Beyond Short-Term Snippet: Video Relation Detection with Spatio-Temporal Global Context

Chenchen Liu[1], Yang Jin[2], Kehan Xu[1], Guoqiang Gong[1], Yadong Mu[1*]
[1]Peking University, [2]Beihang University
[1]{liuchenchen, yurina, gonggq, myd}@pku.edu.cn, [2]jy0205@buaa.edu.cn

Abstract

Video visual relation detection (VidVRD) aims to describe all interacting objects in a video. Different from relationships in static images, videos contain an addition temporal channel. A majority of existing works divide a video into short segments, predict relationships in each segment, and merge them. Such methods cannot capture relations involving long motions. Predicting the same relationship across neighboring video segments is also inefficient. To address these issues, this work proposes a novel sliding-window scheme to simultaneously predict short-term and long-term relationships. We run windows with different kernel sizes on object tracklets to generate sub-tracklet proposals with different duration, while the computational load is similar to that in segment-based methods. To fully utilize spatial and temporal information in videos, we construct one spatial and one temporal graph and employ Graph Convloutional Network to generate contextual embedding for tracklet proposal compatibility evaluation. We only predict relationships on highly-compatible proposal pairs. Our method achieves state-of-the-art performance on both ImageNet-VidVRD and VidOR dataset across multiple tasks. Especially for ImageNet-VidVRD, we obtain an average of 3% (R@50 from 8.07% to 11.21%) improvement under all evaluation metrics.

1. Introduction

Booted by the impressive development of deep learning, we have made unprecedented progress towards machine comprehension on visual information. A lot of research efforts have been made on low-level vision tasks such as object classification/detection [29, 6, 28, 14] and semantic segmentation [53, 23]. However, understanding visual contents at a higher semantic level still remains challenging. To bridge the gap between low-level vision-only tasks and high-level vision-language ones, visual relation detection (VRD) serves as a promising intermediate task.

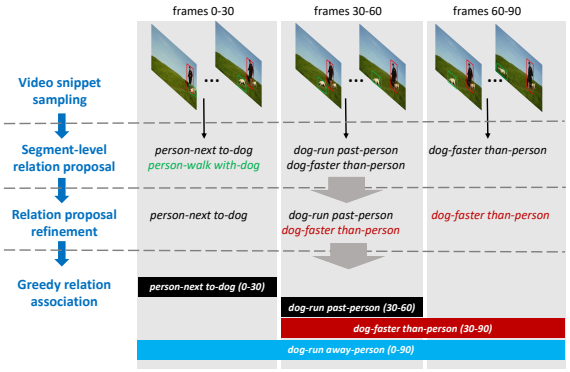

Figure 1. Typical pipeline of existing segment-based video visual relation detection methods. Relationships are separately detected in each short segment and merged afterwards. In particular, for the *greedy relation association* stage, black, red, and blue bars correspond to directly detected, detected after merging, and missing relationships (mainly owing to long duration), respectively.

Visual relations between entities, denoted by the ⟨*subject, predicate, object*⟩ triplet, offers a comprehensive visual content understanding beyond objects. In order to localize objects in one scene and recognize their interactions, visual relation detection requires the model to fully capture fine-grained visual cues, while the dependency on language priors to output visual relations is relatively minor. This is a steady first step towards connecting computer vision and natural language. Early works in visual relation detection mostly focused on static images (ImgVRD) [21, 20, 8, 58, 47, 44] and achieved exciting results. Recently, a new task is proposed by [34] as video visual relation detection (VidVRD). It involves detecting and tracking pairs of objects in a video, as well as recognizing the dynamic interaction between them. Effectively capturing spatial and action relations between objects in a video is especially useful, improving the results in captioning [43, 45], video retrieval [2, 13], visual question answer [1, 39], and many other visual-language tasks.

A direct way to tackle VidVRD problem would be to

*Corresponding author.

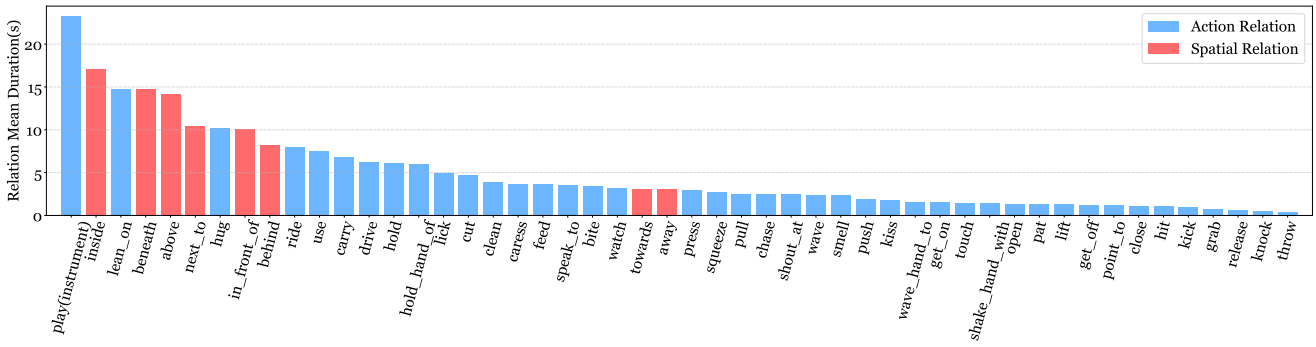

Figure 2. Mean duration of relations in VidOR. The x, y axis are relationship category and mean duration (seconds) respectively.

extend ImgVRD solutions to video case, but it can easily fail due to the ignorance of video-visual-relation-specific characteristics. First, some dynamic interactions between objects, such as ⟨*airplane, move past, watercraft*⟩ , can only be observed in videos. Videos are also capable of resolving ambiguities in motions. Second, interactions between two specific objects in a video can change or disappear over time. Therefore, simply applying ImgVRD methods to each video frame would mean omitting the additional relation information provided in videos.

