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 Bipartite Graph 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 interpretability, scene graphs have been widely used in numerous downstream tasks to help boost model performance, e.g., captioning [6, 10, 50], grounding [8, 25], and QA [9, 11, 19, 28].

Video Scene Graph Generation (VidSGG) has achieved significant progress over the recent years. Currently, almost all existing VidSGG models are proposal-based†. 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., \langle subject, predicate, object \rangle) among adjacent segments [30, 35, 36]. 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.

‡We use proposals to represent paired subject-object tracklet segments.
the whole video and generate tracklet proposals by sliding-windows [24] 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 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 [53], but also avoid exhaustively enumerating all entity pairs for predicate prediction.

Accordingly, we propose a Bi-partite Graph 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 [35] and VidOR [33]. 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 [24,30,40], compositional relation encoding [13], or structure context aggregation [39]; or 2) stronger relation association approaches, such as MHA [36] or online association [30]. 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 [53] 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 [41,42] 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 [12,15,38,56]. 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 [16,52]. Existing models can be roughly grouped into: 1) Anchor-based [1,3,44,47,49,55]: They match all moment proposals to the language query and select the one with the highest matching score as the prediction. 2) Anchor-free [7,26,45,48,51]: 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 [26] 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 \(C_e\) and predicate category set \(C_p\), a temporal bipartite graph is formally defined as \(G = (N_e, N_p, E)\), where \(N_e\), \(N_p\), and \(E\) denote the set of entity nodes, predicate nodes and edges, respectively. For each entity node \(e_i \in N_e\), it associates with an entity class \(c^e_i \in C_e\) and a time slot \((s^e_i, e^e_i)\). Similarly, for each predicate node \(p_j \in N_p\), it associates with a predicate class \(c^p_j \in C_p\) and a set of time slots \(\{(s^p_{j,k}, e^p_{j,k})\}_{k=1}^{K_j}\). 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. \(E \subseteq N_e \times N_p \times C_r\) is a set of mapping that maps an entity-predicate pair to a semantic role, i.e., \(N_e \times N_p \rightarrow C_r\), where \(C_r = \{\text{subject, object}\}\) is a semantic role set. The size of \(N_e\) and \(N_p\) 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 \(N_e\)), and corresponding spatial-temporal locations, categories, and features. Specifically, for each entity \(e_i \in N_e\) with length \(l_i\) (the number of frames), it is characterized by the bounding box coordinates \(b_i \in \mathbb{R}^{l_i \times 4}\), object category \(c^e_i \in C_e\), and a time slot \((s^e_i, e^e_i)\). We fix all the detection results (i.e., \(\{b_i\}\) and \(\{c^e_i\}\)) as the final predictions.

The tracklet feature \(f_i\) for each entity \(e_i\) is a combination of appearance feature and spatial feature. The appearance feature \(f^a_i \in \mathbb{R}^{l_i \times d_a}\) is extracted at each frame based on the box locations by using RoIAlign [31]. The spatial feature \(f^p_i \in \mathbb{R}^{l_i \times 8}\) is the concatenation of all box coordinates \(b_i\), and offsets \(\Delta b_{i,j}\), where \(\Delta b_{i,j}\) is the box coordinate offsets of two consecutive frames, i.e., \(\Delta b_{i,j} = b_{i,j+1} - b_{i,j}\). Then, the tracklet feature \(f_i \in \mathbb{R}^{l_i \times d_a}\) for entity \(e_i\) is

\[
f_i = \text{Conv} \left[ \text{MLP}_a(f^a_i); \text{MLP}_s(f^p_i) \right],
\]

where \(\text{MLP}_a\) and \(\text{MLP}_s\) are two learnable MLPs, \([\cdot]\) is a concatenation operation, and \(\text{Conv}\) is a 1D convolutional layer.

Encoder. Given entity features \(\{f_i\}\), the encoder aims to encode global context among all entities. Thus, we utilize the vanilla Transformer encoder [41] as our encoder, where each layer consists of a multi-head self-attention (MHSA) and a feed-forward network (FFN). Since the sizes of the entity features are different, we first utilize a pooling operation to transform each feature \(f_i \in \mathbb{R}^{l_i \times d_a}\) to a fixed size feature \(f^p_i \in \mathbb{R}^{l \times d_p}\), and use an MLP to mapping it into a vector \(h_i \in \mathbb{R}^{d_p}\)
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 \( \tilde{H} \in \mathbb{R}^{n \times d_e} \).

