An Unified Deep Sparse Graph Attention Network for Scene Graph Generation

Hao Zhou[a], Yazhou Yang[b], Tingjin Luo*[c], Jun Zhang*[a], Shuohao Li[a]

[a] *Science and Technology on Information Systems Engineering Laboratory, National University of Defense Technology, Changsha, Hunan, China*
[b] *College of Advanced Interdisciplinary Studies, National University of Defense Technology, Changsha, Hunan, China*
[c] *College of Liberal Arts and Science, National University of Defense Technology, Changsha, Hunan, China*

Abstract

Scene graph generation plays an important role in deep understanding of the visual scene. Despite the empirical success of traditional methods in many applications, they still have several challenges in the high computational complexity of dense graph and the inaccurate pruning of sparse graph. To tackle these problems, we propose a novel deep sparse graph attention network to mine the rich contextual clues and simultaneously preserve the statistical co-occurrence knowledge of SGG. Specifically, our Relationship Measurement Network (RelMN) is adapted to first classify all object pairs in dense graph as the foreground and background categories to filter the false relationships and then construct a sparse graph efficiently. Meanwhile, we design a novel feature aggregation and update method via graphical message passing to jointly learn the node and edge features for object recognition and relationship classification in the graph attention network. Extensive experimental results on the large scale visual genome dataset demonstrate our proposed method outperforms several state-of-the-art approaches.

Keywords: scene graph generation, statistical co-occurrence knowledge, relationship measurement network, graph attention network, sparse graph

*Tingjin Luo and Jun Zhang are the corresponding authors. Tingjin Luo: tingjinluo@hotmail.com; Jun Zhang: zhangjun1975@nudt.edu.cn.

1. Introduction

Understanding the visual scene is a critical task in computer vision. Scene graph, a structured representation of an image [1], provides a deeper understanding of images than fundamental detection [2] and segmentation [3] by analyzing the semantic summaries of objects and their relationships. As shown in Figure 1, the scene graph displays the location and category of objects and exhibits the relationships between the objects, such as "boy-on-surfboard". Recently, inferring scene graph has attracted many researchers' attention [4, 5, 6, 7, 8], as it extracts the rich semantic information contained in the object interactions [9]. Such richer semantic understanding in scene graph can not only provide context clues for fundamental recognition tasks, but also have broad prospects in various high-level vision applications. For example, it is the key to improve the image retrieval [1] and the natural language based image tasks [10]. Besides, it provides the valuable information for other applications, such as VQA [11], image caption [12, 13] and image generation [8] etc.

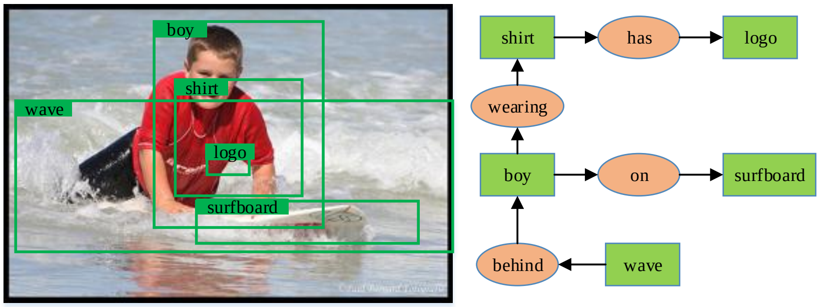


Figure 1: An example of a scene graph, the green boxes represent the object nodes, and the orange ellipses represent the relationships between the objects.

Despite the empirical success of scene graph generation methods in many practical applications, it still remains the challenging to accurately extract a scene graph from an image by reasoning about the complex dependencies between all components[5]. There are many methods [14, 15, 16, 17] already proposed to generate the scene graph in literature. These methods are divided into two categories according to the number of candidate relationships: dense

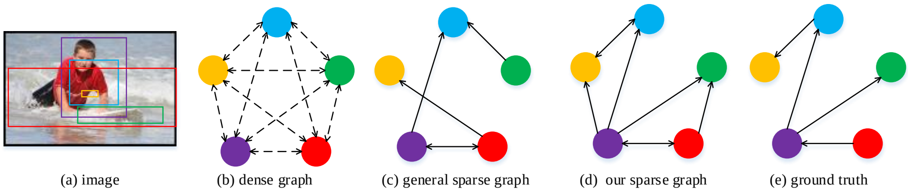


Figure 2: An illustration of dense graph and sparse graph. Given an image (a), all object pairs have edges in dense graph (b), and the general sparse graph (c) imprecisely prunes some edges. Our model generates a more reasonable sparse graph (d), and the ground truth is (e).

graph and sparse graph. Constructing dense graph is popular with scene graph generation [16], in which all object pairs have candidate relations and there are $n(n-1)$ connections for n nodes. With the dense graph, some researchers usually use CRF [15], BiLSTM [18], or GRU units [6] to build encoder-decoder model for message passing and context learning. Other researchers construct sparse graphs by pruning dense graphs [19] or constructing dynamic tree structures [16], which effectively captures contextual information and expresses the inherent parallel or hierarchical relationships. The graphical message passing with sparse graph often achieves by GCN or GAT [19]. Besides, Zellers *et al.* [18] found strong structural regularities in the visual scene, which is essential in the scene graph. The statistical dependence is formally expressed in the form of structured graphs in [20], and they show that the statistical correlation can be represented by a structured knowledge graph, which standardizes the semantic prediction space and deals with the uneven data.

However, there are still some shortcomings in the previous proposed methods that need to be improved. As shown in Figure 2(b), the dense graph is useful in modeling contextual clues, but might increase the computational complexity of the model and generate lots of redundancies for object features because many object pairs in image are unlikely to have attractive relationships [19]. On the contrary, partial reliable connections exist in the sparse graph, which improves the efficiency of message passing. An embarrassing fact is that current sparse graph models are unsatisfactory since they are unable to generate sufficiently accurate candidate edges for relationships classification, just like Figure 2(c).

