Classification-Then-Grounding:
Reformulating Video Scene Graphs as Temporal Bipartite Graphs

Kaifeng Gao[†], Long Chen[‡*], Yulei Niu[‡], Jian Shao[†], Jun Xiao[†]

[†]Zhejiang University, [‡]Columbia University

{kite_phone,jshao,junx}@zju.edu.cn {zjuchenlong,yn.yuleiniu}@gmail.com

Abstract

*Today's VidSGG models are all proposal-based methods, i.e., they first generate numerous paired subject-object snippets as proposals, and then conduct predicate classification for each proposal. In this paper, we argue that this prevalent proposal-based framework has three inherent drawbacks: 1) The ground-truth predicate labels for proposals are partially correct. 2) They break the high-order relations among different predicate instances of a same subject-object pair. 3) VidSGG performance is upper-bounded by the quality of the proposals. To this end, we propose a new **classification-then-grounding** framework for VidSGG, which can avoid all the three overlooked drawbacks. Meanwhile, under this framework, we reformulate the video scene graphs as temporal bipartite graphs, where the entities and predicates are two types of nodes with time slots, and the edges denote different semantic roles between these nodes. This formulation takes full advantage of our new framework. Accordingly, we further propose a novel **BI**partite **G**raph based SGG model: **BIG**. It consists of a classification stage and a grounding stage, where the former aims to classify the categories of all the nodes and the edges, and the latter tries to localize the temporal location of each relation instance. Extensive ablations on two VidSGG datasets have attested to the effectiveness of our framework and BIG. Code is available at* https://github.com/Dawn-LX/VidSGG-BIG.

1. Introduction

To bridge the gap between vision and other modalities (*e.g.*, language), a surge of interests in our community start to convert the vision data into graph-structured representations, called **scene graphs** [20]. Scene graphs are visually-grounded graphs, where the nodes and edges represent object instances (or entities) and their pairwise visual relations (predicates), respectively. Due to the inherent interpretabil-
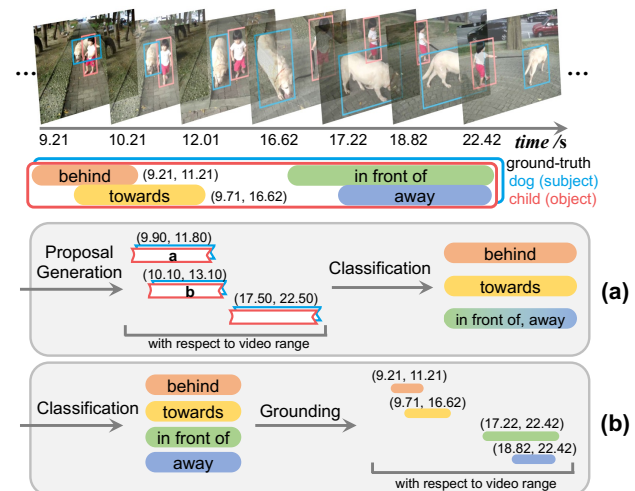

Figure 1. **(a)**: The pipeline of **proposal-based** framework. Given a video, it first generates numerous proposals (with different time slots), and then conducts predicate classification for each proposal. **(b)**: The pipeline of **classification-then-grounding** framework. It first conducts predicate classification based on the whole tracklet pair, and then grounds all the predicted relation instances.

ity, scene graphs have been widely used in numerous downstream tasks to help boost model performance, *e.g.*, captioning [6, 10, 47], grounding [8, 23], and QA [9, 11, 19, 26].

Video Scene Graph Generation (**VidSGG**) has achieved significant progress over the recent years. Currently, almost all existing VidSGG models are proposal-based[1]. Specifically, they can be categorized into two groups: 1) *Segment-proposal based*: They first cut the video into short segments and detect object tracklets in each segment to compose segment proposals, then classify predicates in each proposal and merge all predicted relation triplets (*i.e.*, ⟨subject, predicate, object⟩) among adjacent segments [28, 32, 33]. However, they fail to exploit the long-term context in the video (or tracklets) due to the limits of short segments. 2) *Tracklet-proposal based*: They directly detect tracklets in

---

*Long Chen is the corresponding author. This work started when LC at Zhejiang University, and YN at Nanyang Technological University.

---

[1]We use *proposals* to represent paired subject-object tracklet segments.

the whole video and generate tracklet proposals by sliding-windows [22] or confidence splitting [13], and then conduct predicate classification for each proposal.

Although these proposal-based methods have dominated the performance on VidSGG datasets, it is worth noting that this prevalent framework has three inherent drawbacks:

1. The ground-truth predicate labels for proposals are partially correct. By "partially", we mean that the ground-truth predicate labels sometimes are *WRONG*. Specifically, following the IoU-based strategy in object detection, existing proposal-based models all assign predicate labels to proposals based on the volume IoU (vIoU). This strategy naturally discards some "ground-truth" predicates if their vIoUs are less than the threshold. As shown in Figure 1(a), two relations `behind` and `towards` happen simultaneously on multiple frames inside both proposal$_a$ and proposal$_b$, but the assigned predicate label for proposal$_a$ is only `behind` (and `towards` for proposal$_b$)[2]. Meanwhile, once a predicate label is assigned to the proposal, they assume this relation should last for the whole proposal (*i.e.*, it happens in all the frames of the proposal). Obviously, one of the negative impacts of this issue is that the ground-truth labels for two highly-overlapped proposals (proposal$_{a/b}$) may be totally different, and this inconsistency hurts the model training.

2. They break the high-order relations among different predicate instances of a same subject-object pair. Due to the nature of videos, there are always multiple relations happening between a same subject-object pair, and these relations can serve as critical context (or inductive bias) to benefit the predictions of other relations. For example, `behind`, `towards`, and `away` always happen sequentially between `dog` and `child`. Instead, proposal-based methods explicitly break these high-order relations by pre-cutting tracklets, and classify predicates independently in each proposal[3].

3. VidSGG performance is upper-bounded by the quality of the proposals. The VidSGG performance is sensitive to the heuristic rules for proposal generation (*e.g.*, the sizes or number of proposals). Meanwhile, to achieve higher recalls, they always generate excessive proposals, which significantly increases the computation complexity.

In this paper, we propose a *classification-then-grounding* framework for VidSGG, which can avoid all the mentioned drawbacks in proposal-based methods. Specifically, we first conduct predicate classification based on the whole tracklets, and then ground each predicted predicate instance (Figure 1(b)). Compared to proposal-based methods, we regard all the relations happen between the two tracklets as ground-truth predicate labels (*e.g.*, `behind`, `towards`, `away`, and `in-front-of` are all ground-truth predicates for `dog` and

---
[2]For proposal$_a$, its vIoU with predicate `towards` < 0.5 and its vIoU with predicate `behind` > 0.5. The situation is opposite for proposal$_b$.

[3]Although a few proposal-based models start to resort to some context modeling techniques to remedy this weakness, we claim that the proposal-based framework itself overlooks and breaks these high-order relations.
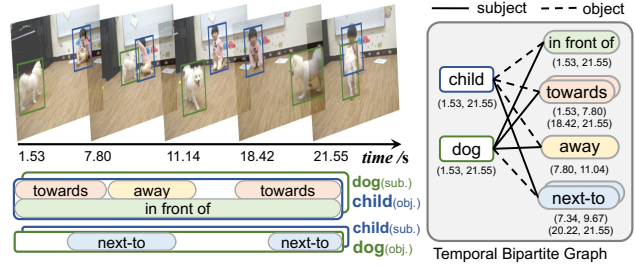

Figure 2. **Left**: A video example and its ground-truth visual relation triplets. **Right**: The corresponding temporal bipartite graph. Comparisons with existing formulation are left in appendix.