**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).

The RaCA module aims to aggregate relation triplet categories from the training set \([39, 54]\). As shown in Figure 4, the RaCA module builds a two-channel attention matrix \( \bar{R} \), 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 \( \bar{Q} \) and cross-attention matrix \( \bar{A} \), the classification head aims to classify the category of each query \( i.e., \) predicate node. As shown in Figure 4, \( \bar{A} \) has two channels which correspond to the output of the last decoder layer is denoted as \( \bar{A} \), 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.

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 \( K = V = \bar{H} \in \mathbb{R}^{n \times d_e} \) 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 \( A_r \in \mathbb{R}^{2 \times m \times n} \) and each channel of \( A \) is calculated as:

\[
A_r = (Q' W_r Q)(K W_r K)^T / \sqrt{d_v},
\]

**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 \( K = V = \bar{H} \in \mathbb{R}^{n \times d_e} \) 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 \( A_r \in \mathbb{R}^{2 \times m \times n} \) and each channel of \( A \) is calculated as:

\[
A_r = (Q' W_r Q)(K W_r K)^T / \sqrt{d_v},
\]

\(^4\)For brevity, we omit subscripts \( i \) in this subsection, \( e.g., Q'(i) \rightarrow Q' \).
where $W_r^Q$, $W_r^K$ 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 $A$ along both the entity axis and the role axis, i.e.,

$$
\bar{A}_{r,j,i} = \frac{\exp(A_{r,j,i})}{\sum_{v=1}^{n} \exp(A_{r,j,v})} \times \frac{\exp(A_{r,j,i})}{\sum_{r'=1}^{2} \exp(A_{r',j,i})}.
$$

(5)

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

$$
Q' = F_s(\bar{A}_1 V) + F_o(\bar{A}_2 V), \quad F_r : \mathbb{R}^{d_e} \rightarrow \mathbb{R}^{d_q},
$$

(6)

where $Q'$ 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 [41], RaCA explicitly learns the adjacency matrix of the bipartite graph based on role-wise normalization (cf. Eq. (5)) and role-aware non-linear mappings ($F_r$ in Eq. (6)). Otherwise the adjacency matrix (or edge linkage) cannot 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 has predicted its category $c_j$ 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 $(c_j, e_{j,s}, e_{j,o})$ (e.g., (person, ride, motorcycle)) in Figure 3(a) as a language query, and extend an existing video grounding model DEBUG [26] 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 pre-trained 13D [5] network to extract frame-level visual feature $F \in \mathbb{R}^{T \times d_f}$, where $T$ is the number of whole video frames. For query $(c_j, e_{j,s}, e_{j,o})$ (refer to predicate node $p_j$), we initialize the query feature $S_j = [\Pi(c_j), \Pi(e_{j,s}), \Pi(e_{j,o})]$, 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 $S_j$, i.e.,

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

(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 $F$ is shared for all the queries with features $\{S_j\}_{j=1}^{m}$.

**Feature Encoder.** This encoder aims to model the interaction between the video feature $F$ and all query features $\{S_j\}_j$. Specifically, we use the same feature encoder as DEBUG [26], which contains two parallel embedding encoders and a multi-modal attention layer. The output of the feature encoder is a fused multimodal feature $M \in \mathbb{R}^{T \times d_w}$. We refer readers to the DEBUG [26] 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 $(c_j, e_{j,s}, e_{j,o})$, denoted as $(\{(s_{j,k}, e_{j,k})\}_{k=1}^{K_j})$.