On one hand, computing the distances and scores for object pairs is flawed. Previous works merely use the object categories features or appearance features to calculate the distances, which is insufficient. On the other hand, they often select top K object pairs to construct a sparse graph and the hyperparameter K is manual which is unsuitable for different scenes. Consequentially, the noisy relationship proposals may damage the scene graphs and their downstream tasks.

To tackle these problems, we design a Relationship Measurement Network (RelMN) for generating sounder relationship proposals and more reasonable sparse graph. Different with general relationship proposal networks, our RelMN is aimed to distinguish whether the relationships between object pairs are foreground (annotated edges) or background (not annotated edges) rather than computing the distances or correlation for object pairs, such as RePN [19], Rel-PN [4] and VCTree [16]. To make up for the shortcomings of single-species features, the RelMN takes multiple features as the inputs for the binary classification of relationships, including appearance features, categories features, spatial features of objects and the prior statistical features. Meanwhile, the network automatically selects all K foreground edges from the classifier outputs and some background edges to construct the sparse graph without any hyperparameter, as shown in Figure 2(d). Specially, top $3K$ background edges with high foreground scores from the softmax outputs are chosen in our model.

In this paper, we propose a novel unified deep sparse graph attention network for scene graph generation. Our model is mainly committed to achieving efficient message passing and mining rich contextual clues by constructing a more reasonable sparse knowledge graph. In the data preprocessing, we calculate the prior statistical probability of object categories and object categories-relationships. The model generates a set of objects proposal based on Fast R-CNN detector [21] in bounding box module. In sparse graph module, the RelMN first classifies the relationships into foreground and background for $n(n-1)$ edges. Then it screens all edges with significant relationships and partial background edges to build the directed sparse graph automatically. In graphical message passing module, the Graph Attention network (GAT) with multi-head attention achieves a new way of features aggregations and updates on the sparse graph. In previous models, relationship features are rough because they are usually determined by object features. With our graphical message passing module, object features and relationships features are learned synchronously. The prior probability will be embedded with the object features to realize the co-occurrence structure modeling and to form the sparse knowledge graph. Extensive experiments on the large scale visual genome (VG) dataset [22] demonstrate the performance of our model. Compared with the current state-of-the-art methods KERN [20], VCTree [16] and GPS-Net [23], the proposed method achieves absolute gains of 2.60, 1.60, and 0.82 on the average of recall@50 and recall@100 measures of three common tasks. We also perform extensive model ablation and analyze the influence of various factors on model performance.

Our contributions are summarized as follows:

(1) We propose a novel feature interaction and knowledge learning framework on sparse graph for scene graph generation. It incorporates the sparse graph construction, graphical message passing, and learning of statistical knowledge.

(2) The Relationship Measurement Network (RelMN) classifies all edges in dense graph into foreground and background and automatically construct a reliable sparse graph to reduce the computational complexity and improve the efficiency of feature interaction.

(3) We design a new feature aggregation and update method between nodes and edges based on the graph attention network, which jointly extracts the object features and relationship features. Besides, the attention weights are learned from subject node features, object node features, and the edge features with multiple subspaces to mine rich contextual clues.

(4) A sparse knowledge graph are explicitly implemented with the co-occurrence structure modeling. We fully explore the role of prior statistical probability, which has been successfully applied to the relationship binary classification and graphical message passing.

The remainder of this paper is organized in the following manner. Section 2 briefly overviews the related works. Then we introduce the proposed methodology in Section 3. The experimental setting, results, and discussions are reported in Section 4. Finally, we draw conclusions in Section 5.

2. Related works

Relationship detection and scene graph generation. Visual relationship detection detects semantic objects in images and infers the relationship between object pairs. In the past decade, many scholars have done a lot of research on visual relationship detection [24, 25]. Early studies focused on specific relational phrases [26] or visual phrases to improve other tasks [27]. Recent researches pay more attention to general visual relationship detection [28, 29], such as geometric relationships, affiliation, and action. A scene graph is a structured representation of visual scene based on relationship detection. One of the most popular ways to represent scene understanding is text description (such as image caption [30]). The text description is usually limited by ambiguity and lack of expressiveness. In contrast, the scene graph provides information about the locations of objects in the scene and the relationships between objects. In recent years, many scene graph analysis methods have been proposed based on a same pipeline. The pipeline is that these models first detect entities by using off-the-shelf detectors [31, 32] or fine-tuned detectors on relationship datasets [33, 4]. Then they predict predicates using the recommended method. Our model also follows this pipeline.

Message passing with dense graph and sparse graph. The idea of improving scene understanding with context has a long history in computer vision [34, 35]. Message passing is a way to integrate context information in scene graph. Lin *et al.* [23] proposed a direction-aware message passing module to extract the edge direction information. In [6], Xu *et al.* decomposed message passing into two sub-graphs for objects and relationships, and performed message passing. Similarly, in [28], they proposed two messaging strategies (parallel and sequential) for spreading information between objects and relationships. However, their message passing strategies are flawed. The indiscriminate message passing between all objects, which forms the dense graph, might make features learning sloppy, and exponentially increase the computational complexity. Some scholars use heuristic methods such as random sampling to solve this problem [15, 36, 37]. Tang *et al.* [16] recommended to compose the dynamic tree structure to put the objects into the visual context. To prune the meaningless relationship edges, the trainable Relation Proposal Network (RePN) is introduced in [19], which is similar to the recently proposed Relationship Proposal Network (Rel-PN) [4]. Different with RePN and Rel-PN, the RelMN first integrates the multiple features to judge the relationships between object pairs. Then, a sparse graph is automatically constructed for each image according to the binary classification.