Currently, some methods [34, 36, 27] are designed to capture such dynamic and time-varying relations between entities in videos. As shown in Figure 1, these methods first decompose the target video into several one-second segments, then detect initial relations fully from local segment information. The dominating way for get final video-level relations are greedy local association as in [34], which greedily merges two adjacent segments if they contain the same relation. In the relation proposal refinement step, the most recent [36] linked all segments as a Markov Random Fields (MRF), which leads to an illusion of a global method. However, the pairwise cliques in MRF mainly enhance smoothness between adjacent segments. The last step of [36] still used greedy local association for obtaining video-level results, therefore it is arguably still local (*e.g.*, unable to recognize the long-duration "dog-run away-person" that crosses 3 segments). Similar argument holds for GCN-based method in [27]. On the other hand, these methods bring computation redundancy by predicting relations for each segment. In Figure 2, we summarized the mean duration of all relations in the VidVOR dataset [34]. Almost all of the relations last more than one second, crossing multiple video segments. Separately predicting relationships in each segment and merging them together contains redundant computation on the same relation under similar appearance.

In this paper, we decompose our method into three independent stages. The first stage generates object tracklet proposals; The second stage refines proposal features and find related subject-object proposals; The third stage focuses on predicting the relationships. To solve the two problems mentioned above, we propose a novel sliding-window strategy in tracklet proposal generation to substitute the common video segmentation approach. We run windows with multi-scale kernels on tracklet frames to get tracklet proposals with varying length. All tracklets in each unique window, indicating they are in the same temporal interval, are passed to stage two for compatibility evaluation. This sliding-window approach ensures the observation of relations on varying temporal scales, and increases efficiency in detecting the same relationship in continuous frames.

Stage two takes in tracklet proposal features and constructs one spatial and one temporal graph to refine these features. Graph Convolutional Network [19] is utilized to embed contextual information into proposal features. Two groups of features incorporated with spatial and temporal knowledge respectively are then fused together. The resulted features pass through one pair correlation embedding module to get the final contextual embedding. Compatibility is evaluated between tracklet embeddings to determine related proposal pairs. In stage three, detection features, I3D features and relative motion features are extracted from pair proposals and combined together to predict the relationship distribution.

Our model is evaluated on two video relation benchmarks: ImageNet VidVRD dataset [34] and VidOR dataset [34]. Utilizing the sliding-window approach for tracklets generation and spatial and temporal graphs for proposal compatibility evaluation, our method achieved state-of-the-art performance on two tasks: video relation detection and video relation tagging. We also conducted comprehensive ablation study and observed intermediate outputs to prove the effectiveness of our pipeline.

2. Related Work

Video Object Detection Recent deep learning-based methods [15, 29, 38] have achieved mature performance on

object detection in static images. For object detection in videos, an additional temporal channel is added, so the goal is to localize objects with bounding box trajectories. Temporal continuity in videos provides useful information for object detection, but the existence of blur, occlusion and camera motion would hamper accurate localization. It is also necessary to improve the speed upon still image detectors for video object detection. Most methods focusing on improving per-frame detection results with temporal information can be classified into two steams: box-level based and feature-level based. Box-level methods [18, 11, 49] combine image object detection and tracking. Bounding boxes are first detected in individual frames, then they are associated across frames by employing external tracking modules. Feature-level methods [55, 56, 57] utilizes temporal information via flow-guided feature propagation from previous frames. This early integration of temporal cues with features directly improves detection accuracy in each frame.

Visual Relation Detection Several methods have been proposed for visual relation detection in static images (ImgVRD). In [24], the task of visual relation detection is defined as both predicting the $\langle$*subject, predicate, object*$\rangle$ triplet and localizing object and subject by bounding box. The most common approach to detect visual relations is to first find all objects in the image and predict relationship between each pair of them, as in [24, 46, 48, 51, 9]. [46] facilitates interaction between local object features and global predicate features by introducing two new pooling modules.Instead of detecting all objects in the first step, [52] proposes a Relation Proposal Network to evaluate the compatibility between class-agnostic proposals. Their pipeline only predicts relation and object and subject category for related proposals.

Recently, the first dataset on video visual relation detection (VidVRD) is introduced by [34]. [33] also proposed a baseline for VidVRD task, decomposing one video into segments and merging relationship predictions in neighbouring segments through a greedy association algorithm. A fully-connected spatial-temporal graph is constructed for each video in [27], and feature interaction is formulated via Graph Convolutional Network. [36] constructs a similar graph as above, but utilizes Conditional Random Field to exploit the statistical dependency between objects. Our work introduces a sliding-window approach to observe relations with different length, without the need to merge the same relations in different segments.

Graph Neural Networks Graph Neural Networks (GNNs) [12, 32] is widely applied to process graph data, propagating neighboring information among elements in the graph. Encouraged by the success of convolutional neural network in computer vision applications, a lot of research efforts have been made to re-define convolution of graph data. All methods fall into two categories, spectral-based [5, 17, 10] and spatial-based [25, 3, 26] graph convolution[42]. Graph Convolutional Network (GCN) [19] successfully bridged the gap between these two approaches, generalizing the operation of convolution from non-structural data to graph-structured data. GCN stacks multiple graph convloutional layers to extract high-level representation for nodes. [27] abstracts videos into fully-connected spatial-temporal graphs and conducted reasoning in these 3D graphs. In our work, we build two graphs to separately formulate spatial and temporal information interactions between tracklet proposal features.

