Target Adaptive Context Aggregation for Video Scene Graph Generation

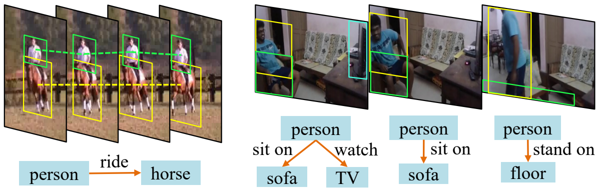
Yao Teng[1] Limin Wang[1] ✉ Zhifeng Li[2] Gangshan Wu[1]
[1]State Key Laboratory for Novel Software Technology, Nanjing University, China
[2]Tencent AI Lab, Shenzhen, China
tengyao19980325@gmail.com, {lmwang, gswu}@nju.edu.cn, michaelzfli@tencent.com

Abstract

This paper deals with a challenging task of video scene graph generation (VidSGG), which could serve as a structured video representation for high-level understanding tasks. We present a new detect-to-track *paradigm for this task by decoupling the context modeling for relation prediction from the complicated low-level entity tracking. Specifically, we design an efficient method for frame-level VidSGG, termed as* Target Adaptive Context Aggregation Network *(TRACE), with a focus on capturing spatio-temporal context information for relation recognition. Our TRACE framework streamlines the VidSGG pipeline with a modular design, and presents two unique blocks of Hierarchical Relation Tree (HRTree) construction and Target-adaptive Context Aggregation. More specific, our HRTree first provides an adpative structure for organizing possible relation candidates efficiently, and guides context aggregation module to effectively capture spatio-temporal structure information. Then, we obtain a contextualized feature representation for each relation candidate and build a classification head to recognize its relation category. Finally, we provide a simple temporal association strategy to track TRACE detected results to yield the video-level VidSGG. We perform experiments on two VidSGG benchmarks: ImageNet-VidVRD and Action Genome, and the results demonstrate that our TRACE achieves the state-of-the-art performance. The code and models are made available at https://github.com/MCG-NJU/TRACE.*

1. Introduction

Video understanding tasks, such as action recognition [29, 1, 38, 39], temporal action localization [47, 20, 31], spatio-temporal action detection [18, 5], have received lots of research attention in the past few years. Most of these methods simply provide a single label or spatio-temporal extent of each action instance in a long video sequence.



(a) Video-level video scene graph (b) Frame-level video scene graph

Figure 1. (a) An example of video-level VidSGG. The subject/object nodes in this graph are entity trajectories and their relation is constant for this clip. (b) An example for frame-level VidSGG. The frame-level VidSGG is determined by short-term information for each frame and could vary along time.

However, an ideal video understanding system is expected to not only recognize the action types, but also provide detailed and structured interpretation of the whole scene by parsing an event into a sequence of interactions between different visual entities. This structured video representation, known as video scene graph [27], can contribute to more accurate action recognition [10] and allow for our vision models to tackle high-level and complex inference tasks, such as video caption [7, 36], video retrieval [4, 16], and video question answering [33, 15]. Nevertheless, video scene graph generation (VidSGG) [25, 34, 22, 30] has received much less research efforts in our community, when compared with image scene graph generation [41, 43, 17].

As shown in Figure 1, the existing benchmark of VidSGG can be roughly grouped into two types according to the granularity of its graph representation: (1) video-level scene graph generation, where each graph node represents an object trajectory, and the edge captures the relation between visual entities, which is constant for one clip. (2) frame-level scene graph generation, where the graph is defined at frame level and the relation could change over time in this short clip. For video-level VidSGG, it requires to accurately trim long videos into short clips (e.g., 30 frames) in advance, according to the precise temporal boundaries of relations. This setting cannot be easily adapted to realistic VidSGG in untrimmed videos, as trimming is difficult and subjective due to temporal ambiguity. In contrast, frame-level VidSGG provides a more flexible mechanism for rela-

✉: Corresponding author.

tion representation in continuous video streams. In addition, these frame-level VidSGG could easily yield the video-level scene graph by using temporal association to track adjacent results. However, previous works [27, 25, 22] on VidSGG mainly neglect the frame-level scene graphs, and directly recognize video-level relations based on the results of object tracking. As a result, they all yield a heavy pipeline highly dependent on tracking.

In this paper, we aim to present a new method to address the above two tasks simultaneously. Our basic idea is to first generate the video scene graph at each frame by utilizing short-term video information, and then track each frame-level scene graph along time dimension to obtain the video-level result. We argue that this *detect-to-track* VidSGG paradigm will decouple the tasks of video relation recognition and temporal tracking, making our method focus more on modeling spatio-temporal context in videos. The key of recognizing visual relation is that inferring the interaction between visual entities usually requires comprehensive understanding of spatio-temporal context information in the video. For instance, recognizing whether a person is sitting on a sofa or standing from a sofa is based on the temporal variation of human movement with respect to sofa over time. Thus, we aim to devise a modular framework that can effectively determine and capture such complex spatio-temporal contextual information (e.g., temporal motion, object relation, person relation etc.) for efficient VidSGG.

Spatio-temporal context information is much more complex and diverse in videos than single images. To handle this issue, we design an efficient adaptive framework to select and propagate contextual information in videos, coined as TaRget Adaptive Context AggrEgation Network (TRACE). The key to TRACE is to organize relation candidates with an adaptive hierarchical relation tree (HRTree), and then perform target-adaptive context information aggregation for each relation candidate based on it. HRTree is not only helpful for the information aggregation, but also enables efficient processing of numerous relation candidates in a limited memory consumption. As for effective context information aggregation, we present an attentive module to fuse temporal information selectively and a directional propagation module to capture spatial structured information. Finally, the target-adaptive aggregated representation for each candidate can provide sufficient contextual information for relation classification. Furthermore, we employ a common temporal association algorithm to link frame-level graphs into a video-level result.