Statistical dependence in scene graph generation. Many methods have focused on investigating the importance of the regularities of objects and relationships in scene graphs generation. Visual scene understanding usually relies on statistical patterns [38, 39] and spatial layout [40] of objects. In [18], Zeller *et al.* analyzed the statistical dependences of object pairs and relationships on the VG dataset, and concluded that they could provide powerful regularization for relationship prediction. They came up with a strong baseline. And this baseline directly used frequency priors to predict relationships and integrated the regularity into the graph structure, surpassing most previous studies. In [15], it is pointed out that the entropy of the prior probability distribution $P(R)$ on the VG dataset is 2.88, but the entropy of $P(R|S,O)$ is 1.21 given the conditional probability of object pairs (subject node and object node). This difference demonstrates the importance of statistical dependence between objects and relationships. Works [18, 15] also noted statistical dependences, and they designed deep models to mine the information through message passing implicitly. The model of [20] formally expressed statistical co-occurrence knowledge and incorporated the graph into deep propagation network to promote scene graph generation. We also take the prior statistical probability as part of learning statistical occurrence knowledge to generate the sparse knowledge graph, rather than the dense knowledge graph.

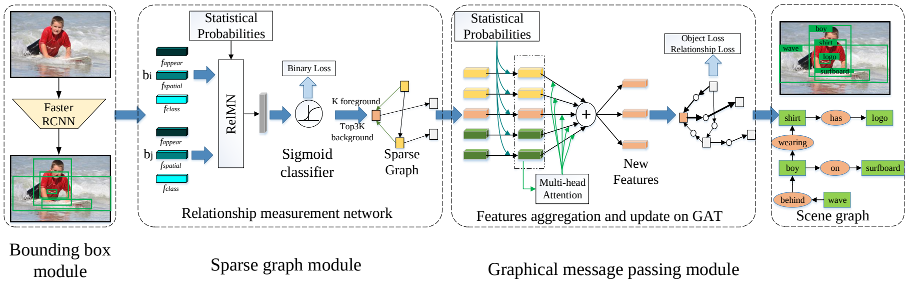


Figure 3: A brief pipeline of our model. It include three modules: Bounding box module adopt the Faster RCNN to generate object proposals; Sparse graph module divides all edges of object pairs into foreground relationships and background by the binary classifier and generates the reasonable sparse graph. The key of this module is our novel relationship measurement network (RelMN); Graphical message passing module includes a new feature aggregation and update way between nodes and edges based on the GAT to achieve effective contextual learning and recognize the object categories and their relationships. The statistical probabilities are embedding into the sparse graph module and graphical message passing module to generate the sparse knowledge graph and learn the statistical co-occurrence modeling.

3. Method

As illustrated in Figure 3, we summarize a brief pipeline of our model. We first show the model overview and problem formulation in Section 3.1. Then our model can be decomposed the following three steps. (a) The object bounding boxes in images are extracted through the off-the-shelf Faster R-CNN in Section 3.2. (b) In Section 3.3, a novel relationship measurement network divides all edges of object pairs into foreground relationships and background by the binary classifier, and generates a reasonable sparse graph for each image. (c) In Section 3.4, we show a new feature aggregation and update way based on multi-head graph attention network for efficient message passing and relationship classification on the sparse graph. It is worth mentioning that to better play the role of statistical dependence, we incorporate the prior statistical probability into Section 3.3 and Section 3.4 respectively, and successfully construct the sparse knowledge graph.

3.1. Model overview and problem formulation

In this work, a structured representation of an image has generated for the scene graph, which mainly consists of object locations, object categories, and the relationships between them. Given a scene graph G, it can be represented as a set of triples $G = \{B, O, R\}$:

$B = \{b_1, b_2, ..., b_n\}$ is the region candidate set. For the i-th candidate region, $b_i \in \mathbb{R}^4$ expresses coordinates location information.

$O = \{o_1, o_2, ..., o_n\}$ represents the object set. The candidate region b_i has a corresponding category label $o_i \in C$, where C represents the collection of all object categories.

$R = \{r_1, r_2, ..., r_m\}$ is the binary relationship set. The relationship r_i is a triple, including: the subject node $(b_i, o_i) \in B \times O$, the object node $(b_j, o_j) \in B \times O$, and the relationship between them $r_{i \to j} \in \mathcal{R}$. $\mathcal{R}$ represents the all relationships set.

Given an image I, the model decomposes the probability of a graph G into three factors:

$$P(G|I) = P(B|I)P(E|I, B)P(R, O|I, B, E), \tag{1}$$

where $E \subseteq \begin{pmatrix} B \\ 2 \end{pmatrix}$ and $\begin{pmatrix} B \\ 2 \end{pmatrix}$ represents the edges connected by any two object bounding boxes.

Bounding box module $P(B|I)$ is the basis of the scene graph, which extracts the proposal candidates from the image and provides the location information. Similar to the previous methods, we use Fast R-CNN, a widely used object detection model, to obtain these bounding boxes which covers most of the critical objects.

Sparse graph module $P(E|I, B)$ selects the object pairs with potential relationships to construct the sparse graph. Our RelMN is different from traditional methods from the following two aspects. First, the main target of RelMN is to divide all edges into two categories: foreground and background rather than to compute the distances between objects. Second, all foreground edges and partial background edges are automatically selected to build the sparse graph in RelMN.

Graphical message passing module $P(R, O|I, B, E)$ achieves effective contextual learning and recognizes the object categories and their relationships. First, the fused features form new node features and edge features on the graph. Then, the messages on the graph are efficiently transmitted and integrated through the graph attention network with a new feature aggregation and update way.