3. The Proposed Approach

As shown in Figure 3, we can naturally decompose video relation detection into three sub-tasks: Object Tracklets Proposal, Relationship Pair Proposal and Relationship Classification. Stage one generates object tracklet proposals with varying length using the sliding-window approach. For each unique window, all tracklet proposals in this temporal interval will be the input to stage two. Stage two assembles compatible subject-object pairs by congregating spatial and temporal information into proposal features. Stage three then predicts relationship for each pair with the detected visual features, I3D features, and position-related relative motion features. As irrelevant pairs are filtered out in stage two, all calculations in stage three would result in valid relationships. Therefore, our pipeline is highly efficient.

3.1. Object Tracklet Proposal

Video Object Detection Following [34], we first employ an object detector in video frames, then track frame-level detection results across the whole video. For image object detection, we use a Faster-RCNN [29] with ResNet101 [16] backbone as our detector. The detector is trained on images from MS-COCO [22] and ILSVRC2016-DET [30] datasets, with 35 categories in total. In order to reduce the overlapping area between bounding boxes, we perform non-maximum suppression (NMS) with Intersection Over Union (IoU) $>$ 0.5 on image object detection results. For object tracking, we run a Multiple Object Tracking (MOT) algorithm on the entire video to connect the same object between neighbouring frames. Performing MOT on the entire video is more difficult than on segments, as in [34], for a complete video can possibly contain occlusions and illumination variations. We choose deep sort [41] as our tracker which can integrate visual features as matching descriptors to improve tracking performance. Object detection features obtained from RoI Pooling [29] is used as visual features in deep sort. Different from [34], we perform NMS in the detection part instead of the tracking part. The reason is that NMS in tracking is category-agnostic, so overlapping trajectories from different classes

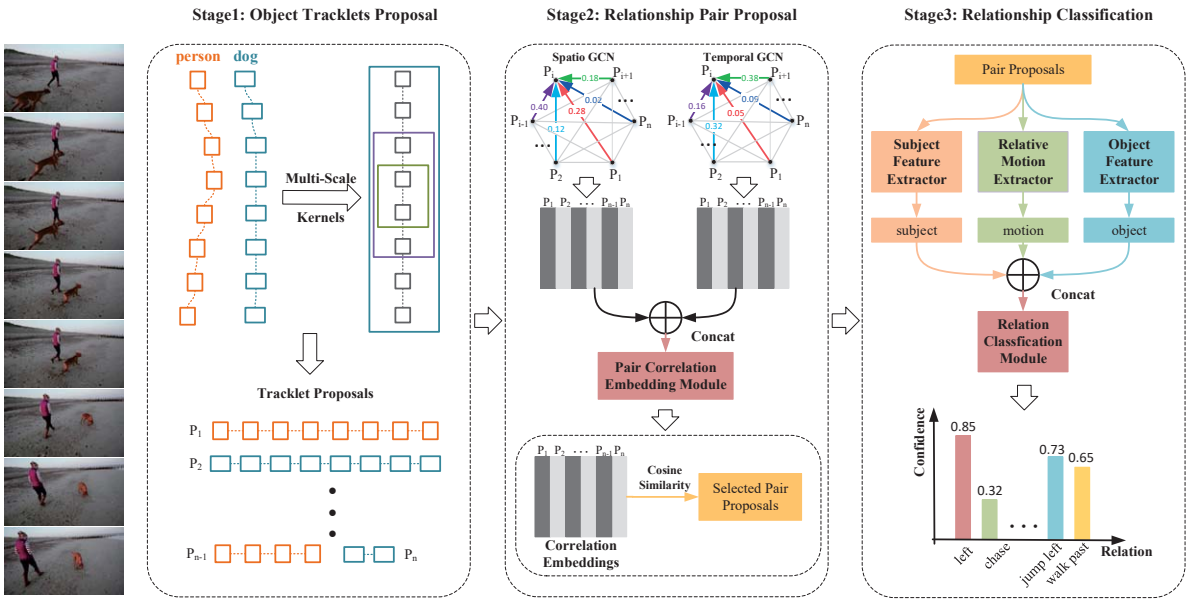

Figure 3. Illustration of our proposed method for video visual relation detection. Our model contains three stages: Object Tracklets Proposal, Relationship Pair Proposal and Relationship Classfication. See the main text for more explanation. Better viewing in color mode.

might be wrongly removed. NMS in detection is aware of object categories, avoiding such a problem, making our approach more effective than [34].

Proposal Generation from Sliding Windows We adopt a sliding-window approach to generate object tracklet proposals, as shown in Figure 3. Considering that true relation instances often exhibit large variance in their duration, we run the sliding window routine with multiple kernel size. Given one window unique on the temporal scale, all tracklets in this window are passed into stage two. Windows are enumerated according to the temporal position of tracklets, and the sliding window length value set depends on the length of each tracklet. To be more specific, suppose the tracklet frame length is L, we set the smallest window size to 30 frames and the largest to L. We would sample all multiples of the smallest window size that is smaller than L and also L as the length of sliding window. The stride of sliding windows sampling is empirically set to be half of window size without much fine-tuning.

Predicting relation on video parts with varying temporal scales has two advantages. The first is that only our pipeline can observe certain relationships that only exist in long video clips. For example, it is hard to distinguish between a person playing the piano and a person sitting in front of the piano from an one-second video clip, therefore previous works [34, 27, 36] observing only on short video segments can fail such cases. But as our method predicts relationship from tracklets with various length, we are able to figure out the correct motion by looking at a longer clip. The second is that our method does not require merging same relationship in neighboring segments, meaning that we can avoid both the cost of merging and that of redundant prediction on the same relation under similar scenes.