`child`). Our framework not only provides more accurate ground-truth predicate labels, but also preserves the ability to utilize high-order relations among predicates. Moreover, it avoids superfluous proposals and heuristic rules.

Under this framework, we propose to reformulate video scene graphs as temporal bipartite graphs, where the entities and predicates are two types of nodes with time slots, and the edges denote different semantic roles (*i.e.*, `subject` and `object`) between these nodes (Figure 2). Each entity node is an object tracklet, and its time slot is the temporal range of this tracklet. Each predicate node is a set of relation *instances* between two entities with the same predicate category, where each time slot denotes the temporal range of each relation instance (*e.g.*, predicate node `towards` in Figure 2 has two time slots). Thus, each entity node can be linked with multiple predicate nodes to represent multiple relations involved, and each predicate node can be linked with at most one entity node for each role. This formulation can not only be easily extended to more general relations with more semantic roles [50], but also avoid exhaustively enumerating all entity pairs for predicate prediction.

Accordingly, we propose a **BI**partite **G**raph based model **BIG**, which consists of a *classification* stage and a *grounding* stage. Specifically, the former aims to classify the categories of all the nodes and edges, and the latter tries to localize the temporal location of each relation instance. **For the classification stage**, it is a Transformer-based model, where the inputs for the encoder and decoder are tracklet features and learnable predicate embeddings, respectively. To distinguish different semantic roles, we also propose a role-aware cross-attention which introduces role-wise distinctions into predicate embeddings. **For the grounding stage**, we regard the triplet categories of each predicate node as a language query (*e.g.*, ⟨`dog`, `towards`, `child`⟩ in Figure 2), and ground this language query in the video. Since each relation category may happen multiple times between two tracklets, we design a multi-instance grounding head at this stage.

We evaluate models on two challenging VidSGG benchmarks: VidVRD [32] and VidOR [30]. Extensive ablations and results have demonstrated the effectiveness of our new

classification-then-grounding framework and BIG model.

In summary, we make three contributions in this paper:

1. We propose a new classification-then-grounding framework for VidSGG. It avoids three inherent drawbacks of the existing proposal-based framework.
2. We reformulate video scene graphs as temporal bipartite graphs, and take full advantage of the new framework.
3. We propose a novel model BIG, which achieves state-of-the-art performance on two VidSGG datasets.

2. Related Works

Video Scene Graph Generation. Today's VidSGG models are all proposal-based. They usually focus on designing: 1) more effective context fusing mechanism among segment or tracklet proposals, *e.g.*, GCNs or CRFs [22,28,37], compositional relation encoding [13], or structure context aggregation [36]; or 2) stronger relation association approaches, such as MHA [33] or online association [28]. In contrast, we are the first to avoid the proposal generation step and solve VidSGG task in a new classification-then-grounding manner. Meanwhile, we propose a temporal bipartite graph formulation by extending the image bipartite graph [50] into the video domain, *i.e.*, assigning predicate nodes with time slots. Accordingly, we propose a novel BIG model.

Transformer Structure for SGG. Transformer structures [38, 39] regain vision community attention after the pioneering work DETR [4], which regards the object detection task as a set prediction problem. Inspired from DETR, several recent works start to use Transformer models for image scene graph generation [12,15,35,53]. Similarly, these models utilize a set of learnable embeddings as the input of the decoder, and predict the triplets based on the encoded global object features. Inspired from these works, we also adopt the Transformer structure in our classification stage, and design a role-aware cross-attention module to explicitly model different edges of the temporal bipartite graph.

Video Grounding. It aims to localize the video segment depicted by a language query [16,49]. Existing models can be roughly grouped into: 1) Anchor-based [1,3,41,44,46,52]: They match all moment proposals to the language query and select the one with the highest matching score as the prediction. 2) Anchor-free [7,24,42,45,48]: They directly predict the probability of being a boundary for each frame, or directly regress the temporal locations of the target moment. In this paper, we convert the grounding stage as a video grounding problem, and build on top of a SOTA model DEBUG [24] by extending it into multiple segment outputs.

3. Approach

In this paper, we reformulate video scene graphs as temporal bipartite graphs. Given an entity category set $\mathcal{C}_e$ and predicate category set $\mathcal{C}_p$, a temporal bipartite graph is formally defined as $\mathcal{G} = (\mathcal{N}_e, \mathcal{N}_p, \mathcal{E})$, where $\mathcal{N}_e$, $\mathcal{N}_p$, and $\mathcal{E}$

denote the set of entity nodes, predicate nodes and edges, respectively. For each entity node $e_i \in \mathcal{N}_e$, it associates with an entity class $c_i^e \in \mathcal{C}_e$ and a time slot (s_i^e, e_i^e). Similarly, for each predicate node $p_j \in \mathcal{N}_p$, it associates with a predicate class $c_j^p \in \mathcal{C}_p$ and a set of time slots $\{(s_{j,k}^p, e_{j,k}^p)\}_{k=1}^{K_j}$. This multiple time slots setting implies that each predicate node has K_j instances with the same category (happens K_j times) in the same subject-object pair. $\mathcal{E} \subseteq \mathcal{N}_e \times \mathcal{N}_p \times \mathcal{C}_r$ is a set of mapping that maps an entity-predicate pair to a semantic role, *i.e.*, $\mathcal{N}_e \times \mathcal{N}_p \to \mathcal{C}_r$, where $\mathcal{C}_r = \{\texttt{subject}, \texttt{object}\}$ is a semantic role set. The size of $\mathcal{N}_e$ and $\mathcal{N}_p$ are denoted as n and m, respectively.

Under this new bipartite graph formulation, we propose a novel VidSGG model: **BIG**. The overview pipeline of BIG is illustrated in Figure 3, which consists of two stages: classification stage (Sec. 3.1) and grounding stage (Sec. 3.2).

3.1. Classification Stage

3.1.1 Overview

The classification stage aims to classify the categories of all the nodes (*i.e.*, entity and predicate), and the edges between them (*i.e.*, the semantic roles). As shown in Figure 3(a), the classification stage consists of four parts: a tracklet detector, an encoder, a decoder, and a classification head.

Tracklet Detector. Given a video, we use a pretrained tracklet detector to detect all tracklets in the video (denoted as entity set $\mathcal{N}_e$), and corresponding spatial-temporal locations, categories, and features. Specifically, for each entity $e_i \in \mathcal{N}_e$ with length l_i (the number of frames), it is characterized by the bounding box coordinates $\boldsymbol{b}_i \in \mathbb{R}^{l_i \times 4}$, object category $c_i^e \in \mathcal{C}_e$, and a time slot (s_i^e, e_i^e). We fix all the detection results (*i.e.*, $\{\boldsymbol{b}_i\}$ and $\{c_i^e\}$) as the final predictions.

