017. Graph-Structured Representations for Visual Question Answering. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*. 3233–3241.

[29] Jack Valmadre, Luca Bertinetto, João F. Henriques, Andrea Vedaldi, and Philip H. S. Torr. 2017. End-to-End Representation Learning for Correlation Filter Based Tracking. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*. 5000–5008.

[30] Rianne van den Berg, Thomas N. Kipf, and Max Welling. 2017. Graph Convolutional Matrix Completion. *CoRR* abs/1706.02263 (2017).

[31] Heng Wang and Cordelia Schmid. 2013. Action Recognition with Improved Trajectories. In *IEEE International Conference on Computer Vision, ICCV 2013, Sydney, Australia, December 1-8, 2013*. 3551–3558.

[32] Xiaolong Wang, Ross B. Girshick, Abhinav Gupta, and Kaiming He. 2018. Non-Local Neural Networks. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*. 7794–7803.

[33] Xiaolong Wang and Abhinav Gupta. 2018. Videos as Space-Time Region Graphs. In *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part V*. 413–431.

[34] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. 2017. Simple online and realtime tracking with a deep association metric. In *2017 IEEE International Conference on Image Processing, ICIP 2017, Beijing, China, September 17-20, 2017*. 3645–3649.

[35] Danfei Xu, Yuke Zhu, Christopher B. Choy, and Li Fei-Fei. 2017. Scene Graph Generation by Iterative Message Passing. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*. 3097–3106.

[36] Ning Xu, An-An Liu, Yongkang Wong, Yongdong Zhang, Weizhi Nie, Yuting Su, and Mohan Kankanhalli. 2018. Dual-stream recurrent neural network for video captioning. *IEEE Transactions on Circuits and Systems for Video Technology* (2018).

[37] Jianwei Yang, Jiasen Lu, Stefan Lee, Dhruv Batra, and Devi Parikh. 2018. Graph R-CNN for Scene Graph Generation. In *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part I*. 690–706.

[38] Guojun Yin, Lu Sheng, Bin Liu, Nenghai Yu, Xiaogang Wang, Jing Shao, and Chen Change Loy. 2018. Zoom-Net: Mining Deep Feature Interactions for Visual Relationship Recognition. In *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part III*. 330–347.

[39] Rex Ying, Ruining He, Kaifeng Chen, Pong Eksombatchai, William L. Hamilton, and Jure Leskovec. 2018. Graph Convolutional Neural Networks for Web-Scale Recommender Systems. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2018, London, UK, August 19-23, 2018*. 974–983.

[40] Ruichi Yu, Ang Li, Vlad I. Morariu, and Larry S. Davis. 2017. Visual Relationship Detection with Internal and External Linguistic Knowledge Distillation. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*. 1068–1076.

[41] Rowan Zellers, Mark Yatskar, Sam Thomson, and Yejin Choi. 2018. Neural Motifs: Scene Graph Parsing With Global Context. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*. 5831–5840.

[42] Hanwang Zhang, Zawlin Kyaw, Shih-Fu Chang, and Tat-Seng Chua. 2017. Visual Translation Embedding Network for Visual Relation Detection. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*. 3107–3115.

[43] Bolei Zhou, Alex Andonian, Aude Oliva, and Antonio Torralba. 2018. Temporal Relational Reasoning in Videos. In *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part I.* 831–846.

[44] Bohan Zhuang, Lingqiao Liu, Chunhua Shen, and Ian D. Reid. 2017. Towards Context-Aware Interaction Recognition for Visual Relationship Detection. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017.* 589–598.