3.2. Extraction of object bounding boxes

In the bounding box module, the model generates a set of candidate regions. Similar to other studies [18, 20], we use the Fast R-CNN framework as the underlying detector to automatically extract object regions $B = \{b_1, b_2, ..., b_n\}$. For region b_i, it includes the location information represented by bounding box $b_i \in \mathbb{R}^4$, the object appearance features $\boldsymbol{f}^i_{appear}$, and the object category probability $\boldsymbol{p}_i$.

3.3. A novel relationship measurement network

We design a novel relationship measurement network (RelMN) to identify edges of all object pairs into foreground or background and construct sparse graph. Compared to dense graphs, message passing on sparse graphs can significantly reduce the computational complexity, making information transfer more accurate and effective. RelMN consists of three components: (1) extraction of multiple features, (2) binary classification of foreground and background, and (3) generation of the sparse graph. Figure 4 shows the process of constructing sparse graph with our RelMN.

Extraction of multiple features. In terms of experience, two aspects are utilized to judge the meaningful relationship between object pairs. One aspect is the spatial distances. Compared with the distant object pairs, the closer object pairs are more likely to have meaningful relationships. Another aspect is the object categories. It could happen that meaningful relationships do not exist between some object categories, even if they are usually packed together, e.g.

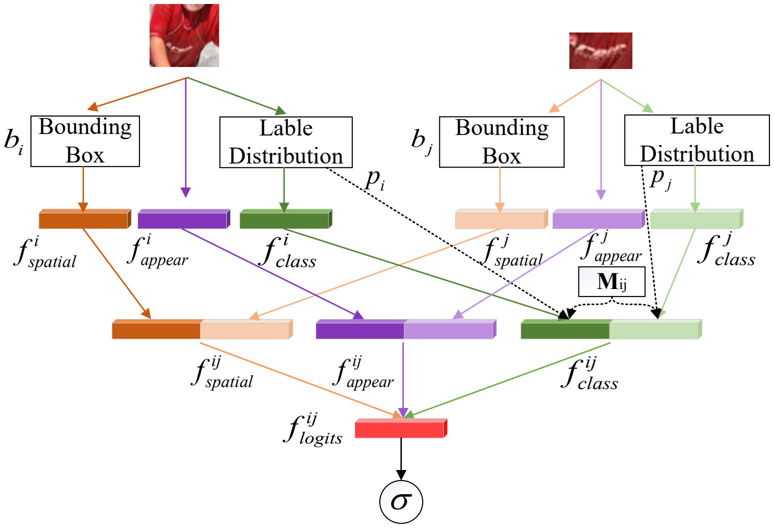


Figure 4: Binary classification for object pairs in RelMN. First, we transfer the location coordinates and category probability distributions of the objects into spatial features and category features. The category features are equipped with the prior statistical features. Then the union spatial features, category features and appearance features are concatenated to generate the logits features. Finally, the output probabilities are computed by the sigmoid classifier.

hats and glasses. On the contrary, though the distance between some categories may be a little larger, they are more likely to have a meaningful relationship (such as “person-play-football”). Therefore, the RelMN will mainly rely on the objects category features, spatial features, and appearance features to determine the type of edges for all object pairs.

We transform the location coordinates and category probability distributions of the objects into spatial features and category features. For spatial features, the location coordinates of $M \times 4$ dimensions are expanded to $M \times 16$ dimensions through duplication and concatenation, where M is the number of objects. Then it is transformed into $M \times 128$ dimensional spatial feature vectors: $\boldsymbol{f}^i_{spatial} = MLP([b_i, b_i])$, where MLP is a multi-layer perceptron, $[\cdot, \cdot]$ is a concatenation.

There are 151 categories in the VG dataset, and the object generates a probability distribution of 151-dimensional vector. Similarly, we transform the category probability distribution of the object into the category features of

$M \times 128$: $\boldsymbol{f}^{i}_{class} = MLP(\boldsymbol{p}_i)$.

Binary classification of foreground and background. For an edge of two object regions b_i and b_j $(i \neq j)$, the RelMN classifies the edge based on the appearance features, spatial features, and category features of b_i and b_j. The union appearance features $\boldsymbol{f}^{ij}_{appear}$ and the union spatial features $\boldsymbol{f}^{ij}_{spatial}$ of b_i and b_j are computed as:

$$\begin{cases} \boldsymbol{f}^{ij}_{appear} = MLP([\boldsymbol{f}^{i}_{appear}, \boldsymbol{f}^{j}_{appear}]), \\ \boldsymbol{f}^{ij}_{spatial} = MLP([\boldsymbol{f}^{i}_{spatial}, \boldsymbol{f}^{j}_{spatial}]). \end{cases} \tag{2}$$

To learn the statistical co-occurrence knowledge for categories, we introduce the prior statistical probabilities of object categories when calculating the union category features. The statistical information plays a vital role in associating objects and predicting the object labels. The prior probability matrix of object categories is $\mathbf{M}_{st} \in \mathbb{R}^{C \times C}$, where C is the number of all categories. For two different categories of s and t, the probability that there exist at least two objects belonging to s and t in an image can be computed as:

$$m_{st} = \frac{Num(t \cap s)}{Num(s)}. \tag{3}$$

The model learns the statistical co-occurrence knowledge among object categories based on the prior probability matrix and the union category features. The union category features $\boldsymbol{f}^{ij}_{class}$ of b_i and b_j can be formulated as:

$$\boldsymbol{f}^{ij}_{class} = MLP\Big([\sum_{t=1}^{C} p_{i_s} m_{st} \boldsymbol{f}^{i}_{class}, \sum_{s=1}^{C} p_{j_t} m_{ts} \boldsymbol{f}^{j}_{class}]\Big), \tag{4}$$

where p_{i_s} represents the probability of the i-th node belonging to category s.