3.2. Relationship Pair Proposal

Stage two aims at filtering out incompatible proposal pairs, feeding only related pairs into the next stage for relationship classification. We construct one spatial graph and one temporal graph for information aggregation. Each node in the graph represents an object tracklet proposal, while tracklet proposal features from stage one is the initial value for each node. The edge between each two nodes is calculated as spatial IoU and temporal IoU respectively in two graphs. We use Graph Convolutional Networks to embed spatial and temporal contextual information individually into tracklet features based on the above two graphs. The two resulted feature vectors are concatenated to form the final feature for each node, congregating both spatial and temporal information in one representation. The embedding for all nodes are then fed into one pair correlation embedding module, generating compatibility-augmented representations for each tracklet. This module ensures compatible tracklets admit high cosine similarity between feature vectors. Two tracklet proposals bearing high similarity between feature are judged as related proposal pairs and sent to stage three.

We represent the tracklet proposals obtained from stage one as $\{P_1, P_2, ..., P_N\}$, where N is the number of object tracklet proposals.

Spatial Graph To encode spatial contextual information into tracklet features, we build a spatial graph using the spatial IoU (sIoU) between every two object tracklets.

Suppose the object in tracklet proposal P_i is detected in M frames. We average over all object bounding boxes in these M frames and denote the mean bounding box area as $\bar{p_i}$. sIoU between object tracklet proposal P_i and P_j is defined the Intersection Over Union between $\bar{p_i}$ and $\bar{p_j}$.

We perform normalization on each row of the graph matrix so that the weight of all edges connecting to one tracklet proposal P_i sum up to be 1. Motivated by the recent works [37, 40], we adopt the softmax function for normalization to obtain the value of edges in spatial graph G^{spa}:

$$G_{ij}^{spa} = exp(\text{sIoU}(P_i, P_j))/\sum_{j=0}^{N-1}(exp(\text{sIoU}(P_i, P_j))). \quad (1)$$

Temporal Graph To encode temporal contextual information into tracklet features, we build a temporal graph using the temporal IoU (tIoU) between every two object tracklets. tIoU between object tracklet proposal P_i and P_j is defined as the Intersection Over Union between their temporal interval.

Similar to spatial Graph, we use the softmax function for normalization to obtain the edge weights in temporal graph G^{tem}:

$$G_{ij}^{tem} = exp(\text{tIoU}(P_i, P_j))/\sum_{j=0}^{N-1}(exp(\text{tIoU}(P_i, P_j))). \quad (2)$$

Convolutions on Graph To embed spatial and temporal contextual information, we apply Graph Convolutional Network (GCN) proposed in [19]. Different from standard convolutions which operates on a local regular grid, graph convolution allows us to operate on graph data. Graph convolution is computed through gathering the response from each node's neighbours, and multiple graph convolutional layers are stacked to extract high-level representation for nodes. Therefore, graph convolution is suitable for formulating information exchange between nodes. The outputs of GCN are updated features for each node. The computation in one graph convolutional layer can be represented as:

$$\text{Z} = \text{GXW}, \quad (3)$$

where G represents the weights from one of our $N \times N$ adjacency graph (G^{spa}, G^{tem}); X is the input to this layer with dimension $N \times d$, containing features of each graph node; W is the trainable weight matrix of this layer with dimension $d \times d$. Therefore, the dimension of output Z is still $N \times d$, the same as input. The graph convolution operation can be stacked into multiple layers. After each layer of graph convolutions, we apply two non-linear functions including Layer Normalization [4] and ReLU activation before feature Z is forwarded to the next layer: $\text{Z} = \text{norm}(\sigma(\text{GXW}))$.

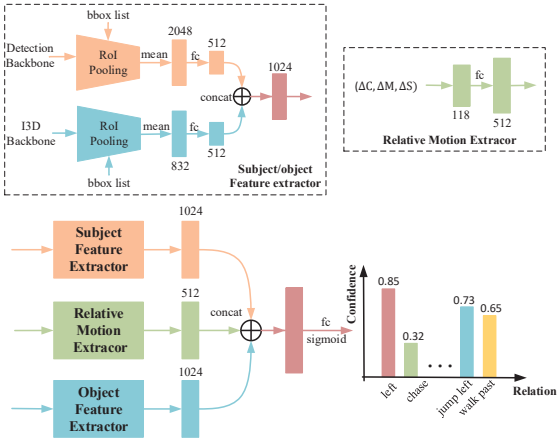

Figure 4. Illustration of our relationship classification module.

Pair Proposal After combining spatial and temporal contextual information with tracklet features in two graphs via GCN, we concatenate the resulted features, creating a comprehensive representation for each tracklet. To get relationship pair proposals, we send these features to another pair correlation embedding module for object compatibility evaluation. This module is implemented by two fully connect layers with ReLU. We then calculate cosine similarity on the output node features and select proposal pairs with high feature similarity. These pairs are considered to be related and sent to stage three for relationship classification.

3.3. Relationship Classification

We use $\langle P_s, P_o \rangle$ to represent a tracklet pair proposed by stage two, where P_s, P_o are both a sequence of bounding boxes. To predict the relation triplet $\langle subject, predicate, object \rangle$ involves the recognition of both object, subject categories, and the interactions between them. Given C object classes and R relationships, the total number of possible relationships will be $\mathcal{O}(C^2R)$. Learning an unbiased model on so many relationships with limited labeling data is a challenging task. One common solution to this problem is to learn separate models for object and relationship detection, reducing the complexity of training detectors to $\mathcal{O}(C + R)$. Our method following this solution to predict object and relation category separately.

For object classification, assuming that tracklet proposal P_i has M objects. The category of each object is detected in stage one. We find the mode of the M object categories C_i in stage three, and use C_i as the final category of the tracklet proposal.