Specifically, we utilize a 3D CNN for extracting temporal features, a 2D CNN for extracting center frame representation, and an object detection network for object candidates with their visual features. Based on these low-level visual representations, TRACE streamlines the VidSGG pipeline with modules of HRTree construction, context aggregation, relation classification, and optional temporal association. We evaluate TRACE on two datasets: Action Genome (AG) [10] and ImageNet-VidVRD (VidVRD) [27]. AG is a brand-new dataset for frame-level VidSGG and only the methods for image SGG are evaluated on it, while VidVRD is a video-level VidSGG dataset. On AG, as a new specific method for frame-level VidSGG, TRACE achieves the state-of-the-art performance on the standard three evaluation modes: scene graph detection (SGDet), scene graph classification (SGCls), and predicate classification (PredCls). Specifically, TRACE outperforms the best model on average by 1.5% and 1.2% for mAP$_{rel}$ [46] and mean Recalls [32] of the three modes, respectively, and get comparable performance at Recalls. On VidVRD, with a simple temporal linking strategy, our TRACE achieves good performance under the video-level metrics. Concretely, TRACE outperforms the best model with ground-truth trajectories and the same association algorithm by 2.8%, 1.0% and 2.1% at mAP, Recall@50 and Recall@100 respectively. Given the same features, TRACE also outperforms the state-of-the-art model by 2.8% at mAP. To sum up, our contributions are as follows:

1) We propose a new *detect-to-track* paradigm for video-level VidSGG, coined as Target Adaptive Context Aggregation Network (TRACE). This new perspective decouples the context modeling for relation prediction from the complicated low-level entity tracking. With this new paradigm, we provide a baseline method for VidSGG and gains improvement over the state-of-the-art method on ImageNet-VidVRD dataset.

2) As a pure frame-level VidSGG framework, TRACE presents a more modular framework to capture spatio-temporal context information for relation recognition than the previous methods, and obtains the best performance on Action Genome dataset.

3) In our TRACE, we propose an adaptive structure called as hierarchical relation tree (HRTree). By using HRTree, the efficient context information aggregation among candidates is enabled. Moreover, our experiment demonstrates that this module allows us to save memory for more parameters, thus resulting a better performance than a fully connected graph.

2. Related Work

Scene Graph Generation (SGG). Since the concept of scene graph was defined in [11], SGG task has become an important problem in computer vision. Along the research line, aggregating context information among interacted objects is quite effective for SGG. Xu et al. [41] constructed a primal graph and utilize GRU [2] to pass message between

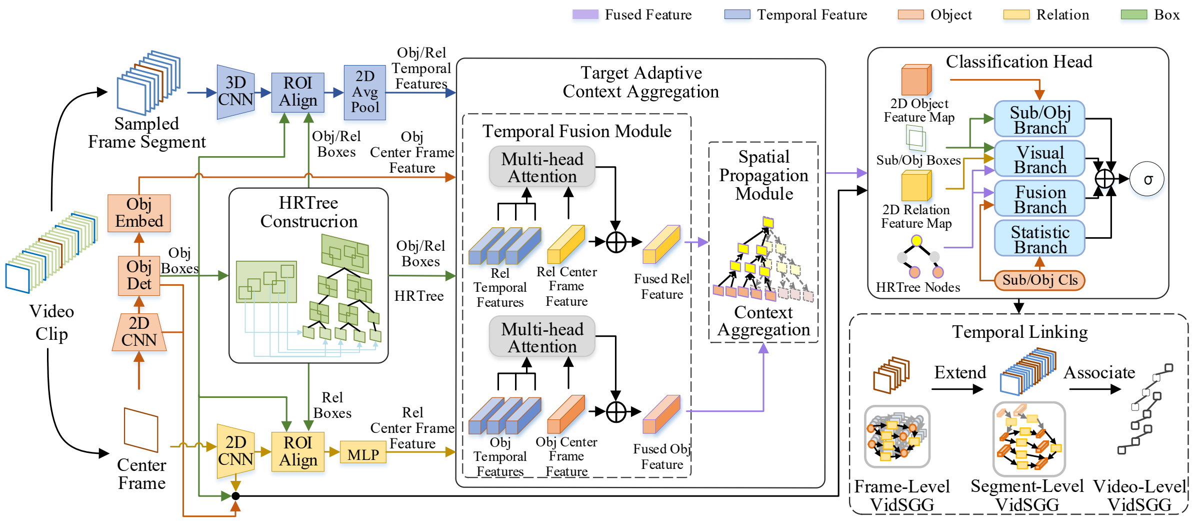


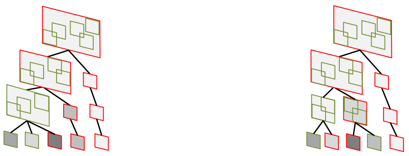
Figure 2. **TRACE framework.** Our TRACE framework is composed of feature extraction, hierarchical relation tree (HRTree) construction, context aggregation, classification head, and optional temporal linking, for VidSGG in a *detect-to-track* manner. Our model takes a clip as input and generates a spatio-temporal feature representation using a 3D CNN. The center frame is passed through an object detection network and a 2D CNN for static feature extraction. Based on these detection results, we build HRTree to organize relation candidates. The feature representation and HRTree are passed through a target-adaptive context aggregation module to obtain the contextualized representation for each relation candidate. This contextualized relation features is used to classify each candidate into relation classes. Finally, a simple temporal linking strategy is used to associate frame-level scene graphs to yield video-level results. The round-headed black arrow indicates transmitting the features directly. $\oplus$ denotes the operation of plus, and σ denotes the sigmoid function.

its nodes. Yang et al. [43] utilized graph convolutional networks [13] and attention mechanism [42] for context information aggregation. Li et al. [17] reduced the number of candidate relations by non-maximal suppression [6]. Tang et al. [32], Yin et al. [44] and Wang et al. [40] explored the application of tree structure in the domain of SGG from different aspects. In this paper, we argue that a better approach for graph construction is important for SGG frameworks. Therefore, we propose to combine the relation candidates hierarchically and adopt a tree structure for message passing between object and relation features directly, which is more memory efficient and differs from [32, 44, 40].

Video Scene Graph Generation (VidSGG). The concept of VidSGG was first proposed by Shang et al. [27] and they released a dataset named ImageNet-VidVRD. In [27], they used improved Dense Trajectories feature [37] to predict the pairwise relations in video segments and then associated these relation triplets into video level. Subsequently, several works focused on video-level VidSGG have been released [25, 34, 22, 30]. However, all these methods used the *track-to-detect* paradigm and required complicated preprocessing to link detections into tubes. Thus, they heavily depended on the tracking results and lacked the flexibility of capturing relations in a frame level for more accurate results. Recently, Ji et al. [10] released Action Genome dataset which only focuses on frame-level VidSGG, which prompted us to consider unifying these tasks in a concise way. In this paper, we present a new *detect-to-track* paradigm for VidSGG and could be used for both frame-level and video-level tasks of SGG. Our experiment results indicate that this new paradigm exhibit high flexibility and effectiveness for video-level VidSGG.