The tracklet feature $\boldsymbol{f}_i$ for each entity e_i is a combination of appearance feature and spatial feature. The appearance feature $\boldsymbol{f}_i^a \in \mathbb{R}^{l_i \times d_a}$ is extracted at each frame based on the box locations by using RoIAlign [29]. The spatial feature $\boldsymbol{f}_i^s \in \mathbb{R}^{l_i \times 8}$ is the concatenation of all box coordinates $\boldsymbol{b}_i$ and offsets $\Delta \boldsymbol{b}_i$, where $\Delta \boldsymbol{b}_{i,j}$ is the box coordinate offsets of two consecutive frames, *i.e.*, $\Delta \boldsymbol{b}_{i,j} = \boldsymbol{b}_{i,j+1} - \boldsymbol{b}_{i,j}$. Then, the tracklet feature $\boldsymbol{f}_i \in \mathbb{R}^{l_i \times d_e}$ for entity e_i is

$$\boldsymbol{f}_i = \texttt{Conv}\left[\texttt{MLP}_a(\boldsymbol{f}_i^a); \texttt{MLP}_s(\boldsymbol{f}_i^s)\right], \tag{1}$$

where $\texttt{MLP}_a$ and $\texttt{MLP}_s$ are two learnable MLPs, $[;]$ is a concatenate operation, and $\texttt{Conv}$ is a 1D convolutional layer.

Encoder. Given entity features $\{\boldsymbol{f}_i\}$, the encoder aims to encode global context among all entities. Thus, we utilize the vanilla Transformer encoder [38] as our encoder, where each layer consists of a multi-head self-attention ($\texttt{MHSA}$) and a feed-forward network ($\texttt{FFN}$). Since the sizes of the entity features are different, we first utilize a pooling operation to transform each feature $\boldsymbol{f}_i \in \mathbb{R}^{l_i \times d_e}$ to a fixed size feature $\boldsymbol{f}_i' \in \mathbb{R}^{l \times d_e}$, and use a MLP to mapping it into a vector $\boldsymbol{h}_i \in$
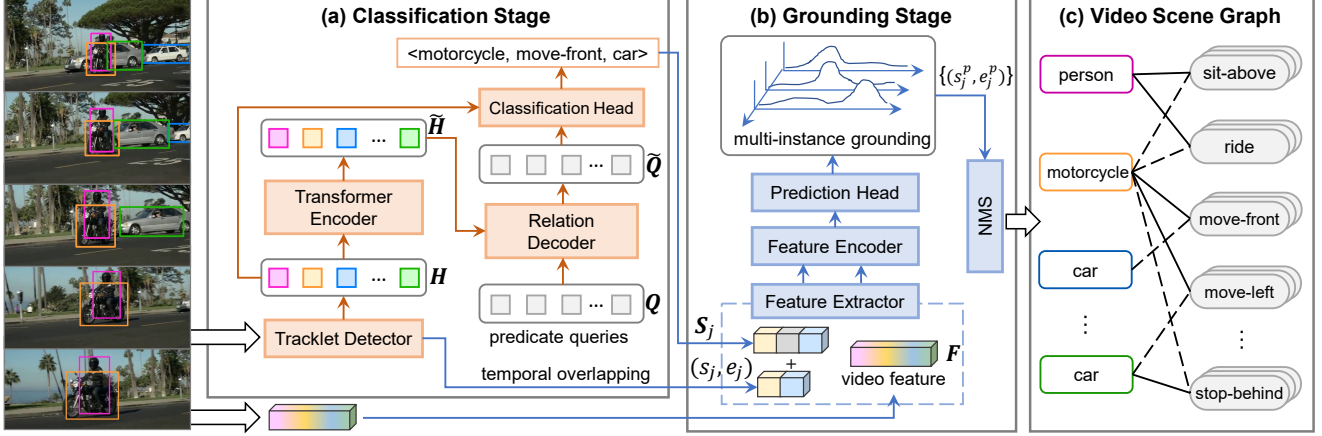
Figure 3. The overall pipeline of the proposed BIG model, which consists of a classification stage (a) and a grounding stage (b).

$\mathbb{R}^{d_e}$. Then, we stack all entity features $\{h_i\}$ into a matrix $H \in \mathbb{R}^{n \times d_e}$, and feed the H into the encoder. The outputs of the encoder are contextualized features $\widetilde{H} \in \mathbb{R}^{n \times d_e}$.

Decoder. The decoder is designed to predict the edges of the graph, and derive enhanced predicate representations for the following predicate classification. The inputs for the decoder is a fixed-size set of m predicate queries with corresponding learnable embeddings $Q \in \mathbb{R}^{m \times d_q}$. Each query is responsible for a predicate node in the bipartite graph. We built on top of the Transformer decoder and replace the original cross-attention with a *Role-aware Cross-Attention* (RaCA). Therefore, each decoder layer is summarized as:

$$
\begin{aligned}
Q'_{(i)} &= \mathrm{LNorm}(Q_{(i)} + \mathrm{MHSA}(Q_{(i)})), \\
\bar{Q}'_{(i)} &= \mathrm{RaCA}(Q'_{(i)}, \widetilde{H}, \widetilde{H}), \\
Q''_{(i)} &= \mathrm{LNorm}(Q'_{(i)} + \bar{Q}'_{(i)}), \\
Q_{(i+1)} &= \mathrm{LNorm}(Q''_{(i)} + \mathrm{FFN}(Q''_{(i)})),
\end{aligned}
\tag{2}
$$

where LNorm is the layer normalization [2], $Q_{(i)}$ is the input query embeddings of i-th decoder layer. The output of the last deocder layer is denoted as $\widetilde{Q}$, *i.e.*, the enchanced query embeddings. Meanwhile, the cross-attention matrix (inside RaCA module) of the last decoder layer is denoted as $\widetilde{A}$, which can be regarded as a soft edge linkage of the bipartite graph. More details and discussions about the RaCA module (vs. original cross-attention) are in Sec. 3.1.2.

Classification Head. Given the query embeddings $\widetilde{Q}$ and cross-attention matrix $\widetilde{A}$, the classification head aims to classify the category of each query (*i.e.*, predicate node). As shown in Figure 4, $\widetilde{A}$ has two channels which correspond to two different semantic roles in the bipartite graph. Based on $\widetilde{A}$, we first derive the predicted subject and object for each predicate node p_j by selecting the entity with the highest attention score in each channel, of which the indices are denoted as j_s and j_o, respectively. Then, the classification feature f^p_j for predicate p_j is a concatenation of three types
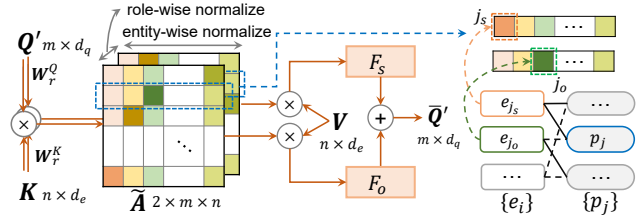

Figure 4. Illustration of the Role-aware Cross-Attention module.

of features: query embedding $\widetilde{q}_j$, subject/object entity features h_{j_s} and h_{j_o}, and word embeddings of subject/object entity categories, *i.e.*, $f^p_j = [\widetilde{q}_j; h_{j_s}; h_{j_o}; \Pi(c^e_{j_s}); \Pi(c^e_{j_o})]$, where $\Pi(c^e_i) \in \mathbb{R}^{d_w}$ is the GloVe embedding [27] of object category c^e_i. Finally, the predicate category is classified by:

$$
P(c^p_j) = \mathrm{Softmax}(\mathrm{MLP}_p(f^p_j) + b_{c^e_{j_s}, c^e_{j_o}}),
\tag{3}
$$

where MLP_p is a MLP, and $b_{*,*}$ is the statistical prior of the relation triplet categories from the training set [36, 51].