For relationship classification, we first extract a relation feature vector for each pair proposal. As shown in Figure 4, the relationship classification module contains two different types of feature extractor: subject/object feature extractor and relative motion extractor. In subject/object feature

extractor, two backbones are responsible for extracting detection feature and I3D feature [7], and each is followed by an RoI pooling layer to get features with fixed dimension 2048 and 832. For relative motion feature extractor, we follow [34] to calculate ΔC, ΔS and ΔM between subject P_s and object P_o. The notations ΔC, ΔS, ΔM represent relative position, relative size and relative motion between subject and object tracklet proposals, respectively. These three vectors are concatenated into an 118-dimensional relative motion feature. The overall relation feature vector for relationship prediction between tracklet proposal pairs is the concatenation of P_s and P_o object features and the relative motion feature.

In training, we sample certain tracklet proposal pairs from stage two for relationship prediction. Only pairs overlapping with the ground truth by more than 0.5 in vIoU (volume IoU) are selected. we use multilabel classification and binary cross entropy loss function to calculate the loss between prediction and ground truth. In testing, we are consistent with [34], keeping top 20 prediction results for each pair proposal.

4. Experiments

To validate the effectiveness of our proposed method, we perform comprehensive experiments and compare the results with state-of-the-art VidVRD methods. In addition, we conducted a number of ablation studies to show the benefit of our modules.

4.1. Dataset Description

We adopt two video relation benchmarks in our experiments: ImageNet-VidVRD [34] and VidOR [33] dataset. ImageNet-VidVRD contains 1,000 videos (800 for training and rest 200 for evaluation) selected from ILVSRC2016-VID dataset [30]. Videos are chosen based on whether they contain clear visual relations. This dataset is well-labeled with object categories and corresponding trajectories. It covers 35 common subject/object categories and 132 relationships. All relations can be divided into three types: relative spatial positions, actions and actions adjectives. VidOR (Video Object Relation) is a large scale user-generated video dataset from social media. It has the same annotation format as ImageNet-VidVRD and is ten times larger, containing 10,000 videos selected from YFCC100M collection. VidOR contains 80 object categories and 50 predicate categories. It is split into 7,000 videos for training, 835 videos for validation, and 2,165 videos for testing. Due to its large size, VidOR brings great challenge to relation detection and tagging.

4.2. Evaluation Protocol

Following the setting in [34], we evaluate our method on two standard tasks: relation detection and relation tagging. For relation detection task, we aim to generate a set of $\langle$*subject, predicate, object*$\rangle$ relation triplets with tracklet proposals from a given video. The prediction is considered to be correct if there is a same relation triplet tagged in ground truth and both subject and object trajectories have sufficient vIoU (volume IoU). We set the overlapping threshold of vIoU to 0.5, the same as [34]. We adopt mean Average Precision (mAP) and Recall@K to evaluate the detection performance. Recall@K measures the fraction of the positive detection in the top K results, and K is set to 50 and 100. We use the mAP metric to evaluate the overall precision performance at different Recall values. For relation tagging task, we only consider the accuracy of predicted video relation triplets and ignore the object localization result. We aligned ourselves with [34] to use Precision@1, Precision@5 and Precision@10 to measure the accuracy of the tagging results.

4.3. Ablation Studies

To prove the effectiveness of stage two and three in our architecture, we design several ablation studies. For the Relationship Pair Proposal module, we explore how the number of spatial/temporal GCN layers can influence the performance of our method. The potency of our spatial and temporal graph is also investigated. For the Relationship Classification module, we show the effect of different feature combinations on both the task of relation detection and tagging.

Spatial and Temporal GCN In the Pair Proposal module (PPN), we build spatial and temporal graph on tracklet proposals. Two Graph Convolutional Networks are used to individually embed the spatial and temporal contextual information of each object tracklet proposal. In this part, we only evaluate on the output of stage two, judging on the accuracy of related pairs. We analyze the influence of GCN layer number on experimental performance. The results are shown in Table. 1 and Table. 2. For spatial GCN, our architecture achieves the best results at 3 layers. For temporal GCN, the best layer number is 2. GCN with more layers is prone to overfitting, resulting in a decline in performance. We use the best result for spatial and temporal GCN here in other experiments.

layer number	relation pair proposal		
	R@50	R@100	R@200
0	16.08	17.77	18.29
1	16.64	19.68	21.70
2	16.70	19.72	21.65
3	**16.94**	**19.88**	**21.84**
4	16.63	19.55	21.41

Table 1. Evaluation of our method with different numbers of spatial GCN layers on ImageNet-VidVRD dataset. Temporal GCN is removed in all the above experiments.

In addition, we investigate how these two GCN networks

layer number	relation pair proposal		
	R@50	R@100	R@200
0	16.08	17.77	18.29
1	16.54	19.54	21.51
2	**16.91**	**19.70**	21.77
3	16.64	19.54	21.60
4	16.40	19.40	**21.83**

Table 2. Evaluation of our method with different numbers of temporal GCN layers on ImageNet-VidVRD dataset. Spatial GCN is removed in all the above experiments.

cooperated with each other on improving the relation detection results. The results are shown in Table. 3. We can observe that both spatial GCN and temporal GCN lead to a significant increase in Recall@K. But the performance gain boosted by spatial GCN is greater than temporal GCN since the ImageNet-VidVRD dataset contains more relative spatial position relations. Literally, spatial relationships such as "walk left" and "stand front" are not sensitive to changes in time. Compared to spatial graph, temporal one does not provide much information in detecting such relationships. Nevertheless, the model shows the best results when spatial and temporal GCN are combined, which boosted a 3.8% improvement than only using spatial GCN in Recall@50.

graph type		relation pair proposal		
spatial	temporal	R@50	R@100	R@200
		16.08	17.77	18.29
✓		16.94	19.88	21.84
	✓	16.91	19.70	21.77
✓	✓	**17.58**	**20.46**	**22.23**

Table 3. Evaluation for the combination of spatial and temporal GCN used for Pair Proposal module on ImageNet-VidVRD dataset.