3. Technical Approach

Overview. As shown in Figure 2, we propose a new method for frame-level VidSGG, termed as *Target Adaptive Context Aggregation Network* (TRACE). The input of our model is a dense sampled short clip and its center frame. Our TRACE streamlines the VidSGG pipeline with the components of feature extraction, HRTree construction, context aggregation, relation classification, and optional temporal linking. *First*, objects are detected in the center frame. The spatial features in the center frame are extracted by using a 2D CNN and temporal features in the clip are extracted with a 3D CNN. Furthermore, the static object features are combined with word-embedding [24] for the subsequent blocks. *Second*, hierarchical relation tree (HRTree) is built to organize visual relation candidates in a compact and efficient way. *Third*, we perform target-adaptive context feature aggregation at a relatively low memory cost with the help of HRTree. Specifically, we devise a temporal attentive module for the fusion of temporal features. Then, a directional spatial aggregation module is responsible for propagating context information. *Finally*, a classification module is used for inferring the relation class of each relation candidate. Additionally, our method can be extended to the video level with a simple temporal association strategy.



(a) Scheme 1.

(b) Scheme 2.

Figure 3. The two schemes for HRTree construction. The shades of color represent the scores and the darker color means the higher score. The edges of centers are in red.

3.1. Hierarchical Relation Tree Construction

Hierarchical Relation Tree (HRTree) for relation candidates organization is built in a hierarchical bottom-up way. The leaf nodes in HRTree represent the objects detected in the center frame. The non-leaf nodes are derived from their child nodes and represent their composite relations. Specifically, HRTree is constructed in a progressive manner based on spatial proximity. Given the spatial coordinates of nodes in one layer, we use Gaussian kernel function to calculate the sum of pairwise similarity for each node:

$$\text{score}_k = \sum_i e^{-\|f_k - f_i\|^2}, \tag{1}$$

where score_k encodes the relative location information for node k and f represents the spatial coordinates. After obtaining the scores of nodes in one layer, we sort the nodes based on their scores and select part of them as the centers. Then, the other nodes are merged into the centers closest to them measured by their spatial union. Therefore, the updated centers form the parent layer of the current layer and this process is repeated until there is one node left. Euclidean distance is used as a measure of distance.

With regard to the selection of centers in each layer, as shown in Figure 3, we propose two implementation schemes: (1) From the highest to the lowest score, one node is chosen as the center every other node. (2) We fix the number of nodes to be rounded half of the total number of their child nodes. Then, we select half of the nodes from the part with the highest score and the others from the part with the lowest score as the centers. In our case, the number of visual relation candidates is $O(n)$, which means relation candidates greatly decrease in quantity compared with previous fully-connected graph, thus saving more computational and memory overhead for context aggregation.

3.2. Target Adaptive Context Aggregation

Temporal Fusion Module. After introducing the construction of HRTree, we are ready to describe how use this structure to guide context information aggregation. First, we describe the temporal context information fusion in this subsection and then the spatial context aggregation in next subsection. As analyzed above, temporal motion information is important to recognize some relations such as *wiping*. So, we follow the common practice of extracting temporal features with a box tube [5]. We use a 3D CNN to extract spatio-temporal features to provide motion information for relation candidates.

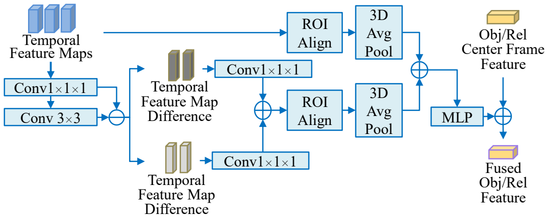


Figure 4. The temporal difference module for temporal fusion.

Specifically, for each non-leaf node, i.e., relation candidate, we extract a feature representation corresponding to this relation candidate from 3D CNN feature map. It is implemented by first stretching the candidate bounding box along time with repetition to form a tube. Then, we extract a feature at each time point with the corresponding box in the tube by using the standard RoI Align operation [8]. The resulting features across time are used for temporal information aggregation for current relation candidate. We propose two ways of fusing temporal information: (1) As shown in Figure 2, the multi-head attention mechanism [35] is applied to these temporal features with the spatial feature as query. It is essentially the weighted sum of 3D features and the weights are learnt adaptively based on 2D features. (2) As shown in Figure 4, a temporal difference operation is applied to the output of 3D backbone to extract motion features, and simple average pooling operation is employed for temporal fusion. In the experiment section (see Sec. 4.3), we show that these two kinds of temporal fusion modules are effective for the certain types of relations related to motion (e.g., *writing on*, *carrying*). However, for some short-term relation recognition, its improvement is not so evident.

Spatial Propagation Module. In this subsection, we describe the spatial context aggregation mechanism based on HRTree. Specifically, we adopt a group tree-GRU scheme for context aggregation in bidirectional progation manner. The features of nodes in HRTree are divided into multiple groups across the feature dimension. Then, features in each group of are fed into an independent tree-GRU [48]. In each tree-GRU, bottom-up feature aggregation is performed at first. Then, the top-down feature refinement which is equivalent to a common GRU [2] takes place. Subsequently, a multi-layer perceptron (MLP) is applied for the concatenation of features to yield the contextualized features. In experiment, we observe that this spatial propagation module is effective to aggregate spatial context information for relation recognition.

3.3. Classification Head

As shown in Figure 5, the classification head is responsible for relation inference. It consists of four branches and

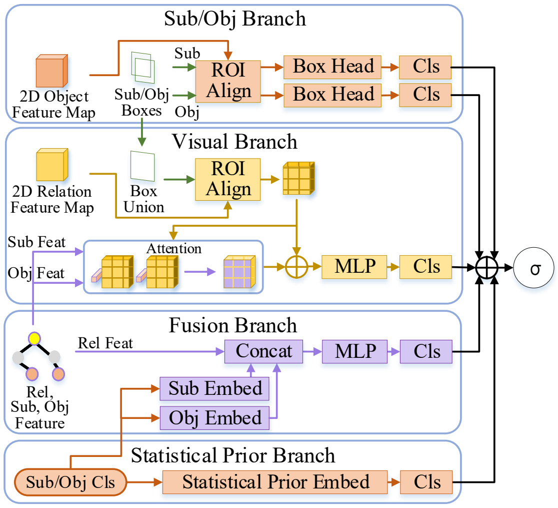


Figure 5. Illustration of the classification head.

each branch provides one result. The final score for classification is the sum of them followed by a sigmoid function.