3.1.2 Role-aware Cross-Attention (RaCA)

As shown in Figure 4, the RaCA module aims to aggregate entity features from different semantic roles into query embeddings based on the cross-attention matrix. To distinguish different semantic roles (*i.e.*, subject or object), we perform cross-attention for each semantic role separately, and then fuse these role-wise features with two non-linear transformations. Specifically, let $K = V = \widetilde{H} \in \mathbb{R}^{n \times d_e}$ be the key and value matrix which are the output of the encoder, and $Q' \in \mathbb{R}^{m \times d_q}$ be the query matrix which are the output from the first subnet in each decoder layer[4] (cf. Eq. (2)). RaCA constructs a two-channel attention matrix $A \in \mathbb{R}^{2 \times m \times n}$ and each channel of A is calculated as:

$$
A_r = (Q'W^Q_r)(KW^K_r)^{\mathrm{T}}/\sqrt{d_e},
\tag{4}
$$

---
[4]For brevity, we omit subscripts i in this subsection, *e.g.*, $Q'_{(i)} \to Q'$.
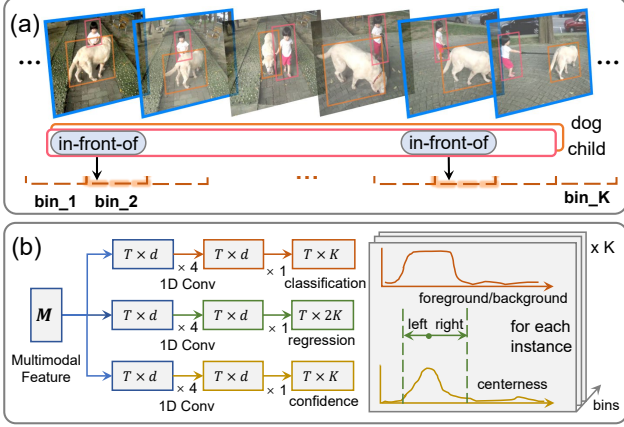
Figure 5. **(a)**: Illustration of label assignment in the multi-instance grounding. **(b)**: The overview of multi-instance grounding head.

where $\boldsymbol{W}_r^Q, \boldsymbol{W}_r^K$ are learnable weights and $r \in \{1, 2\}$ represent the subject and object channel, respectively. In our formulation, since we assume each predicate query can only link to one entity in each role and each entity-predicate pair has one type of semantic roles at most. Thus, we normalize $\boldsymbol{A}$ along both the entity axis and the role axis, i.e.,

$$\widetilde{\boldsymbol{A}}_{r,j,i} = \frac{\exp(\boldsymbol{A}_{r,j,i})}{\sum_{i'=1}^{n} \exp(\boldsymbol{A}_{r,j,i'})} \times \frac{\exp(\boldsymbol{A}_{r,j,i})}{\sum_{r'=1}^{2} \exp(\boldsymbol{A}_{r',j,i})}. \quad (5)$$

Then, we use two role-specific non-linear MLPs (F_*) to introduce role-wise distinctions into query embeddings,

$$\bar{\boldsymbol{Q}}' = F_s(\widetilde{\boldsymbol{A}}_1 \boldsymbol{V}) + F_o(\widetilde{\boldsymbol{A}}_2 \boldsymbol{V}), \ F_* : \mathbb{R}^{d_e} \mapsto \mathbb{R}^{d_q}, \quad (6)$$

where $\bar{\boldsymbol{Q}}'$ is the output of the RaCA module (cf. Eq. (2)), which aggregates role-aware information from each entity. **Discussions.** Compared to the plain cross-attention in original Transformer [38], RaCA explicitly learns the adjacency matrix of the bipartite graph based on role-wise normalization (cf. Eq. (5)) and role-aware non-linear mappings (F_* in Eq. (6)). Otherwise the adjacency matrix (or edge linkage) can not be modeled by the plain cross-attention module.

3.2. Grounding Stage

The grounding stage aims to localize the temporal location of each predicted predicate node. So far, for each predicate node p_j, the classification stage have predicted its category c_j^p and two linked entity tracklets: subject e_{j_s} and object e_{j_o}. In this stage, we regard the predicate localization as a video grounding problem [16]. Specifically, we treat the triplet categories sequence $(c_{j_s}^e, c_j^p, c_{j_o}^e)$ (e.g., ⟨person, ride, motorcycle⟩ in Figure 3(a)) as a language query, and extend an existing video grounding model DEBUG [24] for multi-instance predicate localization. As shown in Figure 3(b), this stage consists of three parts: a feature extractor, a feature encoder, and a multi-instance grounding head.

Feature Extractor. For the given video, we use a pretrained I3D [5] network to extract frame-level visual feature $\boldsymbol{F} \in \mathbb{R}^{T \times d_v}$, where T is the number of whole video frames. For query $(c_{j_s}^e, c_j^p, c_{j_o}^e)$ (refer to predicate node p_j), we initialize the query feature $\boldsymbol{S}_j = [\Pi(c_{j_s}^e), \Pi(c_j^p), \Pi(c_{j_o}^e)]$, i.e., the GloVe embeddings of the triplet categories. Meanwhile, since each predicate only happen in the overlapping time of its subject and object, we use the temporal boundaries of this overlapping time as a prior feature to enhance $\boldsymbol{S}_j$, i.e.,

$$\widetilde{\boldsymbol{S}}_j = \text{MLP}_w(\boldsymbol{S}_j) + \text{MLP}_t([s_j, e_j]), \ \widetilde{\boldsymbol{S}}_j \in \mathbb{R}^{3 \times d_w}, \quad (7)$$

where $[s_j, e_j] \in \mathbb{R}^2$ are the overlapping boundaries of the subject and object linked to predicate node p_j. Note that only those predicate nodes referring to overlapped subject-object tracklets are used in the grounding stage. Visual features $\boldsymbol{F}$ is shared for all the queries with features $\{\widetilde{\boldsymbol{S}}_j\}_{j=1}^m$.

Feature Encoder. This encoder aims to model the interaction between the video feature $\boldsymbol{F}$ and all query features $\{\widetilde{\boldsymbol{S}}_j\}$. Specifically, we use the same feature encoder as DE-BUG [24], which contains two parallel embedding encoders and a multi-modal attention layer. The output of the feature encoder is a fused multimodal feature $\boldsymbol{M} \in \mathbb{R}^{T \times d}$. We refer readers to the DEBUG [24] paper for more details.

Multi-instance Grounding Head. Different from the existing video grounding task where each query only refers to a single segment, in VidSGG, a predicate category can happen multiple times between a same subject-object pair, i.e., each language query may refer to multiple segments (cf. Figure 2). Since the number of time slots varies widely in different predicate nodes, it is difficult to directly predict a variable number of temporal segments for each query. Instead, we set K bins for each language query. As shown in Figure 5(a), in the training stage, we divide the whole normalized video length evenly into K intervals, referring to K bins. Then, each bin is assigned with the target time slots centered in its interval[5]. In the test stage, all time slots predictions are processed by NMS to reduce false positives. Finally, the NMS operation results in K_j time slots for the triplet query $(c_{j_s}^s, c_j^p, c_j^o)$, denoted as $\{(s_{j,k}^p, e_{j,k}^p)\}_{k=1}^{K_j}$.