Relationship Classification module Our relationship classification module combines detected visual features, I3D features, and position-related relative motion features for relation prediction. Here, I3D features reflect action information in videos, while motion features indicate relative position information between subject and object. Detected visual features contain category information of object tracklet, which is the most crucial one for predicting relationship in the triplet. Therefore, we keep detected visual features in all cases in ablation study. Our ablation study compares between three situations: only detected features, detected features and I3D features and all three features.

Table. 4 displays the performance of these three models. We can observe that both I3D features and motion features can boost the performance of our network. It's worth noting that motion features improves the performance to a greater extent than I3D features in both relation tasks. The reason is the same as that in the previous spatio-temporal GCN ablation study: the ImageNet-VidVRD dataset has more relative spatial position relations than action ones; Motion features contains the relative position, size and velocity information between two object trajectories, which are all crucial for spatial relation.

Our method obtains the best performance when fusing detected visual features, I3D features and motion features together. This demonstrates that all three features are effective in relation detection and tagging.

feature type			relation detection			relation tagging		
detection	I3D	motion	R@50	R@100	mAP	P@1	P@5	P@10
✓			8.47	11.00	14.01	56.50	36.70	26.60
✓	✓		9.14	11.39	14.81	55.50	38.90	28.90
✓	✓	✓	**11.21**	**13.69**	**18.38**	**60.00**	**43.10**	**32.24**

Table 4. Evaluation for different kinds of features used in relation classification stage on ImageNet-VidVRD dataset.

4.4. Results Analysis

In this part, we compare our proposed method with some state-of-the-art methods on the above two datasets.

Methods to Compare For ImageNet-VidVRD benchmark, we consider to compare with the following methods: VP [31], Lu's-V [24], Lu's [24], VTransE [50], VidVRD [34], GSTEG [36] and VRD-GCN [27]. The first four methods focus on feature extraction from static images but ignore dynamic features in videos. VidVRD detects visual relations in videos through object tracklet generation, relation prediction and greedy relational association. GSTEG constructs a Conditional Random Field on a fully-connected spatio-temporal graph and designs a novel gated energy function. VRD-GCN extracts video features from a similar spatial-temporal graph convolutional network and proposes an association algorithm using vIoU similarity confidence scores.

For VidOR benchmark, we compare our method with the top 2 competitors in ACM Multimedia 2019 Grand Challenge. We use their team names to represent their methods, which are MAGUS.Gamma [35] (the first place) and RELAbuilder [54] (the second place).

Quantitative Analysis To evaluate the performance of the PPN module, we compare our results with VidVRD [34] baseline. Here we use the same metrics as those in ablation studies. The results are shown in Table. 5. We can observe that Recall@K (K equals 50, 100 and 200) of our method are much better than those in VidVRD baseline. This is because Pair Proposal module filters out most incompatible proposal pairs, reducing computational waste in stage three and making it easier to train the classification network. By adding PPN module, we can select better proposal pairs and get a higher Recall.

We show our overall performance and comparisons results to the baselines in Table. 6 (ImageNet-VidVRD benchmark) and Table. 7 (VidOR benchmark). On ImageNet-VidVRD benchmark, our proposed method outperforms all the comparison methods by a large margin on all

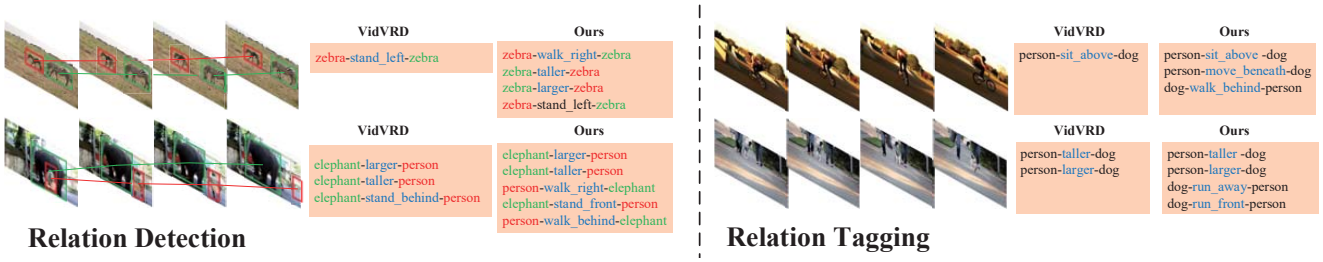

Figure 5. Visualization of video relation detection and relation tagging results using VidVRD baseline and our model. Better viewing if enlarging the images.

Method	relation detection		
	R@50	R@100	R@200
VidVRD[34]	10.87	12.18	14.51
PPN(Ours)	16.08	17.77	18.29
PPN-STGCN(Ours)	**17.58**	**20.46**	**22.23**

Table 5. Experimental results of relation pair proposal stage. See the main text for more explanation.

Method	relation detection			relation tagging		
	R@50	R@100	mAP	P@1	P@5	P@10
VP [31]	0.89	1.41	1.01	36.50	25.55	19.20
Lu's-V [24]	0.99	1.80	2.37	20.00	12.60	9.55
Lu's [24]	1.10	2.23	2.40	20.50	16.30	14.05
VTransE [50]	0.72	1.45	1.23	15.00	10.00	7.65
VidVRD [34]	5.54	6.37	8.58	43.00	28.90	20.80
GSTEG [36]	7.05	8.67	9.52	51.50	39.50	28.23
VRD-GCN [27]	8.07	9.33	16.26	57.50	41.00	28.50
Ours	**11.21**	**13.69**	**18.38**	**60.00**	**43.10**	**32.24**

Table 6. Experimental results of different methods for standard video relation detection and video relation tagging on ImageNet-VidVRD benchmark.

metrics. On VidOR benchmark, our method achieved great improvement in relation detection task and also comparable results in relation tagging task. [35] gained better results in relation tagging because they utilized optical flow models to gain superior results in video object detection. This is especially effective for VidOR dataset because it contains lots of motion relations. However, optical flow requires huge amounts of computation, and thus their pipeline are far less efficient than ours. We use the same video object detector with [54], and achieved better results.