Visual Branch. A relation feature map is generated by applying ROI Align [8] to the 2D CNN backbone's output with the union of pairwise object proposals. After obtaining each relation feature map, the feature vectors of the subject and object perform an attention mechanism on its dimension-reduced version. Specifically, the score map is first generated by the cosine similarity of the feature map and the feature vector on each pixel, and then the attention map is derived from the element-wise product between the feature map and the post-softmax score map. The attention maps and feature maps are used for classification.

Fusion Branch. We first input subject and object classification scores into word-embedding blocks [24]. Then, the embedded feature vectors are concatenated with the contextualized relation feature from the spatial propagation module for a classifier. The contextualized relation feature corresponding to the concatenated vector of subject and object belongs to their least common ancestors in HRTree.

Subject/Object Branch and Statistical Prior Branch. We adopt the same subject/object structure as [46]. Following [46, 45], we also employ the subject/object classification statistics as our input for better results.

3.4. Temporal Linking

Finally, we describe the temporal linking strategy to fuse frame-level scene graphs to the video-level results. The long video clip is first divided into overlapping video segments (e.g., 30 frames for one segment, 15 frames for the interval) and then the tracking is performed on each segment. The object trajectories obtained from tracking are used for this linking. As for one video segment, we sample a quarter of the frames with frame-level scene graphs for linking. If one triplet appears in only one frame, it is directly counted with its predicted score. For triplets with the same predicted categories in multiple frames, the triplet is counted once with summed scores if their subjects and objects belong to the same trajectory, respectively. As for the whole video, the triplets among two neighbouring segments are associated only if their predicted categories are the same and their viou of subject/object trajectories is beyond a threshold of 0.5 [27]. The video-level scores can be either average [27] or maximum [25]. In a greedy manner, high-scoring triplets take precedence over other ones during the association process.

4. Experiments

In this section, we present experimental results on two datasets: ImageNet-VidVRD [27] and Action Genome [10]. First, we report the evaluation settings and implementation details. Then, we show the ablation studies and comparisons to state-of-the-art methods.

4.1. Evaluation Settings

ImageNet-VidVRD (VidVRD). VidVRD [27] focuses on a wider range of relations not limited to human-object interaction. Especially, the subjects in VidVRD could be various categories more than just humans. VidVRD contains 35 object categories and 132 relation categories. Unlike the traditional SGG datasets, such as VG [14], multiple relations could occur between a subject and an object in VidVRD [27]. The annotations of VidVRD are spatio-temporal originally, so the label of one relation contains both spatial location and duration. Different from [27, 25], in our work, we assign the spatio-temporal annotations to each frame. After the conversion, the average number of relations and objects is 9.7 and 2.5 in each frame respectively. Furthermore, the number of relations in each object pair is about 2.0 on average. In the same way as [27, 25], Recalls and mAP of relation detection are used to evaluate our model. Relation Tagging [27] is also considered for comparison. Furthermore, consistent with [27], top 20 predicted relations for each pair of objects are kept for evaluation. The threshold for viewing a predicted box as a hit is 0.5.

Action Genome (AG). AG [10] is a dataset bridging human action and human-object relations. In AG, the relations are all human action and all subjects belong the category of *person*. AG is built on Charades [28] dataset so it contains a large number of indoor scenes. The number of object and relation categories in AG is 36 and 26 respectively. Similar to VidVRD [27], multiple relations may exist between subjects and objects in AG. After preprocessing, the number of relations and objects is 7.3 and 3.2 in each frame on average respectively. Furthermore, the number of relations in each object pair is 3.3 on average. Furthermore, the triplets with overlapped subject bounding box and object bounding box in AG is more than 85%. In line with [10], we adopt three modes for the evaluation on AG: scene graph detection (SGDet), scene graph classification (SGCls) and predicate classification (PredCls) [23]. The traditional specific metric for the three standard modes is Recall. However, due to the imbalanced distribution of relations, we introduce mean Recall (mR) [32], mAP$_{rel}$ and wmAP$_{rel}$ [46] to AG. The predicted boxes that have an IoU of at least 0.5 with the ground-truth boxes are counted as a hit. It is worth noting that since the relations in AG are all human-object interaction, PredCls and SGCls in our benchmark provide not only the ground-truth object boxes but also the potentially related pairs. Due to the multiple relations, the graph constraint which restricts each pair of objects to one prediction of triplet is not suitable here. Additionally, to avoid the situation where the predictions randomly hit the ground-truth triplets, each pair of objects is only allowed corresponding to k predictions and k is set to 6 or 7.

4.2. Implementation Details

The input of our model is a video segment sampled from the video clip and its center frame. The segment except the center frame is composed of $T = 8$ neighboring frames of the center frame with a temporal stride $v = 4$.

Loss. The loss is the weighted sum of the binary cross entropy for relations and cross entropy for the objects. The weight for relations is 1.0 while the weight for objects is 0.05. The relations are predicted by the classification module while the objects participating in the context aggregation and subsequent blocks are predicted by a classifier which is a duplicate of the one in Faster R-CNN [26]. Notably, when testing, this classifier is not activated.

Training. We use RTX 2080ti with 11G GPU memory for training. In line with [10, 27], Faster R-CNN [26] with ResNet [9] pretrained on COCO [21] is first trained on each dataset. We utilize 2D ResNet-50 [9] to extract relation feature on the center frame, and use I3D ResNet-50 [1] pretrained on Kinetics [12] to extract temporal information. All layers in the backbone for object feature extraction are frozen when training TRACE. We use SGD with momentum to optimize TRACE with batch size 1. The initial learning rate for AG and VidVRD is set to 0.01 and 0.025 respectively. The ratio between the foreground relations and the background relations is 1:3, and 2048 relations with 512 objects are used for training. A triplet is defined as foreground if its relation and object classes are identical to that of a ground truth and its objects have an overlap of $iou > 0.5$ with that of the ground-truth respectively. The other triplets are background. We randomly select at most m foreground and k background relations. We set m as 512 and ensure m + k = 2048. In particular, for VidVRD, due to the conversion from video segments to frames, we randomly select 15.9% frames from the training set to train TRACE.