The logits of edge $\boldsymbol{f}^{ij}_{logits}$ and the probabilities $\boldsymbol{p}_{e_{ij}}$ can be computed as:

$$\begin{cases} \boldsymbol{f}^{ij}_{logits} = MLP([\boldsymbol{f}^{ij}_{appear}, \boldsymbol{f}^{ij}_{spatial}, \boldsymbol{f}^{ij}_{class}]), \\ \quad \boldsymbol{p}_{e_{ij}} = \sigma(\boldsymbol{f}^{ij}_{logits}), \end{cases} \tag{5}$$

where σ is the sigmoid function. Multiple features provide more supports for edges binary classification than single-species features.

Generation of the sparse graph. With the outputs of the classifier, the RelMN first select all K object pairs whose edges are predicted as the foreground. Note that K is not a hyperparameter because it is directly determined by the classifier. In addition, the RelMN still automatically select top $3K$ background edges with high foreground probabilities to construct the sparse graph with the K foreground edges. The background edges enhance the robustness of relationship classification and reduce the risk of filtering out ground truth relationships. Finally, we can get a sparse graph containing n nodes and $4K$ edges for an image. In general, the RelMN generates sounder relationship proposals and more reasonable sparse graph than previous works.

3.4. Graphical message passing on graph attention network

To learn contextual clues through the sparse graph, we propose a new feature aggregation and update way based on multi-head graph attention network. The relationship features in previous methods are usually determined by object features. After message passing, they only extracts the object features that are used for objects recognition. The relationship features are simply calculated by the concatenation of object features, which is rough and cannot well express the difference between a node as a subject or object. In our graphical message passing module, the object features and relationships features are learned synchronously by nodes and edges on sparse graph. Specifically, the message passing on our GAT is based on directed sparse graph, which explicitly indicates the subject nodes and object nodes. The inherent weight from prior statistical probability and the attention weights of nodes and edges on our GAT are beneficial to extract object features and relationship features. This new method mainly includes: (a) the generation of node and edge features with knowledge embedding, (b) the weights learning of the node features and the edge features based on the inherent weight and attention, (c) learning different distribution from multiple subspaces, and (d) objects and relationships classification based on the node features and edges features.

Generation of node features and edge features with knowledge em-

bedding. The model first aggregates appearance features, spatial features, and category features and compresses them by an encoder-decoder to obtain fusion object features $\boldsymbol{f}^i_{fusion}$. Then we assign node features and edge features to the corresponding nodes and edges in the graph. The node features $\boldsymbol{h}^0_{v_i}$ are initialized by using the fused object features, i.e. $\boldsymbol{h}^0_{v_i} = \boldsymbol{f}^i_{fusion}$. For the initialization of edge features $\boldsymbol{h}^0_{e_{ij}}$, the features of the subject node and object node will be concatenated sequentially, and then dimensionally compressed through the fully connected layer, i.e. $\boldsymbol{h}^0_{e_{ij}} = L([\boldsymbol{f}^i_{fusion}, \boldsymbol{f}^j_{fusion}])$, where L is LeakyRelu layer.

Next, the prior statistical probabilities of object categories-relationships embed with the initialized node features and edge features. Specifically, in the prior probability matrix of categories-relationships, the probability m_{str} of all possible relationships given the node s and t is:

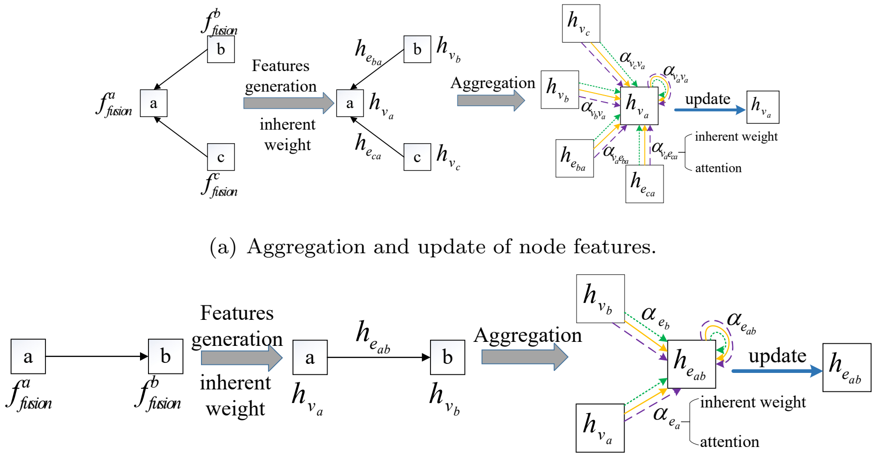
$$m_{str} = \frac{Num(r \cap (s \to t))}{Num(s \to t)}, \tag{6}$$

where $s \to t$ represents the subject node s and the object node t. The prior probability matrix is $\mathbf{M}_{str} \in \mathbb{R}^{C \times C \times R}$ by calculating the probabilities for all categories-relationships, where R is the number of all relationships. Therefore, the node features $\boldsymbol{h}_{v_i}$ and edge features $\boldsymbol{h}_{e_{ij}}$ can be computed as:

$$\begin{cases} \boldsymbol{h}_{v_i} = \displaystyle\sum_{r=1}^{R}\sum_{s=1}^{C}\sum_{t=1}^{C} p_{i_s} m_{str} \boldsymbol{h}^0_{v_i}, \\ \boldsymbol{h}_{e_{ij}} = \displaystyle\sum_{r=1}^{R}\sum_{s=1}^{C}\sum_{t=1}^{C} p_{i_s} m_{str} p_{j_t} \boldsymbol{h}^0_{e_{ij}}, \end{cases} \tag{7}$$

where p_{i_s} represents the probability of the i-th node belonging to category s, and m_{str} is the probability of the subject node of category s and the object node of category t having a relationship r.