Method	relation detection			relation tagging	
	R@50	R@100	mAP	P@1	P@5
RELAbuilder [54]	1.58	1.85	1.47	33.05	35.27
OTD+CAI [35]	6.19	8.16	5.65	48.31	38.49
OTD+GSTEG [35]	6.40	8.43	5.58	51.20	37.26
MAGUS.Gamma [35]	6.89	8.83	6.56	**51.20**	**40.73**
Ours	**8.21**	**9.90**	**6.85**	48.92	36.78

Table 7. Experimental results for standard video relation detection and video relation tagging on VidOR benchmark. [35] achieved better results in relation tagging, but it requires huge amounts of computation. See the main text for more explanation.

Carefully observing the results in Table. 6 and Table. 7, we find that among all the evaluation metrics, Recall@K (K equals 50 and 100) have the largest increase compared to other methods. There exist two main reasons. The first is that our proposed method utilizes a sliding-window approach with multi-scale kernels to generate object tracklet proposals, so that we can detect relationships with varying duration and predict temporal boundary more accurately. The second is that our PPN module filters out many inaccurate proposal pairs. Therefore, our method outperforms all the baselines on Recall@K (K equals 50 and 100) metric.

Qualitative Analysis We illustrate our qualitative results in Figure 5. As shown in this figure, VidVRD baseline detects few relationships and fails to classify relations like "walk" and "stand", especially when the object moves very slow. This is because the whole scene changes little in short video segments, making it difficult to distinguish between stand or walk. Benefited from generating tracklet proposals in a varying duration, our method can detect both long-term and short-term relationships. This enables us to achieved better performance than methods relying only on short video segments.

5. Conclusions

In this paper, we propose a three stage method for both the task of video visual relation detection and tagging. The first stage generates object tracklet proposals; The second stage refines proposal features and find related subject-object proposals; The third stage focuses on predicting the relationships between related pairs. Our pipeline is superior in that we can observe relationships with varying length efficiently using sliding window, while other works can only see relations in short video segments. We also utilized GCN in stage two to construct a contextual embedding incorporating spatial and temporal information for proposal compatibility evaluation. The experimental results on two datasets ImageNet-ViVRD and VidOR demonstrate that our method outperforms the state-of-the-art baselines on both video relation detection and relation tagging tasks.

Acknowledgement: This work is supported by National Key R&D Program of China (2018AAA0100702), Beijing Natural Science Foundation (Z190001), and National Natural Science Foundation of China (61772037).

References

[1] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. L. Zitnick, and D. Parikh. VQA: visual question answering. In *ICCV*, 2015. 1

[2] E. E. Apostolidis, V. Mezaris, M. Sahuguet, B. Huet, B. Cervenková, D. Stein, S. Eickeler, J. L. R. García, R. Troncy, and L. Pikora. Automatic fine-grained hyperlinking of videos within a closed collection using scene segmentation. In *ACM MM*, 2014. 1

[3] J. Atwood and D. Towsley. Diffusion-convolutional neural networks. In *NeurIPS*, 2016. 3

[4] L. J. Ba, J. R. Kiros, and G. E. Hinton. Layer normalization. *CoRR*, abs/1607.06450, 2016. 5

[5] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun. Spectral networks and locally connected networks on graphs. *arXiv preprint arXiv:1312.6203*, 2013. 3

[6] Z. Cai and N. Vasconcelos. Cascade R-CNN: delving into high quality object detection. In *CVPR*, 2018. 1

[7] J. Carreira and A. Zisserman. Quo vadis, action recognition? A new model and the kinetics dataset. In *CVPR*, 2017. 6

[8] B. Dai, Y. Zhang, and D. Lin. Detecting visual relationships with deep relational networks. In *CVPR*, 2017. 1

[9] B. Dai, Y. Zhang, and D. Lin. Detecting visual relationships with deep relational networks. In *CVPR*, 2017. 3

[10] M. Defferrard, X. Bresson, and P. Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *NeurIPS*, 2016. 3

[11] C. Feichtenhofer, A. Pinz, and A. Zisserman. Detect to track and track to detect. In *ICCV*, 2017. 3

[12] M. Gori, G. Monfardini, and F. Scarselli. A new model for learning in graph domains. In *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, volume 2, pages 729–734. IEEE, 2005. 3

[13] H. Guo, J. Wang, M. Xu, Z. Zha, and H. Lu. Learning multi-view deep features for small object retrieval in surveillance scenarios. In *ACM MM*, 2015. 1

[14] K. He, G. Gkioxari, P. Dollár, and R. B. Girshick. Mask R-CNN. In *ICCV*, 2017. 1

[15] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 2

[16] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 3

[17] M. Henaff, J. Bruna, and Y. LeCun. Deep convolutional networks on graph-structured data. *arXiv preprint arXiv:1506.05163*, 2015. 3

[18] K. Kang, W. Ouyang, H. Li, and X. Wang. Object detection from video tubelets with convolutional neural networks. In *CVPR*, 2016. 3

[19] T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. In *ICLR*, 2017. 2, 3, 5

[20] Y. Li, W. Ouyang, X. Wang, and X. Tang. Vip-cnn: Visual phrase guided convolutional neural network. In *CVPR*, 2017. 1

[21] W. Liao, B. Rosenhahn, L. Shuai, and M. Y. Yang. Natural language guided visual relationship detection. In *CVPR*, 2019. 1

[22] T. Lin, M. Maire, S. J. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft COCO: common objects in context. In *ECCV*, 2014. 3