Following DEBUG [26], 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 convolution 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

$^5$Although multiple targets may still fall into a same bin, such samples only account for a small proportion (details are in the appendix).
edges and the predicate nodes. Let \( \hat{N}_p = \{ \hat{p}_j \}_{j=1}^{m} \) be the predicted set of \( m \) predicate nodes, and \( N^*_p \) be the ground-truth predicate set of size \( m \) padded with \( \emptyset \) (background). We adopt a one-to-one label assignment by finding a bipartite matching between \( \hat{N}_p \) and \( N^*_p \). Specifically, we search for a permutation of \( m \) elements \( \hat{\sigma} \) by optimizing the cost:

\[
\hat{\sigma} = \arg \min_\sigma \sum_{j=1}^{m} L_{\text{match}}(\hat{p}_j^*, \hat{p}_{\sigma(j)}).
\]

This matching problem can be computed efficiently with the Hungarian algorithm [27], 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 [31]). Thus, each predicate node can be described by its category and the two edges to subject/object.

We denote \( p^*_j = (c^*_j, a^*_j) \), where \( c^*_j \) is the predicate category (which may be \( \emptyset \) and \( a^*_j \in \{0, 1\}^{2 \times n} \) is the \( j \)-th row of \( A^* \) (ground-truth adjacent matrix) for two channels. Note that \( a^*_j = 0 \) 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 \( \hat{a}_{\sigma(j)} \in \mathbb{R}^{2 \times n} \), which is the \( \sigma(j) \)-th row of the predicted \( \hat{A} \) for two channels. With the above notations, the matching cost is defined as:

\[
L_{\text{match}}(\hat{p}_j^*, \hat{p}_{\sigma(j)}) = -1(c^*_j \neq \emptyset) \log P(\hat{c}^P_{\sigma(j)} = c^*_j) + 1(c^*_j \neq \emptyset) \lambda_{\text{att}} L_{\text{att}}(a^*_j, \hat{a}_{\sigma(j)}),
\]

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

\[
L_c = \sum_{j} L_{\text{match}}(p^*_j, \hat{p}_{\sigma(j)}) - \sum_{j=1}^{m} c_j \log P(\hat{c}^P_{\sigma(j)} = \emptyset).
\]

**Grounding Stage.** The grounding stage is trained separately from the classification stage, and we use ground-truth triplet categories for training. Following DEBUG [26], the training objectives consist of three losses for three respective branches. The total loss is averaged among all \( K \) bins.

**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 [43] to generate object tracklets.

**Adapting BIG to VidVRD.** For each relation triplet in the training set of VidVRD [35], 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., \( (s^e_j, c^e_j) \cap (s^e_j, c^e_j) \), 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 \( f^R \) (cf. \( f^R \) 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. [37].

**4.4. Experiments**

**4.1. Datasets and Evaluation Metrics**

**Datasets.** We evaluated BIG on two benchmarks: 1) VidVRD [35]: 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 [33]: 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.
doesn’t consider the locations of relation triplets.

**Number of Predicate Queries.** We compared BIG-C with different numbers of predicate queries (m) in Table 4. It can be observed that more predicate queries always improve the final VidSSG 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_r) in the RaCA module on BIG-C. From the results in Table 5, we can observe that both the R-norm and F_r 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)).

### 4.4.1 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 [35], GSTEG [40], VRD-GCN [30], MHA [36], IVRD [22], VidVRD-II [34], and TRACE [39]. 2) Tracklet-proposal based models methods: Liu et al. [24] and Chen et al. [13]. For more fair comparisons, we also reported the results of BIG-C with the same features as [24].

**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 [24], BIG-C outperforms TRACE [39] and Liu et al. [24] on both RelDet and RelTag tasks, especially we achieve significant performance gains on mAP (17.56%
vs. 14.01%). 3) When using the same RoI and I3D features as [24], 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 [36], IVRD [22], VidVRD-II [34], Liu et al. [24], Chen et al. [13], and two top-1 methods [37, 46] from Video Relation Understanding (VRU) Challenges. All results are reported in Table 3. It is worth noting that we only include 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) and (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).

**Acknowledgement.** This work was supported by National Key Research & Development Project of China (2021ZD0110700), National Natural Science Foundation of China (U19B2043, 61976185), Zhejiang Natural Science Foundation (LR19F020002), Zhejiang Innovation Foundation (2019R52002), and Fundamental Research Funds for Central Universities.