As another part of the sparse knowledge graph, the prior probability m_{str} and category probability p_i form the inherent weight of nodes and edges in (7). The inherent weight of nodes and edges reflects the category-relationships activity of between the nodes v_i and other nodes, and between the edges e_{ij} and other edges in the VG dataset, respectively.



(a) Aggregation and update of node features.

(b) Aggregation and update of edge features.

Figure 5: The aggregation and update for node features and edge features. The inherent weight embed with the fused features to acquire the nodes features and the edges features. Then the nodes aggregate the features of adjacent nodes and connected edges, and the edges aggregate the subject node features and object node features with the attention weights from multiple subspaces.

The weights learning of the subject node features, object node features, and the edge features. Before describing our model, we briefly review the graph attention network (GAT) [41]. For the node i in the graph, given the features $\boldsymbol{h}_i$ and the expression of its neighbor node $\{\boldsymbol{h}_j | j \in No(i)\}$, where $No(i)$ represents the neighbor node of i, the graph attention weight α_{ij} can be calculated as:

$$\alpha_{ij} = \frac{exp\left(L(\vec{a}^T[\boldsymbol{W}\boldsymbol{h}_i || \boldsymbol{W}\boldsymbol{h}_j])\right)}{\sum_{k \in No(i)} exp\left(L(\vec{a}^T[\boldsymbol{W}\boldsymbol{h}_i || \boldsymbol{W}\boldsymbol{h}_k])\right)}, \tag{8}$$

where $\vec{a}$ is the weight vector of a layer feed-forward neural network, and $\boldsymbol{W}$ is the learning weight parameter. Then, the normalized attention coefficients between different nodes are calculated, which predict the output features $\boldsymbol{h}'_i$:

$$\boldsymbol{h}'_i = \sigma\Big(\sum_{j \in No(i)} \alpha_{ij} \boldsymbol{W}\boldsymbol{h}_j\Big). \tag{9}$$

In our model, the node needs to aggregate the features of adjacent nodes and connected edges. Simultaneously, the edge features also are aggregated and updated based on the subject node features and object node features. Therefore, we need to calculate the attention weight between nodes, and between nodes and edges.

For node message aggregation, the attention weight $\alpha_{v_iv_j}$ between nodes is calculated as follows:

$$\alpha_{v_iv_j} = \frac{F(\boldsymbol{W}\boldsymbol{h}_{v_i}||\boldsymbol{W}\boldsymbol{h}_{v_j})}{\sum\limits_{k\in No(i)} F(\boldsymbol{W}\boldsymbol{h}_{v_i}||\boldsymbol{W}\boldsymbol{h}_{v_k}) + \sum\limits_{l\in Ne(i)} F(\boldsymbol{W}\boldsymbol{h}_{v_i}||\boldsymbol{W}\boldsymbol{h}_{e_{il}})}, \tag{10}$$

where $F(x) = exp(LeakyReLU(\vec{a}^T(x)))$, $Ne(i)$ represents the set of edges connected to node i. The attention weight $\alpha_{v_ie_{ij}}$ between nodes and edges is calculated as:

$$\alpha_{v_ie_{ij}} = \frac{F(\boldsymbol{W}\boldsymbol{h}_{v_i}||\boldsymbol{W}\boldsymbol{h}_{e_{ij}}])}{\sum\limits_{k\in No(i)} F(\boldsymbol{W}\boldsymbol{h}_{v_i}||\boldsymbol{W}\boldsymbol{h}_{v_k}]) + \sum\limits_{l\in Ne(i)} F(\boldsymbol{W}\boldsymbol{h}_{v_i}||\boldsymbol{W}\boldsymbol{h}_{e_{il}}])}. \tag{11}$$

For the aggregation of edge messages, the attention weight α_{e_l} is:

$$\alpha_{e_l} = \frac{F([\boldsymbol{W}\boldsymbol{h}_{e_l}||\boldsymbol{W}\boldsymbol{h}_{v_i}])}{\sum_{j\in No(l)} F([\boldsymbol{W}\boldsymbol{h}_{e_l}||\boldsymbol{W}\boldsymbol{h}_{v_j}])}, \tag{12}$$

where $No(l)$ represents the subject node and object node connected to edge l. From Eq. (7), (10), (11), and (12), we can know that different from general GAT, the aggregate weight of node features or edge features have be guided by attention and inherent weight in our model.

Learning different distribution from multiple subspaces. The model merges the aggregated features with the previous hidden state by multi-head attention, each of which represents a subspace. The new node features $\tilde{\boldsymbol{h}}_{v_i}$ and edge features $\tilde{\boldsymbol{h}}_{e_{ij}}$ are updated through the non-linear activation function:

$$\begin{cases} \tilde{\boldsymbol{h}}_{v_i} = \sigma\left(\dfrac{1}{Z}\sum\limits_{z=1}^{Z}(\sum\limits_{k\in No(i)}\alpha^z_{v_iv_k}\boldsymbol{W}^z\boldsymbol{h}_{v_k} + \sum\limits_{l\in Ne(i)}\alpha^z_{v_ie_{il}}\boldsymbol{W}^z\boldsymbol{h}_{e_{il}})\right), \\ \tilde{\boldsymbol{h}}_{e_{ij}} = \sigma\left(\dfrac{1}{Z}\sum\limits_{z=1}^{Z}(\alpha^z_{e_{ij}}\boldsymbol{h}_{e_{ij}} + \alpha^z_{e_i}\boldsymbol{W}^z\boldsymbol{h}_{v_i} + \alpha^z_{e_j}\boldsymbol{W}^z\boldsymbol{h}_{v_j})\right), \end{cases} \tag{13}$$

where Z represents the number of subspaces, and z represents the z-th attention. It should be noted that we have added an edge for each node pointed to itself, i.e.