Following DEBUG [24], we design three branches network for grounding: a classification subnet, a boundary regression subnet, and a confidence subnet (cf. Figure 5(b)). In particular, we extend the output channels of the last conv-layer to K for the classification and confidence branch, and $2K$ for the regression branch (corresponding to K bins).

3.3. Training Objectives

Classification Stage. Since we fix all the tracklets from the detection backbone as the final entity nodes predictions, we only consider the training losses for classification of the

---

edges and the predicate nodes. Let $\hat{\mathcal{N}}_p = \{\hat{p}_j\}_{j=1}^m$ be the predicted set of m predicate nodes, and $\mathcal{N}_p^*$ be the ground-truth predicate set of size m padded with $\varnothing$ (background). We adopt a one-to-one label assignment by finding a bipartite matching between $\hat{\mathcal{N}}_p$ and $\mathcal{N}_p^*$. Specifically, we search for a permutation of m elements $\hat{\sigma}$ by optimizing the cost:

$$\hat{\sigma} = \arg\min_{\sigma} \sum_{j=1}^m \mathcal{L}_{\text{match}}(p_j^*, \hat{p}_{\sigma(j)}). \quad (8)$$

This matching problem can be computed efficiently with the Hungarian algorithm [25], following prior work DETR [4].

The matching cost considers both predicate classification and edge prediction. Since all the entity nodes are fixed, the cost can be considered at the view of predicate nodes, and entity nodes are assigned to their ground-truths in advance (which is based on vIoU and the criterion is similar to that in Faster R-CNN [29]). Thus, each predicate node can be described by its category and the two edges to subject/object. We denote $p_j^* = (c_j^{p*}, \boldsymbol{a}_j^*)$, where c_j^{p*} is the predicate category (which may be $\varnothing$) and $\boldsymbol{a}_j^* \in \{0,1\}^{2 \times n}$ is the j-th row of $\boldsymbol{A}^*$ (ground-truth adjacent matrix) for two channels. Note that $a_{j,r,i}^* = 0$ when the i-th entity has no ground-truth to match. For the predicted predicate with index $\sigma(j)$, the corresponding edges are described by $\hat{\boldsymbol{a}}_{\sigma(j)} \in \mathbb{R}^{2 \times n}$, which is the $\sigma(j)$-th row of the predicted $\hat{\boldsymbol{A}}$ for two channels. With the above notations, the matching cost is defined as:

$$\begin{aligned}\mathcal{L}_{\text{match}}(p_j^*, \hat{p}_{\sigma(j)}) = &- \mathbf{1}_{\{c_j^{p*} \neq \varnothing\}} \log P(\hat{c}_{\sigma(j)}^p = c_j^{p*}) \\ &+ \mathbf{1}_{\{c_j^{p*} \neq \varnothing\}} \lambda_{\text{att}} \mathcal{L}_{\text{att}}(\boldsymbol{a}_j^*, \hat{\boldsymbol{a}}_{\sigma(j)}),\end{aligned} \quad (9)$$

where λ_{att} is hyperparameter, and $\mathcal{L}_{\text{att}}$ is defined as a binary-cross entropy (BCE) loss, $\mathbf{1}_{\{\cdot\}}$ is an indicator function. After obtaining $\hat{\sigma}$, the loss $\mathcal{L}_c$ for classification stage consists of the matching loss between $(p_j^*, \hat{p}_{\hat{\sigma}(j)})$ pairs, and the background classification loss for other predicate nodes, i.e.,

$$\mathcal{L}_c = \sum_j \mathcal{L}_{\text{match}}(p_j^*, \hat{p}_{\hat{\sigma}(j)}) - \sum_{c_j^{p*} = \varnothing} \log P(\hat{c}_{\hat{\sigma}(j)}^p = \varnothing). \quad (10)$$

Grounding Stage. The grounding stage is trained separately from the classification stage, and we use ground-truth triplet categories for training. Following DEBUG [24], the training objectives consist of three losses for three respective branches. The total loss is averaged among all K bins.

4. Experiments

4.1. Datasets and Evaluation Metrics

Datasets. We evaluated BIG on two benchmarks: 1) **VidVRD** [32]: It consists of 1,000 videos, which covers 35 object categories and 132 predicate categories. We used the official splits: 800 videos for training and 200 videos for test. 2) **VidOR** [30]: It consists of 10,000 videos, which covers 80 object categories and 50 predicate categories. We used official splits: 7,000 videos for training, 835 videos for

validation, and 2,165 videos for test. Since the annotations of the test set are not released, we only evaluated the val set.

Evaluation Metrics. We evaluated BIG on two tasks: 1) *Relation Detection* (**RelDet**): It detects a set of visual relation triplets, and corresponding tracklets of subject and object. A detected triplet is considered to be correct if there is a same triplet tagged in ground-truth, and both subject and object tracklets have a sufficient vIoU (e.g., 0.5) with the ground-truth. We used mAP and Recall@K (R@K, K=50, 100) as metrics for RelDet. 2) *Relation Tagging* (**RelTag**): It only focuses on the precision of visual relation triplets and ignores the localization results of tracklets. For RelTag, we used Precision@K (P@K, K=1,5,10) as metrics.

4.2. Implementation Details

Tracklet Detector. We utilized the video object detector MEGA [14, 17] with backbone ResNet-101 [18] to obtain initial frame-level detection results, and adopted deepSORT [40] to generate object tracklets.

Adapting BIG to VidVRD. For each relation triplet in the training set of VidVRD [32], we noticed that only a portion of ground-truth segments is annotated as foreground, which makes the annotated temporal boundaries unreliable for training. Therefore, we only used the classification stage of BIG for VidVRD, termed **BIG-C**. Consequently, the time slot for each predicate p_j is calculated as the overlap of its subject and object, i.e., $(s_{j_s}^e, e_{j_s}^e) \cap (s_{j_o}^e, e_{j_o}^e)$, and $K_j = 1$.

More implementation details are left in the appendix.

4.3. Ablation Studies

Effectiveness of Classification-Then-Grounding. We designed a baseline model to show the effectiveness of this framework and the two stages (classification & grounding). Specifically, it directly classifies predicate categories of all tracklet pairs through multi-label classification, based on feature $\boldsymbol{f}_j'^p = [\boldsymbol{h}_{j_s}; \boldsymbol{h}_{j_o}; \Pi(c_{j_s}^e); \Pi(c_{j_o}^e)]$ (cf. $\boldsymbol{f}_j^p$ in Eq.(3)), namely **Base-C**. Then, we apply the grounding stage to the Base-C, termed **Base**. All results are in Table 2. From this table, we can observe that even without the BIG model, the simple classification-then-grounding baseline (Base) still outperforms SOTA proposal-based model Sun et al. [34].