---

**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 [34]. Mask: It means the localization mask of entities [46].

| Models       | Detector   | Features Visual | Features Lang | Features Motion | Features Mask | RelDet (%) | RelTag (%) |
|--------------|------------|-----------------|---------------|-----------------|--------------|------------|------------|
| Liu et al. [24] | RefineDet  | Rol + I3D_r     | ✓             | ✓               | ✓            | 6.85       | 8.21       |
| Chen et al. [13] | Faster R-CNN | Rol + I3D_r     | ✓             | ✓               | ✓            | 10.04      | 8.94       |
| Chen et al. [13] | Faster R-CNN | Rol + I3D_r     | ✓             | ✓               | ✓            | 11.21      | 9.99       |
| IVRD [22] | Faster R-CNN | Rol             | ✓             | ✓               | ✓            | 7.42       | 7.36       |
| Chen et al. [13] | Faster R-CNN | Rol             | ✓             | ✓               | ✓            | 8.93       | 7.38       |
| VidVRD-II [34] | Faster R-CNN | Rol             | ✓             | ✓               | ✓            | 8.65       | 8.59       |
| BIG-C (Ours) | MEGA       | Rol + I3D_f     | ✓             | ✓               | ✓            | 8.03       | 7.60       |
| BIG (Ours)   | MEGA       | Rol + I3D_f     | ✓             | ✓               | ✓            | 8.28       | 7.74       |
| BIG-C (Ours) | MEGA       | Rol + I3D_f     | ✓             | ✓               | ✓            | 8.13       | 7.62       |
| BIG (Ours)   | MEGA       | Rol + I3D_f     | ✓             | ✓               | ✓            | 8.13       | 7.62       |

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

| #Bins | fR @5 (%) | fR @5 (%) | fR @5 (%) |
|-------|-----------|-----------|-----------|
| 1     | 12.96     | 15.59     | 16.76     |
| 5     | 13.07     | 15.83     | 17.26     |
| 10    | 13.04     | 15.89     | 17.61     |

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

| #Bins | fR @5 (%) | fR @5 (%) |
|-------|-----------|-----------|
| 1     | 12.96     | 15.59     |
| 5     | 13.07     | 15.83     |
| 10    | 13.04     | 15.89     |
Appendix

This supplementary document is organized as follows:

• More explanations about the temporal bipartite graph are in Sec. A.
• More detailed implementation details are in Sec. B.
• Statistics about predicates with multiple instances for multi-instances grounding are in Sec. C.
• More qualitative results are in Sec. D.
• Potential negative societal impact are in Sec. E.

A. More Explanation about Temporal Bipartite Graph

As shown in Figure 7, compared to existing video scene graphs (a), our temporal bipartite graph (b) have multiple advantages: 1) avoids exhaustively enumerating all entity pairs for predicate prediction; 2) is easier to model entity pairs with multiple predicates; and 3) can be easily extended to more general relations with more semantic roles (e.g., instrument [53]).

B. More Implementation Details

Tracklet Detector. We utilized the video object detector MEGA [14] with backbone ResNet-101 [18] to obtain initial frame-level detection results, and adopted deepSORT [43] to generate object tracklets. This detector was trained on a video set and an image set. The video set is a set of downsampled videos from the training set, and the downsample rate were set to 5 and 32 for VidVRD and VidOR, respectively. The image set consists of images with the same categories as the VidSGG dataset, which are selected from the training and validation set of MSCOCO [23] (for VidOR), or MSCOCO plus ILSVRC2016-DET [32] (for VidVRD).

Parameter Settings. The dimension of bounding box RoI feature $d_r$ was set to 1024, which was determined by MEGA. The dimensions of video I3D feature $d_i$ and word embedding $d_w$ were set to 1024 and 300, respectively. The hidden dimensions $d_q$ and $d_e$ were set to be 512. The output length $l_o$ of pooling operation was 4. The non-linear transforms $F_s$, $F_o$ are both MLPs. All the MLPs are two-layer FC networks with ReLU and the hidden dimension was 512. All bounding box coordinates and time slots are normalized to the range between (0, 1) w.r.t video size and video length, respectively. The loss factor was set as $\lambda = 30$. The parameters of DEBUG are set to the default settings [26].