Testing. Top 100 object proposals are kept after object detection and per-class non-maximal suppression [6] with an IoU of 0.5 is used in each frame. Due to the most objects in AG touching each other, following [46], we only predict the relations in pairs with overlapped bounding boxes for SGDet. However, this trick is not applied in VidVRD.

4.3. Ablation Studies

We carry out ablation studies on VidVRD dataset. Recalls and mAP are adopted for evaluations. Aside from these, a per category breakdown experiment in Table 1 illustrates the effectiveness of our temporal fusion module.

Study on the context aggregation structure. We begin our ablation study by exploring the effectiveness of the context aggregation structure in TRACE. We implement fairly comparable models by removing this module and changing the scheme of HRTree. However, due to the large memory cost of the fully connected graph (FC-G) which forms a complete bipartite graph with its pairwise relation nodes, we reduce the parameters of FC-G and report its performance. In Table 2, TRACE with HRTree of each scheme outperforms the model without fusion under the three metrics. Due to the reduction of parameters, the performance of FC-G is quite lower than the others in this table. TRACE with HRTree of scheme 1 consumes 10.5GB and 6.6GB GPU memory for training and testing per batch, respectively. However, we found that our model with FC-G and complete parameters consumes too memory to train, so we reduce the parameters to the same level for fair comparison between FC-G and HRTree. Table 4 shows the results of FC-G is quite comparable to HRTree and the number of parameters is important. It illustrates that HRTree reduces the memory cost without much performance decrease. Furthermore, we compare our model without temporal fusion module to RelDN [46] in Table 7. It demonstrates the effectiveness of our pure frame-level context aggregation.

Study on the temporal fusion module. We compare the different schemes of temporal fusion module. Moreover, We report the result of TRACE without the temporal fusion. In Table 3, The performance of TRACE with temporal fusion module of scheme 1 at Recalls is better than the model with temporal fusion of scheme 2 and the model without temporal fusion, but it is worse at mAP. For further research, we conduct a per category breakdown experiment. As shown in Table 1, the temporal fusion scheme 1 drastically improves the performance at servel relation categories such as *writing on*, *lying on* and *carrying*. However, the attention across temporal dimension for each relation candidates may disturb the spatial information, which leads to decreases in *beneath* and *standing on*. Scheme 2 does not adopt the adaptive temporal information aggregation and the increases on each category are not salient.

Study on the *detect-to-track* framework. We conduct experiments to show the effectiveness of our *detect-to-track*

Temp Fusion	beneath	carrying	eating	lying on	standing on	touching	wiping	writing on
-	47.65	12.51	19.98	19.55	45.40	35.39	4.71	25.63
1	46.70(-0.95)	14.96(+2.45)	20.61(+0.63)	26.64(+7.09)	44.54(-0.86)	36.76(+1.37)	6.27(+1.56)	26.71(+1.08)
2	47.7(+0.05)	13.74(+1.23)	19.89(-0.09)	19.10(-0.45)	46.08(+0.68)	35.79(+0.40)	5.10(+0.39)	27.08(+1.45)

Table 1. The Recall@20 (%) of partial relation categories on AG [10] with top 6 Predictions for each pair. The format of values except the first line is the model's output with different temporal fusion scheme and its difference compared to the first line.

Fusion	mAP	R@50	R@100
FC-G	28.11	16.96	21.78
-	28.62	17.48	23.56
Tree-2	29.28	17.70	22.65
Tree-1	**29.32**	**18.45**	**23.85**

Table 2. **Study on the context fusion structure**. We compare the models with different context aggregation methods.

Fusion	mAP	R@50	R@100
-	**29.94**	18.01	23.56
Temp-2	29.80	18.08	23.23
Temp-1	29.32	**18.45**	**23.85**

Table 3. **Study on the temporal fusion structure**. We compare the models with different temporal fusion modules.

Fusion	R@20	R@50	mR@20	mR@50
FC-G*	32.32	44.63	**27.60**	38.17
Tree-1*	32.24	44.60	26.94	37.60
Tree-1	**33.41**	**45.67**	27.58	**38.61**

Table 4. We compare the tree structure in our model with ResNet-50-FPN [19] when the parameters are reduced at SGDet on AG. * means the parameters of the model are reduced.

Group Number	mAP	R@50	R@100
2	**29.81**	**18.51**	23.72
4	29.32	18.45	**23.85**

Table 5. **Study on the group number**. We compare TRACE with different group number in the context aggregation structure.

Model	mAP	R@50	R@100
VidVRD-C [27]	7.17	4.36	5.36
Liu's [22](Obj)	14.01	**8.47**	**11.00**
Ours	**15.06**	7.67	10.32

Table 6. **Study on the framework**. We compare TRACE to other models with only object features.

Model	R@20	R@50	R@100
RelDN [46]	23.95	35.39	42.91
Ours	**24.80**	**36.52**	**45.33**

Table 7. **Study on the context aggregation structure without temporal fusion**. We compare our model without temporal fusion to RelDN [46] under the frame-level metrics on VidVRD [27].

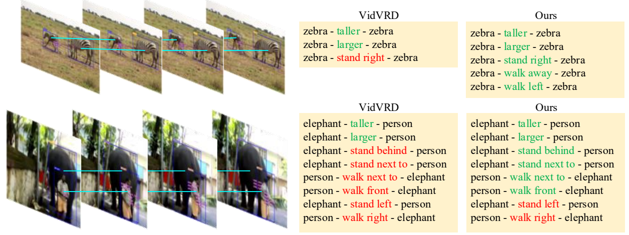

VidVRD
zebra - taller - zebra
zebra - larger - zebra
zebra - stand right - zebra
Ours
zebra - taller - zebra
zebra - larger - zebra
zebra - stand right - zebra
zebra - walk away - zebra
zebra - walk left - zebra
VidVRD
elephant - taller - person
elephant - larger - person
elephant - stand behind - person
elephant - stand next to - person
person - walk next to - elephant
person - walk front - elephant
elephant - stand left - person
person - walk right - elephant
Ours
elephant - taller - person
elephant - larger - person
elephant - stand behind - person
elephant - stand next to - person
person - walk next to - elephant
person - walk front - elephant
elephant - stand left - person
person - walk right - elephant

Figure 6. The Visualization of predictions hitting the ground-truth from VidVRD and TRACE. The correct results are marked in green. The relations with incomplete duration are marked in red.

framework. Since almost all previous works are in the manner of *track-then-detect* [27, 25, 22], we compare TRACE to VidVRD-C [27] and Liu's [22] with only object features. In Table 6, mAP of our model without the fusion is better than Liu's [22] while the Recalls are comparable. Thus, the performance of our framework is comparable to that of the *track-then-detect* models, but more flexible.