Furthermore, we reported the number of average relation candidates for the grounding stage (#Cand.) in Table 2 to demonstrate the effectiveness of each stage. For the classification stage, by comparing BIG-C with Base-C, we can observe that BIG-C outperforms Base-C on all metrics, especially with a large margin on RelTag while having fewer average relation candidates (135.4 vs. 482.1), which demonstrates the superiority of the encoder-decoder pipeline under the temporal bipartite graph formulation. For the grounding stage, we can observe that it can consistently improve detection mAP and recall for both two backbones (Base-C and BIG-C). The improvements of RelTag are slight because it

Models	Features		RelDet			RelTag		
	Visual	Motion	mAP	R@50	R@100	P@1	P@5	P@10
VidVRD [32]$_{MM'17}$	iDT	✓	8.58	5.54	6.37	43.00	28.90	20.80
GSTEG [37]$_{CVPR'19}$	iDT	✓	9.52	7.05	8.67	51.50	39.50	28.23
VRD-GCN [28]$_{MM'19}$	iDT	✓	16.26	8.07	9.33	57.50	41.00	28.5 0
MHA [33]$_{MM'20}$	iDT	✓	19.03	9.53	10.38	57.50	41.40	29.45
IVRD [21]$_{MM'21}$	RoI	✓	22.97	12.40	14.46	68.83	49.87	35.57
VidVRD-II [31]$_{MM'21}$	RoI	✓	29.37	19.63	22.92	70.40	53.88	40.16
Liu *et al*. [22]$_{CVPR'20}$	RoI+I3D†	✓	18.38	11.21	13.69	60.00	43.10	32.24
Chen *et al*. [13]$_{ICCV'21}$	RoI+I3D	✓	20.08	13.73	16.88	62.50	49.20	38.45
Liu *et al*. [22]$_{CVPR'20}$	RoI†		14.01	8.47	**11.00**	56.50	36.70	26.60
TRACE [36]$_{ICCV'21}$	RoI		15.06	7.67	10.32	—	—	—
BIG-C (Ours)	RoI†		**17.56**	**9.59**	10.92	**56.50**	**44.30**	**32.35**
Liu *et al*. [22]$_{CVPR'20}$	RoI+I3D†		14.81	9.14	**11.39**	55.50	38.90	28.90
TRACE [36]$_{ICCV'21}$	RoI+I3D		17.57	9.08	11.15	**61.00**	**45.30**	**33.50**
BIG-C (Ours)	RoI+I3D†		**17.67**	**9.63**	11.29	56.00	43.80	32.85
BIG-C (Ours)	RoI‡		**26.08**	**14.10**	**16.25**	**73.00**	**55.10**	**40.00**

Table 1. Performance (%) on VidVRD of SOTA methods. **Visual**: † means that these models use the same tracklets and features as Liu *et al*. [22], and ‡ means that these models use tracklets and features generated by MEGA. **Motion**: It refers to the relative motion feature of entity pairs [31].
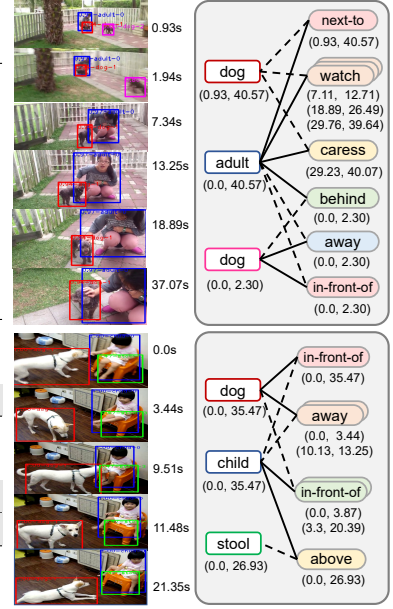

Figure 6. Qualitative results on VidOR.

Models	RelDet (%)			RelTag (%)		#Cand.
	mAP	R@50	R@100	P@1	P@5	
Sun *et al*. [34]	6.56	6.89	8.83	51.20	40.73	—
Base-C	7.05	7.17	9.19	59.01	47.07	482.1
Base	7.19	7.32	9.50	59.49	47.28	482.1
BIG-C	8.29	7.92	9.65	64.42	51.70	135.4
BIG	**8.54**	**8.03**	**10.04**	64.42	**51.80**	135.4

Table 2. Ablations on effectiveness of different stages on VidOR.

doesn't consider the locations of relation triplets.

Number of Predicate Queries. We compared BIG-C with different number of predicate queries (m) in Table 4. It can be observed that more predicate queries always improve the final VidSGG performance, but also result in more computations (*e.g*., #Cand.). To trade-off effectiveness and efficiency, we set $m = 192$ for all the following experiments.

Ablations on the RaCA Module. We analyzed the impact of role-wise normalization (R-norm) and the two role-specific mappings (F_*) in the RaCA module on BIG-C. From the results in Table 5, we can observe that both the R-norm and F_* are important for the role-aware information encoding. Particularly, when both two techniques are used, the model achieves the best results, especially on P@1.

Ablations on the Multi-instance Grounding. We further investigated the influence of different number of bins in the multi-instance grounding. Since each predicate category of a same subject-object may have multiple instances, we regarded relation triplets with the same subject-object pair and predicate category as a sample. Each ground-truth sample can be partially hit with a **fraction recall** (**fR**), which is

calculated as the fraction of hit relation triplets of each sample. For more precise, we evaluated fR@K for ground-truth samples with a single instance (fR$_S$) and multiple instances (fR$_M$), separately. From the results in Table 6, we can observe that: 1) With the increase of K, corresponding fR@K increases on both single-instance and multi-instance samples. 2) Our multi-instance grounding (*e.g*., #Bins=5,10) is more capable of improving the fR of predicates on multi-instance samples, *e.g*., the relative gains of fR$_M$ are larger than fR$_S$ (3.97% (5.53→5.75) vs. 0.61% (12.96→13.04)).

4.4. Comparisons with State-of-the-Arts

4.4.1 Performance on VidVRD

Settings. For VidVRD, we compared our BIG-C with several state-of-the-art methods, which can be coarsely categorized into two groups: 1) Segment-proposal based methods: **VidVRD** [32], **GSTEG** [37], **VRD-GCN** [28], **MHA** [33], **IVRD** [21], **VidVRD-II** [31], and **TRACE** [36]. 2) Tracklet-proposal based models methods: **Liu *et al*.** [22] and **Chen *et al*.** [13]. For more fair comparisons, we also reported the results of BIG-C with the same features as [22]. **Results.** All results are reported in Table 1. From this table, we have following observations: 1) When using the MEGA backbone, BIG-C (with only RoI feature) beats most of the proposal-based methods even without the grounding stage. Particularly, we achieve a very high mAP (*i.e*., 26.08%) and the highest P@1 (*i.e*., 73.00%). 2) When using the same RoI feature as [22], BIG-C outperforms TRACE [36] and Liu *et al*. [22] on both RelDet and RelTag tasks, especially we achieve significant performance gains on mAP (17.56%