$i \in No(i)$. The node features are updated according to the hidden node features, adjacent nodes features, and connecting edges features. The edge features are updated according to the hidden edge features, subject node features, and object node features. Figure 5 shows the aggregation and update process of node features and edge features.

Under the guidance of the prior probability matrix of categories-relationships and the multi-head attention mechanism, the model can focus on learning statistical co-occurrence knowledge and contextual cues in the large-scale dataset. The model classifies the objects and relationships by the output features. Specifically, the object categories are classified according to the output node features, and the relationships are classified by the output edge features.

4. Experiments and Results

In this section, we first describe experimental settings in detail in section 4.1. Then the experimental results and comparison with other latest studies are shown in Section 4.2. Finally, we present ablative study and qualitative results in Section 4.3 and Section 4.4, respectively.

4.1. Experimental settings

Dataset: The VG dataset is the validated dataset for our proposed model. There are several inconsistencies for dividing and evaluating the VG dataset in the different scene graph generation branches [36, 6, 4]. In our experiment, we employ the most commonly used preprocessing and data partitioning models [6]. The original VG dataset contains 108077 images. After preprocessing, an image contains 25 different objects and 22 relationships on average. As in [18, 6, 20, 19], we select the most frequent 150 object categories and 1 background category, and 50 predicate relationships as the evaluation criteria. Within the range of these categories and predicate relationships, an image contains 11.5 objects and 6.2 relationships on average. 70% of the images are selected as the training set and the other 30% images are testing set.

Model training. We trained our model in the PyTorch framework. To be consistent with the previous work [18, 20], we adopt two-stage training, in which we first train the object detector, and then conduct the joint training of the entire scene graph generation model. In the object detection, the Fast R-CNN detector generates a series of candidate regions. The detector is based on VGG-16 [42] as the backbone, and pre-trained and initialized by the ImageNet dataset[43]. The input image size is 592×592 and the anchor sizes and dimension ratios are similar to Yolo-9000 [44].

Task settings. The goal of scene graph generation is to predict a series of subject-relationship-object triples. We use the following three standard measures to evaluate the model of scene graph generation:

Predicate classification (PredCls): Given ground truth of object category and bounding box, the model predicts the relationships of object pairs.

Scene graph classification (SGCls): Given ground truth of object bounding box, the model predicts the object categories and relationships of object pairs.

Scene graph generation (SGGen): Model needs to detect and identify the objects in the image, and predict the relationships for all object pairs.

Evaluation metrics. We mainly use Recall@K as the primary performance measure, which is the proportion of true instances correctly recalled in the Top K predictions. In [20, 16], they propose to use the mean recall@K (mR@K) to evaluate the performance of all relationships more comprehensively. This metric calculates R@K for each relationship, and then averages the R@K of all relationships to obtain mR@K. Therefore, we adopt the Recall@K and mean Recall@K measures to evaluate our model in experiments, specifically including Recall@20, Recall@50, Recall@100 and mR@50, mR@100. Besides, some previous works [6] calculate R@K under the constraint that only a relationship is obtained for a given object pair. Some other works [45] omit this constraint so that multiple relationships can be obtained for an object pair, resulting in higher scores. In this work, we give a comprehensive comparison of R@K and mR@K with and without constraints, respectively.

Table 1: Comparison of the R@20, R@50 and R@100 in percentage with and without constraint on the three tasks of the VG dataset.

		SGGen			SGCls			PredCls			
	Method	R@20	R@50	R@100	R@20	R@50	R@100	R@20	R@50	R@100	Mean
Constraint	VRD [14]		0.30	0.50		11.8	14.1		27.9	35.0	14.93
	Graph RCNN [19]		11.4	13.7		29.6	31.6		54.2	59.1	33.27
	IMP+ [6]	14.6	20.7	24.5	31.7	34.6	35.4	52.7	59.3	61.3	39.30
	Freq+O [18]	20.1	26.2	30.1	29.3	32.3	32.9	53.6	60.6	62.2	40.72
	SMN [18]	21.4	27.2	30.3	32.9	35.8	36.5	58.5	65.2	67.1	43.68
	KERN [20]	21.7	27.1	29.8	33.8	36.7	37.4	59.2	65.8	67.6	44.07
	VCTree [16]	22.0	27.9	31.1	35.2	38.1	38.8	60.1	66.4	68.1	45.07
	GPS-Net [23]	22.6	28.4	31.7	36.1	39.2	40.1	60.7	66.9	68.8	45.85
	Ours	**23.2**	**28.8**	**32.9**	**37.5**	**40.2**	**41.1**	**62.6**	**67.7**	**69.3**	**46.67**
No constraint	AE [45]		15.5	18.8		35.7	38.4		82.0	86.4	46.13
	IMP+ [6]		22.0	27.4		43.4	47.2		75.2	83.6	49.80
	Freq+O [18]		28.6	34.4		39.0	43.4		75.7	82.9	50.67
	SMN [18]		30.5	35.8		44.5	47.7		81.1	88.3	54.65
	KERN [20]		30.9	35.8		45.9	49.0		81.9	88.9	55.40
	Ours		**31.6**	**37.0**		**49.4**	**52.8**		**83.4**	**89.8**	**57.33**

4.2. The results of the experiments

According to the data preprocessing and data division methods in Section 4.1, we evaluate and compare the proposed model on the VG dataset. In this section, we compare the results with other start-of-the-art methods: Visual Relationship Detection(VRD) model [14], Graph RCNN model [19], improved version of Iterative Message Passing method (IMP+) [6], associative embedding model (AE) [45], the best frequency baseline (Freq+O/Freq), Stacked Motif Networks with LeftRight (SMN-L/SMN) in [18], and currently the start-of-the-art models Knowledge-Embedded Routing Network (KERN) [20], visual context tree model (VCTree) [16], and GPS-Net [23]. For a fair comparison, the mean value in the last column is calculated for R@50 and R@100. The best performance is highlighted in bold face.