Study on the selection for group number. The operation of grouping can reduce the computational complexity. In Table 5, we find that the group number only affect the performance at mAP on VidVRD evidently. Moreover, during training, we found that the one group version consumed too memory to run.

4.4. Comparison with the State of the Art

ImageNet-VidVRD (VidVRD). As shown in Table 8, different conditions lead to different best performing models on VidVRD. With the ground-truth trajectories and the same association algorithm, our model outperforms the best method, VRD-GCN [25], by 2.8%, 1.0% and 2.1% at mAP, Recall@50 and Recall@100 respectively. Furthermore, under the condition of using trajectories provided by [27] and the basic association [27], we push the performance at mAP, Recall@50, Recall@100 to 15.8%, 8.1% and 10.3% respectively. Furthermore, like the siamese association in VRD-GCN [25], we modify the calculation of scores in the basic association algorithm from averaging to maximizing and get improvements under all metrics. Compared to the state-of-the-art method, Liu's [22], with both object features and I3D features, TRACE is comparable to it at Recalls, and be-

Method	Relation Detection			Relation Tagging		
	mAP	R@50	R@100	P@1	P@5	P@10
VidVRD gt [27]	15.53	12.51	16.55	43.50	29.70	23.20
VRD-GCN gt [25]	26.52	17.50	21.80	62.50	44.20	31.10
Ours gt	**29.32**	**18.45**	**23.85**	**65.50**	**45.60**	**33.75**
VidVRD† [27]	8.58	5.54	6.37	43.00	28.90	20.80
GSTEG [34]	9.52	7.05	7.67	51.50	39.50	28.23
MHRA [3]	13.27	6.82	7.39	41.00	28.70	20.95
VRD-GCN† [25]	14.23	7.43	8.75	**59.50**	40.50	27.85
Ours†	**15.81**	**8.07**	**10.30**	56.00	**44.50**	**32.95**
Liu's [22]	14.81	**9.14**	**11.39**	55.50	38.90	28.90
Ours‡	**17.57**	9.08	11.15	**61.00**	**45.30**	**33.50**

Table 8. The metrics [27] (%) of various models on VidVRD. For fair comparison, we compare our method with [22] by using the object features and I3D features. † denotes using the basic temporal linking proposed in [27] with average scores, while ‡ means using maximal scores.

Top k Predictions for Each Pair	Method	PredCls				SGCls				SGDet			
		image		video		image		video		image		video	
		R@20	R@50	R@20	R@50	R@20	R@50	R@20	R@50	R@20	R@50	R@20	R@50
k=7	Freq Prior [45]	87.95	93.02	86.01	88.59	45.10	48.87	44.47	46.39	34.41	44.34	32.50	41.11
	G-RCNN [43]	88.73	93.73	86.28	88.93	45.57	49.75	45.11	47.22	34.28	44.47	32.60	41.29
	RelDN [46]	90.89	96.09	88.77	91.43	46.47	50.31	45.87	47.78	34.92	45.27	33.18	42.10
	Ours	**91.60**	**96.35**	**89.31**	**91.72**	**46.66**	**50.46**	**46.03**	**47.92**	**35.09**	**45.34**	**33.38**	**42.18**
k=6	Freq Prior [45]	85.89	89.43	83.33	84.99	44.90	47.15	43.57	44.63	34.47	43.69	32.38	40.24
	G-RCNN [43]	87.03	90.60	84.02	85.74	45.82	48.31	44.60	45.77	34.60	43.98	32.75	40.65
	RelDN [46]	89.63	93.56	87.01	88.86	46.76	49.11	45.48	46.57	35.22	44.94	33.39	41.64
	Ours	**90.34**	**93.94**	**87.56**	**89.24**	**47.00**	**49.32**	**45.71**	**46.79**	**35.41**	**45.06**	**33.59**	**41.76**

Table 9. Recall (%) of various models with ResNet-101 [9] on AG. For fair comparison, we reproduce the methods based on our object detection and the same training strategy, and our model performs better than the others.

Method	PredCls		SGCls		SGDet	
	mR@20	mR@50	mR@20	mR@50	mR@20	mR@50
Freq Prior [45]	55.17	63.67	34.30	36.96	24.89	34.07
G-RCNN [43]	56.32	61.31	36.19	38.29	27.79	34.99
RelDN [46]	59.81	63.47	39.92	41.93	30.39	39.53
Ours	**61.80**	**65.37**	**41.19**	**43.21**	**30.84**	**40.12**

Table 10. Mean recall [32] (%) of various models with ResNet-101 [9] on all images in AG. The number of triplets per frame is set to a limit of 50 and top 6 predictions for each pair are kept when evaluating.

Method	PredCls		SGCls		SGDet	
	mAP_r	wmAP_r	mAP_r	wmAP_r	mAP_r	wmAP_r
Freq Prior [45]	33.10	65.92	14.29	22.68	9.45	15.58
G-RCNN [43]	41.21	70.89	17.64	22.53	11.76	15.90
RelDN [46]	50.08	72.26	20.07	23.88	12.93	15.94
Ours	**53.27**	**75.45**	**20.71**	**24.61**	**13.43**	**16.56**

Table 11. mAP_{rel} and wmAP_{rel} (%) of various models with ResNet-101 [9] on all images in AG. The number of triplets per frame is set to a limit of 50 and top 6 predictions for each pair are kept when evaluating. mAP_r and wmAP_r indicate mAP_{rel} and wmAP_{rel}, respectively.

yond 2.8% at mAP. Following [27, 25], we also report the results on relation tagging [27] and TRACE achieves good performance under different conditions.