Models	Detector	Features				RelDet			RelTag		
		Visual	Lang	Motion	Mask	mAP	R@50	R@100	P@1	P@5	P@10
Liu *et al.* [22]$_{CVPR'20}$	RefineDet	RoI+I3D$_r$		✓		6.85	8.21	9.90	51.20	40.73	—
Chen *et al.* [13]$_{ICCV'21}$	Faster R-CNN	RoI+I3D$_r$	✓	✓		10.04	8.94	10.69	61.52	50.05	38.48
Chen *et al.* [13]$_{ICCV'21}$	Faster R-CNN	RoI+I3D$_r$	✓	✓	✓	11.21	9.99	11.94	68.86	55.16	43.40
IVRD [21]$_{MM'21}$	Faster R-CNN	RoI		✓		7.42	7.36	9.41	53.40	42.70	—
Chen *et al.* [13]$_{ICCV'21}$	Faster R-CNN	RoI		✓		**8.93**	7.38	9.22	56.89	44.76	34.07
VidVRD-II [31]$_{MM'21}$	Faster R-CNN	RoI		✓		8.65	**8.59**	**10.69**	57.40	44.54	33.30
BIG-C (Ours)	MEGA	RoI				8.03	7.60	9.39	**62.25**	50.96	40.30
BIG (Ours)	MEGA	RoI+I3D$_f$				8.28	7.74	9.82	62.13	**51.25**	40.48
VRU'19-top1 [34]$_{MM'19}$	FGFA	—	✓	✓		6.56	6.89	8.83	51.20	40.73	—
MHA [33]$_{MM'20}$	FGFA	—	✓	✓		6.59	6.35	8.05	50.72	41.56	—
VRU'20-top1 [43]$_{MM'20}$	CascadeRCNN	RoI	✓	✓	✓	**9.93**	**9.12**	—	**67.43**	—	—
Chen *et al.* [13]$_{ICCV'21}$	Faster R-CNN	RoI	✓	✓		9.54	8.49	**10.17**	59.24	47.24	35.99
BIG-C (Ours)	MEGA	RoI	✓			8.29	7.92	9.65	64.42	51.70	**41.05**
BIG (Ours)	MEGA	RoI+I3D$_f$	✓			8.54	8.03	10.04	64.42	51.80	40.96

Table 3. Performance (%) on VidOR of SOTA models. The **Best** and second best are marked in according formats. **Visual**: I3D$_r$ and I3D$_f$ denote region-level and frame-level I3D features, respectively. **Lang**: The word embeddings of entity categories. **Motion**: It refers to the relative motion feature of entity pairs [31]. **Mask**: It means the localization mask of entities [43].

m	RelDet (%)		RelTag (%)		#Cand.
	mAP	R@50	P@1	P@5	
128	7.50	7.15	62.62	51.02	105.1
192	8.29	7.92	64.42	51.70	135.4
256	8.31	7.92	62.86	51.22	169.3

Table 4. Ablations of BIG-C for different number of predicate queries on VidOR.

R-norm	F_*	RelDet (%)		RelTag (%)	
		mAP	R@50	P@1	P@5
	✓	7.98	7.71	61.65	51.10
✓		8.02	7.36	61.65	51.68
✓	✓	8.29	7.92	64.42	51.70

Table 5. Ablations of BIG-C for the R-norm and F_* of RaCA module on VidOR.

#Bins	fR$_S$@K (%)			fR$_M$@K (%)		
	50	100	150	50	100	150
1	12.96	15.59	16.76	5.53	6.86	7.46
5	13.07	15.83	17.26	5.75	7.20	8.05
10	13.04	15.89	17.61	5.75	7.30	8.25

Table 6. Ablations for multi-instance grounding with different number of bins on VidOR.

vs. 14.01%). 3) When using the same RoI and I3D features as [22], BIG-C achieves better results on mAP and R@50.

4.4.2 Performance on VidOR

Settings. For VidOR, we compared our BIG (and BIG-C) with the state-of-the-art methods: **MHA** [33], **IVRD** [21], **VidVRD-II** [31], **Liu *et al*.** [22], **Chen *et al*.** [13], and two top-1 methods [34, 43] from Video Relation Understanding (VRU) Challenges. All results are reported in Table 3. It is worth noting that we only use the frame-level I3D features (I3D$_f$) in BIG model (*i.e.*, the grounding stage), while some works use more stronger region-level I3D features (I3D$_r$).

Quantitative Results. Due to the multifarious object detector backbones and features, it is difficult to fairly compare BIG (BIG-C) with these methods. From Table 3, we can observe: 1) For BIG (BIG-C) without language feature, we achieve significant performance gains on RelTag (*e.g.*, 51.25% vs. 44.76% on P@5), and also have competitive performance on RelDet. 2) For BIG (BIG-C) with language feature, we can also achieve comparable results on RelDet and RelTag, especially the highest P@5 (*i.e.*, 51.80%).

Qualitative Results. Figure 6 shows some qualitative results that justifies the necessity of the grounding stage. Take ⟨dog, away, child⟩ for example: Without grounding, we can only use the temporal intersection of dog and child to approximate the time slot of away, *i.e.*, (0, 35.47). Instead, with the help of multi-instance grounding, the time slots are predicted as (0, 3.44) & (10.13, 13.25). Refer to the appendix for more details.

5. Conclusions and Limitations

In this paper, we pointed out three inherent drawbacks of the prevalent proposal-based framework, and proposed a new classification-then-grounding framework for VidSGG. Under this framework, we reformulated video scene graphs as temporal bipartite graphs, and proposed a novel VidSGG model BIG. We validated the effectiveness of BIG through extensive comparative and ablative experiments.

Limitations. 1) Detecting long object tracklets in videos is still an open problem, and the fragmented tracklets may weaken the advantages of our framework, making it close to the proposal-based one. 2) Multi-instance grounding may not be suitable for some extreme situations where too many targets fall into the same bin (videos with dense relations).

References

[1] Lisa Anne Hendricks, Oliver Wang, Eli Shechtman, Josef Sivic, Trevor Darrell, and Bryan Russell. Localizing moments in video with natural language. In *ICCV*, pages 5803–5812, 2017. 3

[2] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv*, 2016. 4

[3] Meng Cao, Long Chen, Mike Zheng Shou, Can Zhang, and Yuexian Zou. On pursuit of designing multi-modal transformer for video grounding. In *EMNLP*, 2021. 3

[4] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, pages 213–229, 2020. 3, 6

[5] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017. 5

[6] Long Chen, Zhihong Jiang, Jun Xiao, and Wei Liu. Human-like controllable image captioning with verb-specific semantic roles. In *CVPR*, pages 16846–16856, 2021. 1

[7] Long Chen, Chujie Lu, Siliang Tang, Jun Xiao, Dong Zhang, Chilie Tan, and Xiaolin Li. Rethinking the bottom-up framework for query-based video localization. In *AAAI*, pages 10551–10558, 2020. 3

[8] Long Chen, Wenbo Ma, Jun Xiao, Hanwang Zhang, and Shih-Fu Chang. Ref-nms: breaking proposal bottlenecks in two-stage referring expression grounding. In *AAAI*, pages 1036–1044, 2021. 1

[9] Long Chen, Xin Yan, Jun Xiao, Hanwang Zhang, Shiliang Pu, and Yueting Zhuang. Counterfactual samples synthesizing for robust visual question answering. In *CVPR*, pages 10800–10809, 2020. 1

[10] Long Chen, Hanwang Zhang, Jun Xiao, Liqiang Nie, Jian Shao, Wei Liu, and Tat-Seng Chua. Sca-cnn: Spatial and channel-wise attention in convolutional networks for image captioning. In *CVPR*, pages 5659–5667, 2017. 1

[11] Long Chen, Yuhang Zheng, Yulei Niu, Hanwang Zhang, and Jun Xiao. Counterfactual samples synthesizing and training for robust visual question answering. *arXiv*, 2021. 1

[12] Mingfei Chen, Yue Liao, Si Liu, Zhiyuan Chen, Fei Wang, and Chen Qian. Reformulating hoi detection as adaptive set prediction. In *CVPR*, 2021. 3