[23] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015. 1

[24] C. Lu, R. Krishna, M. S. Bernstein, and F. Li. Visual relationship detection with language priors. In *ECCV*, 2016. 3, 7, 8

[25] A. Micheli. Neural network for graphs: A contextual constructive approach. *IEEE Transactions on Neural Networks*, 20(3):498–511, 2009. 3

[26] M. Niepert, M. Ahmed, and K. Kutzkov. Learning convolutional neural networks for graphs. In *ICML*, 2016. 3

[27] X. Qian, Y. Zhuang, Y. Li, S. Xiao, S. Pu, and J. Xiao. Video relation detection with spatio-temporal graph. In *ACM MM*, 2019. 2, 3, 4, 7, 8

[28] J. Redmon, S. K. Divvala, R. B. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *CVPR*, 2016. 1

[29] S. Ren, K. He, R. B. Girshick, and J. Sun. Faster R-CNN: towards real-time object detection with region proposal networks. *IEEE Trans. Pattern Anal. Mach. Intell.*, 39(6):1137–1149, 2017. 1, 2, 3

[30] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. S. Bernstein, A. C. Berg, and F. Li. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015. 3, 6

[31] M. A. Sadeghi and A. Farhadi. Recognition using visual phrases. In *CVPR*, 2011. 7, 8

[32] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini. The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1):61–80, 2008. 3

[33] X. Shang, D. Di, J. Xiao, Y. Cao, X. Yang, and T.-S. Chua. Annotating objects and relations in user-generated videos. In *ICMR*, 2019. 3, 6

[34] X. Shang, T. Ren, J. Guo, H. Zhang, and T. Chua. Video visual relation detection. In *ACM MM*, 2017. 1, 2, 3, 4, 6, 7, 8

[35] X. Sun, T. Ren, Y. Zi, and G. Wu. Video visual relation detection via multi-modal feature fusion. In *ACM MM*, 2019. 7, 8

[36] Y. H. Tsai, S. K. Divvala, L. Morency, R. Salakhutdinov, and A. Farhadi. Video relationship reasoning using gated spatio-temporal energy graph. In *CVPR*, 2019. 2, 3, 4, 7, 8

[37] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 5

[38] M. Wang, C. Luo, R. Hong, J. Tang, and J. Feng. Beyond object proposals: Random crop pooling for multi-label image recognition. *IEEE Transactions on Image Processing*, 25(12):5678–5688, 2016. 2

[39] P. Wang, Q. Wu, C. Shen, and A. van den Hengel. The vqa-machine: Learning how to use existing vision algorithms to answer new questions. In *CVPR*, 2017. 1

[40] X. Wang, R. B. Girshick, A. Gupta, and K. He. Non-local neural networks. In *CVPR*, 2018. 5

[41] N. Wojke, A. Bewley, and D. Paulus. Simple online and realtime tracking with a deep association metric. In *ICIP*, 2017. 3

[42] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu. A comprehensive survey on graph neural networks. *arXiv preprint arXiv:1901.00596*, 2019. 3

[43] K. Xu, J. Ba, R. Kiros, K. Cho, A. C. Courville, R. Salakhutdinov, R. S. Zemel, and Y. Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *ICML*, 2015. 1

[44] J. Yang, J. Lu, S. Lee, D. Batra, and D. Parikh. Graph R-CNN for scene graph generation. In *ECCV*, 2018. 1

[45] X. Yang, K. Tang, H. Zhang, and J. Cai. Auto-encoding scene graphs for image captioning. In *CVPR*, 2019. 1

[46] G. Yin, L. Sheng, B. Liu, N. Yu, X. Wang, J. Shao, and C. Change Loy. Zoom-net: Mining deep feature interactions for visual relationship recognition. In *ECCV*, 2018. 3

[47] G. Yin, L. Sheng, B. Liu, N. Yu, X. Wang, J. Shao, and C. C. Loy. Zoom-net: Mining deep feature interactions for visual relationship recognition. In *ECCV*, 2018. 1

[48] R. Yu, A. Li, V. I. Morariu, and L. S. Davis. Visual relationship detection with internal and external linguistic knowledge distillation. In *ICCV*, 2017. 3

[49] X. Zeng, W. Ouyang, J. Yan, H. Li, T. Xiao, K. Wang, Y. Liu, Y. Zhou, B. Yang, Z. Wang, et al. Crafting gbd-net for object detection. *IEEE transactions on pattern analysis and machine intelligence*, 40(9):2109–2123, 2017. 3

[50] H. Zhang, Z. Kyaw, S. Chang, and T. Chua. Visual translation embedding network for visual relation detection. In *CVPR*, 2017. 7, 8

[51] H. Zhang, Z. Kyaw, S.-F. Chang, and T.-S. Chua. Visual translation embedding network for visual relation detection. In *ICCV*, 2017. 3

[52] J. Zhang, M. Elhoseiny, S. Cohen, W. Chang, and A. Elgammal. Relationship proposal networks. In *CVPR*, 2017. 3

[53] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia. Pyramid scene parsing network. In *CVPR*, 2017. 1

[54] S. Zheng, X. Chen, S. Chen, and Q. Jin. Relation understanding in videos. In *ACM MM*, 2019. 7, 8

[55] X. Zhu, J. Dai, L. Yuan, and Y. Wei. Towards high performance video object detection. In *CVPR*, 2018. 3

[56] X. Zhu, Y. Wang, J. Dai, L. Yuan, and Y. Wei. Flow-guided feature aggregation for video object detection. In *ICCV*, 2017. 3

[57] X. Zhu, Y. Xiong, J. Dai, L. Yuan, and Y. Wei. Deep feature flow for video recognition. In *CVPR*, 2017. 3

[58] Y. Zhu and S. Jiang. Deep structured learning for visual relationship detection. In *AAAI*, 2018. 1