Action Genome (AG). The results are summarized in Table 9, Table 10 and Table 11. However, in Table 9, the difference between the performance of various models at SGDet is not obvious. We analyze that the results at SGDet heavily depends on the object detector while the objects labeled in AG is far fewer than those in standard object detection datasets. Meanwhile, for fair comparison, we use the same detector for all methods and thus the difference is quite small. The performance at Recalls of SGDet in this dataset tends to be saturate. Therefore, PredCls and SGCls are more significant than SGDet in AG. Moreover, mAP_{rel}, wmAP_{rel} and mR are more balanced metrics than Recalls, and are better metrics for revealing the gap between the performance of methods. Specifically, in Table 10 and Table 11, TRACE outperforms RelDN [46] by 3.2% and 0.6% at mAP_{rel} of PredCls and SGCls, by 2.0% and 1.3% at mRs of PredCls and SGCls on average.

4.5. Qualitative Results

Our qualitative results are shown in Figure 6. VidVRD detects few relation triplets in some scenes and fails to detect enough relations containing motion information, such as *walk away* and *walk left*. We analyze that its temporal fusion structure fails to find fine-grained changes of objects in the scenes with slow motion compared to ours. Moreover, the relations detected in VidVRD lasts for shorter duration than TRACE. It illustrates that our *detect-to-track* framework performs better than the *track-then-detect* one, due to the less disturbance of noisy tracking results.

5. Conclusion

In this paper we have proposed a modular framework, coined as Target Adaptive Context Aggregation Network (TRACE) for frame-level VidSGG. To adaptively and efficiently capture spatio-temporal context information, we design a new hierarchical relation tree to guide temporal attentive fusion and spatial message propagation. Our method combined with a simple temporal association strategy yields a modular video-level VidSGG baseline, obtaining the best performance without using complex tracking features under video-level metrics on ImageNet-VidVRD. For pure frame-level VidSGG task, TRACE still achieves new state-of-the-art results on benchmarks of Action Genome.

Acknowledgements. This work is supported by National Natural Science Foundation of China (No. 62076119, No. 61921006), Program for Innovative Talents and Entrepreneur in Jiangsu Province, and Collaborative Innovation Center of Novel Software Technology and Industrialization, Tencent AI Lab Rhino-Bird Focused Research Program (No. JR202025).

References

[1] João Carreira and Andrew Zisserman. Quo vadis, action recognition? A new model and the kinetics dataset. In *CVPR*, pages 4724–4733. IEEE Computer Society, 2017. 1, 6

[2] Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *EMNLP*, pages 1724–1734. ACL, 2014. 2, 4

[3] Donglin Di, Xindi Shang, Weinan Zhang, Xun Yang, and Tat-Seng Chua. Multiple hypothesis video relation detection. In *BigMM*, pages 287–291. IEEE, 2019. 7

[4] Jiyang Gao, Chen Sun, Zhenheng Yang, and Ram Nevatia. TALL: temporal activity localization via language query. In *ICCV*, pages 5277–5285. IEEE Computer Society, 2017. 1

[5] Rohit Girdhar, João Carreira, Carl Doersch, and Andrew Zisserman. Video action transformer network. In *CVPR*, pages 244–253. Computer Vision Foundation / IEEE, 2019. 1, 4

[6] Ross B. Girshick. Fast R-CNN. In *ICCV*, pages 1440–1448. IEEE Computer Society, 2015. 3, 6

[7] Sergio Guadarrama, Niveda Krishnamoorthy, Girish Malkarnenkar, Subhashini Venugopalan, Raymond J. Mooney, Trevor Darrell, and Kate Saenko. Youtube2text: Recognizing and describing arbitrary activities using semantic hierarchies and zero-shot recognition. In *ICCV*, pages 2712–2719. IEEE Computer Society, 2013. 1

[8] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. Mask R-CNN. In *ICCV*, pages 2980–2988. IEEE Computer Society, 2017. 4, 5

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778. IEEE Computer Society, 2016. 6, 8

[10] Jingwei Ji, Ranjay Krishna, Fei-Fei Li, and Juan Carlos Niebles. Action genome: Actions as compositions of spatio-temporal scene graphs. In *CVPR*, pages 10233–10244. IEEE, 2020. 1, 2, 3, 5, 6, 7

[11] Justin Johnson, Ranjay Krishna, Michael Stark, Li-Jia Li, David A. Shamma, Michael S. Bernstein, and Fei-Fei Li. Image retrieval using scene graphs. In *CVPR*, pages 3668–3678. IEEE Computer Society, 2015. 2

[12] Will Kay, João Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, Mustafa Suleyman, and Andrew Zisserman. The kinetics human action video dataset. *CoRR*, abs/1705.06950, 2017. 6

[13] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *ICLR (Poster)*. OpenReview.net, 2017. 3

[14] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A. Shamma, Michael S. Bernstein, and Li Fei-Fei. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *Int. J. Comput. Vis.*, 123(1):32–73, 2017. 5

[15] Jie Lei, Licheng Yu, Mohit Bansal, and Tamara L. Berg. TVQA: localized, compositional video question answering. In *EMNLP*, pages 1369–1379. Association for Computational Linguistics, 2018. 1

[16] Jie Lei, Licheng Yu, Tamara L. Berg, and Mohit Bansal. TVR: A large-scale dataset for video-subtitle moment retrieval. In *ECCV (21)*, volume 12366 of *Lecture Notes in Computer Science*, pages 447–463. Springer, 2020. 1

[17] Yikang Li, Wanli Ouyang, Bolei Zhou, Jianping Shi, Chao Zhang, and Xiaogang Wang. Factorizable net: An efficient subgraph-based framework for scene graph generation. In *ECCV (1)*, volume 11205 of *Lecture Notes in Computer Science*, pages 346–363. Springer, 2018. 1, 3

[18] Yixuan Li, Zixu Wang, Limin Wang, and Gangshan Wu. Actions as moving points. In *ECCV (16)*, volume 12361 of *Lecture Notes in Computer Science*, pages 68–84. Springer, 2020. 1

[19] Tsung-Yi Lin, Piotr Dollár, Ross B. Girshick, Kaiming He, Bharath Hariharan, and Serge J. Belongie. Feature pyramid networks for object detection. In *CVPR*, pages 936–944. IEEE Computer Society, 2017. 7

[20] Tianwei Lin, Xiao Liu, Xin Li, Errui Ding, and Shilei Wen. BMN: boundary-matching network for temporal action proposal generation. In *ICCV*, pages 3888–3897. IEEE, 2019. 1