[13] Shuo Chen, Zenglin Shi, Pascal Mettes, and Cees G. M. Snoek. Social fabric: Tubelet compositions for video relation detection. In *ICCV*, 2021. 2, 3, 7, 8

[14] Yihong Chen, Yue Cao, Han Hu, and Liwei Wang. Memory enhanced global-local aggregation for video object detection. In *CVPR*, pages 10337–10346, 2020. 6

[15] Yuren Cong, Wentong Liao, Hanno Ackermann, Bodo Rosenhahn, and Michael Ying Yang. Spatial-temporal transformer for dynamic scene graph generation. In *ICCV*, pages 16372–16382, 2021. 3

[16] Jiyang Gao, Chen Sun, Zhenheng Yang, and Ram Nevatia. Tall: Temporal activity localization via language query. In *ICCV*, pages 5267–5275, 2017. 3, 5

[17] Kaifeng Gao, Long Chen, Yifeng Huang, and Jun Xiao. Video relation detection via tracklet based visual transformer. In *ACM MM*, pages 4833–4837, 2021. 6

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

[19] Drew A Hudson and Christopher D Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *CVPR*, pages 6700–6709, 2019. 1

[20] Justin Johnson, Ranjay Krishna, Michael Stark, Li-Jia Li, David Shamma, Michael Bernstein, and Li Fei-Fei. Image retrieval using scene graphs. In *CVPR*, pages 3668–3678, 2015. 1

[21] Yicong Li, Xun Yang, Xindi Shang, and Tat-Seng Chua. Interventional video relation detection. In *ACM MM*, pages 4091–4099, 2021. 7, 8

[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 10840–10849, 2020. 2, 3, 7, 8

[23] Daqing Liu, Hanwang Zhang, Zheng-Jun Zha, Meng Wang, and Qianru Sun. Joint visual grounding with language scene graphs. *arXiv*, 2019. 1

[24] Chujie Lu, Long Chen, Chilie Tan, Xiaolin Li, and Jun Xiao. Debug: A dense bottom-up grounding approach for natural language video localization. In *EMNLP*, pages 5144–5153, 2019. 3, 5, 6

[25] James Munkres. Algorithms for the assignment and transportation problems. *Journal of the society for industrial and applied mathematics*, pages 32–38, 1957. 6

[26] Yulei Niu and Hanwang Zhang. Introspective distillation for robust question answering. In *NeurIPS*, 2021. 1

[27] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *EMNLP*, pages 1532–1543, 2014. 4

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

[29] Shaoqing Ren, Kaiming He, Ross B Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NeurIPS*, 2015. 3, 6

[30] Xindi Shang, Donglin Di, Junbin Xiao, Yu Cao, Xun Yang, and Tat-Seng Chua. Annotating objects and relations in user-generated videos. In *ICMR*, pages 279–287, 2019. 2, 6

[31] Xindi Shang, Yicong Li, Junbin Xiao, Wei Ji, and Tat-Seng Chua. Video visual relation detection via iterative inference. In *ACM MM*, pages 3654–3663, 2021. 7, 8

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

[33] Zixuan Su, Xindi Shang, Jingjing Chen, Yu-Gang Jiang, Zhiyong Qiu, and Tat-Seng Chua. Video relation detection via multiple hypothesis association. In *ACM MM*, pages 3127–3135, 2020. 1, 3, 7, 8

[34] Xu Sun, Tongwei Ren, Yuan Zi, and Gangshan Wu. Video visual relation detection via multi-modal feature fusion. In *ACM MM*, pages 2657–2661, 2019. 6, 7, 8

[35] Masato Tamura, Hiroki Ohashi, and Tomoaki Yoshinaga. Qpic: Query-based pairwise human-object interaction detection with image-wide contextual information. In *CVPR*, 2021. 3

[36] Yao Teng, Limin Wang, Zhifeng Li, and Gangshan Wu. Target adaptive context aggregation for video scene graph generation. In *ICCV*, pages 13688–13697, 2021. 3, 4, 7

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

[38] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 3, 5

[39] Wenxiao Wang, Lu Yao, Long Chen, Binbin Lin, Deng Cai, Xiaofei He, and Wei Liu. Crossformer: A versatile vision transformer hinging on cross-scale attention. In *ICLR*, 2022. 3

[40] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. Simple online and realtime tracking with a deep association metric. In *ICIP*, pages 3645–3649, 2017. 6

[41] Shaoning Xiao, Long Chen, Jian Shao, Yueting Zhuang, and Jun Xiao. Natural language video localization with learnable moment proposals. In *EMNLP*, 2021. 3

[42] Shaoning Xiao, Long Chen, Songyang Zhang, Wei Ji, Jian Shao, Lu Ye, and Jun Xiao. Boundary proposal network for two-stage natural language video localization. In *AAAI*, pages 2986–2994, 2021. 3

[43] Wentao Xie, Guanghui Ren, and Si Liu. Video relation detection with trajectory-aware multi-modal features. In *ACM MM*, pages 4590–4594, 2020. 8

[44] Huijuan Xu, Kun He, Bryan A Plummer, Leonid Sigal, Stan Sclaroff, and Kate Saenko. Multilevel language and vision integration for text-to-clip retrieval. In *AAAI*, pages 9062–9069, 2019. 3

[45] Xun Yang, Jianfeng Dong, Yixin Cao, Xun Wang, Meng Wang, and Tat-Seng Chua. Tree-augmented cross-modal encoding for complex-query video retrieval. In *ACM MM*, pages 1339–1348, 2020. 3

[46] Xun Yang, Fuli Feng, Wei Ji, Meng Wang, and Tat-Seng Chua. Deconfounded video moment retrieval with causal intervention. In *SIGIR*, pages 1–10, 2021. 3

[47] Xu Yang, Kaihua Tang, Hanwang Zhang, and Jianfei Cai. Auto-encoding scene graphs for image captioning. In *CVPR*, pages 10685–10694, 2019. 1

[48] Xun Yang, Shanshan Wang, Jian Dong, Jianfeng Dong, Meng Wang, and Tat-Seng Chua. Video moment retrieval with cross-modal neural architecture search. *TIP*, 2022. 3

[49] Yitian Yuan, Xiaohan Lan, Xin Wang, Long Chen, Zhi Wang, and Wenwu Zhu. A closer look at temporal sentence grounding in videos: Dataset and metric. In *ACM MM workshop*, pages 13–21, 2021. 3

[50] Alireza Zareian, Svebor Karaman, and Shih-Fu Chang. Weakly supervised visual semantic parsing. In *CVPR*, pages 3736–3745, 2020. 2, 3

[51] Rowan Zellers, Mark Yatskar, Sam Thomson, and Yejin Choi. Neural motifs: Scene graph parsing with global context. In *CVPR*, pages 5831–5840, 2018. 4

[52] Da Zhang, Xiyang Dai, Xin Wang, Yuan-Fang Wang, and Larry S Davis. Man: Moment alignment network for natural language moment retrieval via iterative graph adjustment. In *CVPR*, pages 1247–1257, 2019. 3

[53] Cheng Zou, Bohan Wang, Yue Hu, Junqi Liu, Qian Wu, Yu Zhao, Boxun Li, Chenguang Zhang, Chi Zhang, Yichen Wei, et al. End-to-end human object interaction detection with hoi transformer. In *CVPR*, 2021. 3