Training Details. We trained our model for the classification stage and grounding stage separately. For the classification stage, the model was trained Adam [21] for total 80/60 epochs with batch size 8/4 for VidVRD and VidOR, respectively. The learning rate was set to 1e-4 for VidVRD. For VidOR, the learning rate was set to 5e-5 in the first 50 epochs and 1e-5 in the last 10 epochs. For the grounding stage, the model was trained using ground-truth triplet categories in VidOR as language queries. It was trained by Adam [21] with 70 epochs and batch size of 8. The initial learning rate was set to 5e-5, and it decays 5 times in the 40-th and 60-th epoch.

Inference Details. In classification stage, following previous works [35], we kept top-k triplet predictions (10 for VidVRD and 3 for VidOR) for each predicate node. Then, we filtered out duplicated triplets or triplets in which subject and object tracklets has no temporal overlap. In grounding stage, for each triplet query, we obtained $K$ time slots predictions through the multi-instance grounding, where each time slot prediction is associated with a score. Then, we filtered out these triplet queries whose highest score among all time slots is less than 0.2 (which might be false positives returned by the classification stage). For the remaining triplet queries, we add the time slot of subject-object overlapping (with score 1.0) to get total $K+1$ instances for more robust prediction. Finally, we apply temporal NMS (with the threshold of 0.8) to these $K+1$ instance, which results in $K_j$ time slots for each predicate $p_j$. The tracklets for the relation triplet are cropped from $e_{j_1}$, $e_{j_2}$ according to the time slots of $p_j$, i.e., each $p_j$ is corresponding to $K_j$ relation triplets.

C. Statistics for Multi-instance Predicates in VidOR

We reported the distribution of predicates with single instance or multiple instances in Figure 9(a). Here each sample is defined as the set of relation triplets with the same subject-object pair and predicate category. Then for those samples with multi-instance predicates, we reported the distribution of number of instances falling into the same bin for multi-instance grounding, as shown in Figure 9(b), where each sample is a bin.

From Figure 9(a), we can observe that although many predicates are single-instance, there are still around 32% of predicates with multiple ground-truth instances. Thus, the proposed multi-instance grounding is indispensable. Furthermore, Figure 9(b) shows that only 1.57% of bins are assigned with two instances or more, i.e., most of the bins are assigned with only one ground-truth target, which shows that our label assignment scheme is suitable for the multi-instance grounding.

D. More Qualitative Results

More qualitative results of our BIG model on VidOR are shown in Figure 8.

E. Potential Negative Societal Impact

Our proposed classification-then-grounding is a general framework of video scene graph generation (VidSGG), and there are no known extra potential negative social impact of our framework and BIG model. As for the challenging VidSGG task itself, there might be some wrong predictions (e.g., (adult, kick, dog)). When VidSGG is applied to numerous down-stream tasks such as video captioning, video question answering, these wrong predictions might result in some ethical issues, e.g., a wrong caption says that a person is abusing a dog. To avoid the potential ethical issues, we can introduce some common sense knowledge into VidSGG models and design some rule-based methods to filter out those unreasonable relation triplets that involve ethical issues.

References

[1] Lisa Anne Hendricks, Oliver Wang, Eli Shechtman, Josef Sivic, Trevor Darrell, and Bryan Russell. Localizing mo-
Figure 8. More qualitative results on VidOR validation set. The solid line and dash line represent the subject and object respectively.

Figure 9. Statistics for multi-instance grounding from VidOR train set. (a):Number of predicate instances with same category in a same subject-object pair. (b):Number of instances falling into the same bin for predicates with multiple instances.

ments in video with natural language. In ICCV, pages 5803–5812, 2017.

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

[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.

[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.

[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.

[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.

[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.

[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.
transformer hinging on cross-scale attention. In *ICLR*, 2022. 3

[43] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. Simple online and realtime tracking with a deep association metric. In *ICIP*, pages 3645–3649, 2017. 6, 9

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

[45] 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

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

[47] 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

[48] 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

[49] 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

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

[51] 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

[52] 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

[53] Alireza Zareian, Svebor Karaman, and Shih-Fu Chang. Weakly supervised visual semantic parsing. In *CVPR*, pages 3736–3745, 2020. 2, 3, 9

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

[55] 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

[56] 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