[21] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. In *ECCV (5)*, volume 8693 of *Lecture Notes in Computer Science*, pages 740–755. Springer, 2014. 6

[22] Chenchen Liu, Yang Jin, Kehan Xu, Guoqiang Gong, and Yadong Mu. Beyond short-term snippet: Video relation detection with spatio-temporal global context. In *CVPR*, pages 10837–10846. IEEE, 2020. 1, 2, 3, 7

[23] Cewu Lu, Ranjay Krishna, Michael S. Bernstein, and Fei-Fei Li. Visual relationship detection with language priors. In *ECCV (1)*, volume 9905 of *Lecture Notes in Computer Science*, pages 852–869. Springer, 2016. 6

[24] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *EMNLP*, pages 1532–1543. ACL, 2014. 3, 5

[25] Xufeng Qian, Yueting Zhuang, Yimeng Li, Shaoning Xiao, Shiliang Pu, and Jun Xiao. Video relation detection with spatio-temporal graph. In *ACM Multimedia*, pages 84–93. ACM, 2019. 1, 2, 3, 5, 7, 8

[26] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. Faster R-CNN: towards real-time object detection with region proposal networks. volume 39, pages 1137–1149, 2017. 6

[27] Xindi Shang, Tongwei Ren, Jingfan Guo, Hanwang Zhang, and Tat-Seng Chua. Video visual relation detection. In *ACM Multimedia*, pages 1300–1308. ACM, 2017. 1, 2, 3, 5, 6, 7, 8

[28] Gunnar A. Sigurdsson, Gül Varol, Xiaolong Wang, Ali Farhadi, Ivan Laptev, and Abhinav Gupta. Hollywood in homes: Crowdsourcing data collection for activity understanding. In *ECCV (1)*, volume 9905 of *Lecture Notes in Computer Science*, pages 510–526. Springer, 2016. 5

[29] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *NIPS*, pages 568–576, 2014. 1

[30] Zixuan Su, Xindi Shang, Jingjing Chen, Yu-Gang Jiang, Zhiyong Qiu, and Tat-Seng Chua. Video relation detection via multiple hypothesis association. In *ACM Multimedia*, pages 3127–3135. ACM, 2020. 1, 3

[31] Jing Tan, Jiaqi Tang, Limin Wang, and Gangshan Wu. Relaxed transformer decoders for direct action proposal generation. In *ICCV*, 2021. 1

[32] Kaihua Tang, Hanwang Zhang, Baoyuan Wu, Wenhan Luo, and Wei Liu. Learning to compose dynamic tree structures for visual contexts. In *CVPR*, pages 6619–6628. Computer Vision Foundation / IEEE, 2019. 2, 3, 6, 8

[33] Makarand Tapaswi, Yukun Zhu, Rainer Stiefelhagen, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. Movieqa: Understanding stories in movies through question-answering. In *CVPR*, pages 4631–4640. IEEE Computer Society, 2016. 1

[34] Yao-Hung Hubert Tsai, Santosh Kumar Divvala, Louis-Philippe Morency, Ruslan Salakhutdinov, and Ali Farhadi. Video relationship reasoning using gated spatio-temporal energy graph. In *CVPR*, pages 10424–10433. Computer Vision Foundation / IEEE, 2019. 1, 3, 7

[35] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NIPS*, pages 5998–6008, 2017. 4

[36] Subhashini Venugopalan, Marcus Rohrbach, Jeffrey Donahue, Raymond J. Mooney, Trevor Darrell, and Kate Saenko. Sequence to sequence - video to text. In *ICCV*, pages 4534–4542. IEEE Computer Society, 2015. 1

[37] Heng Wang and Cordelia Schmid. Action recognition with improved trajectories. In *ICCV*, pages 3551–3558. IEEE Computer Society, 2013. 3

[38] Limin Wang, Wei Li, Wen Li, and Luc Van Gool. Appearance-and-relation networks for video classification. In *CVPR*, pages 1430–1439. IEEE Computer Society, 2018. 1

[39] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks for action recognition in videos. *IEEE Trans. Pattern Anal. Mach. Intell.*, 41(11):2740–2755, 2019. 1

[40] Wenbin Wang, Ruiping Wang, Shiguang Shan, and Xilin Chen. Sketching image gist: Human-mimetic hierarchical scene graph generation. 12358:222–239, 2020. 3

[41] Danfei Xu, Yuke Zhu, Christopher B. Choy, and Li Fei-Fei. Scene graph generation by iterative message passing. In *CVPR*, pages 3097–3106. IEEE Computer Society, 2017. 1, 2

[42] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron C. Courville, Ruslan Salakhutdinov, Richard S. Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *ICML*, volume 37 of *JMLR Workshop and Conference Proceedings*, pages 2048–2057. JMLR.org, 2015. 3

[43] Jianwei Yang, Jiasen Lu, Stefan Lee, Dhruv Batra, and Devi Parikh. Graph R-CNN for scene graph generation. In *ECCV (1)*, volume 11205 of *Lecture Notes in Computer Science*, pages 690–706. Springer, 2018. 1, 3, 8

[44] Guojun Yin, Lu Sheng, Bin Liu, Nenghai Yu, Xiaogang Wang, Jing Shao, and Chen Change Loy. Zoom-net: Mining deep feature interactions for visual relationship recognition. In *ECCV (3)*, volume 11207 of *Lecture Notes in Computer Science*, pages 330–347. Springer, 2018. 3

[45] Rowan Zellers, Mark Yatskar, Sam Thomson, and Yejin Choi. Neural motifs: Scene graph parsing with global context. In *CVPR*, pages 5831–5840. IEEE Computer Society, 2018. 5, 8

[46] Ji Zhang, Kevin J. Shih, Ahmed Elgammal, Andrew Tao, and Bryan Catanzaro. Graphical contrastive losses for scene graph parsing. In *CVPR*, pages 11535–11543. Computer Vision Foundation / IEEE, 2019. 2, 5, 6, 7, 8

[47] Yue Zhao, Yuanjun Xiong, Limin Wang, Zhirong Wu, Xiaoou Tang, and Dahua Lin. Temporal action detection with structured segment networks. *Int. J. Comput. Vis.*, 128(1):74–95, 2020. 1

[48] Yao Zhou, Cong Liu, and Yan Pan. Modelling sentence pairs with tree-structured attentive encoder. In *COLING*, pages 2912–2922. ACL, 2016. 4