Table 1 shows the performance of our model and other start-of-the-art models on the Recall@20, Recall@50, and Recall@100 measurements. The VRD, Graph RCNN, and IMP+ methods in Table 1 take language models or global context to extract the relationship features between objects. Still, their performance on the VG dataset are worse than the Freq baseline method. The Freq baseline method directly predicts the most frequent relationship of object

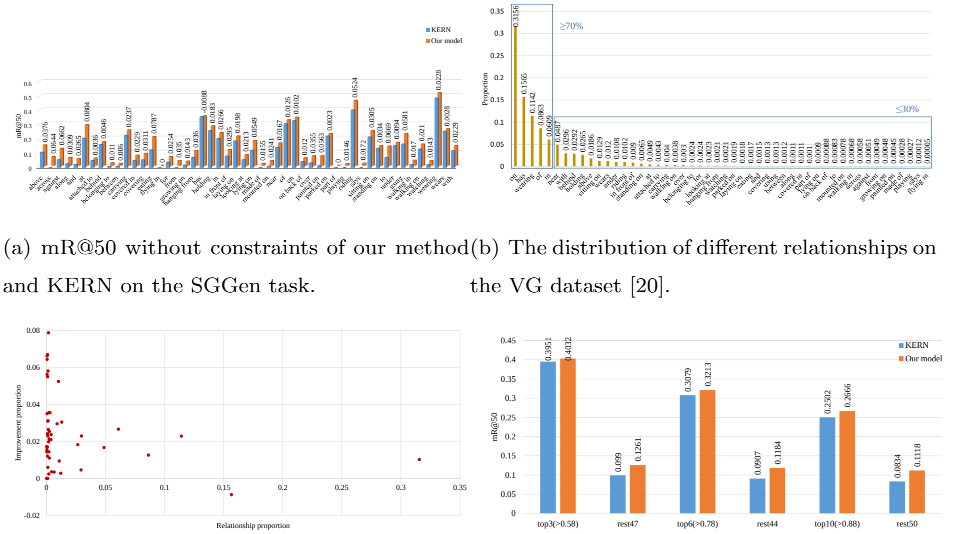
Table 2: Comparison of the mR@50 and mR@100 in percentage with and without constraint on the three tasks of the VG dataset.

		SGGen		SGCls		PredCls		
	Method	mR@50	mR@100	mR@50	mR@100	mR@50	mR@100	Mean
Constraint	IMP [6]	0.6	0.9	3.1	3.8	6.1	6.8	3.75
	IMP+ [6]	3.8	4.8	5.8	6.0	9.8	10.5	6.78
	SMN-L [18]	5.3	6.1	7.1	7.6	13.3	14.4	8.97
	KERN [20]	6.4	7.3	9.4	10.0	17.7	19.2	11.67
	VCTree [16]	6.9	8.0	10.1	10.8	17.9	19.4	12.18
	Ours	**7.8**	**8.9**	**11.8**	**12.3**	**21.0**	**22.3**	**14.02**
No constraint	AE [45]	1.6	2.5	6.0	7.8	15.1	19.5	8.75
	IMP+ [6]	5.4	8.0	12.1	16.9	20.3	28.9	15.27
	Freq [18]	5.9	8.9	13.5	19.6	24.8	37.3	18.33
	SMN [18]	9.3	12.9	15.4	20.6	27.5	37.9	20.60
	KERN [20]	11.7	16.0	19.8	26.2	36.3	49.0	26.50
	Ours	**14.3**	**18.5**	**24.0**	**30.5**	**42.5**	**54.2**	**30.67**

pairs with given category labels, which shows that the statistical dependence between object categories and their relationships also occupies an important role in object relationship detection and scene graph generation tasks.

The most similar to our model pipeline is the Graph RCNN model, but our model gets 13.40% improvement than the Graph RCNN model on the VG dataset. Compared with them, the proposed model not only obtains more accurate relationship recommendations and achieves better graphical message passing, but also integrates the prior statistical probabilities and learn the statistical co-occurrence knowledge.

The SMN model, KERN model, VCTree model and the GPS-Net are the best methods in the existing research on the VG dataset. The SMN model implicitly captures these statistical correlations by encoding the global context. Under the evaluation settings with and without constraints, its mean score is 43.68% and 54.65%, respectively. The KERN model explicitly merges statistical correlations through prior statistical probabilities, and its mean score is 44.07% and 55.40% with and without constraints, respectively. However, both the SMN model and the KERN model apply dense graphs to transfer messages between objects. The VCTree model constructs a dynamic tree structure that



(a) mR@50 without constraints of our method and KERN on the SGGen task.

(b) The distribution of different relationships on the VG dataset [20].

(c) The relation between mR@50 improvement and sample proportion.

(d) mR@50 of our method and KERN on the SGGen with three groups.

Figure 6: The comprehensive analysis and comparison of our model and KERN from the perspective of relationships distribution. Our model achieves better performance on the scene graph generation (SGGen) task, and both the frequent relationships and the relationships with fewer samples get the improvement.

405 allows more content or task-specific messages to be passed between objects. Its mean score is 45.07% under constraints. In our model, we select the object pairs with significant relationships and build the sparse graph by RelMN. On the sparse graph, the model performs more accurate and efficient message passing, and fuses contextual cues based on the GAT with new feature aggregation 410 and update way. Therefore, our model can be improved by 2.99% and 2.60%, respectively compared with SMN model and KERN model with constraints, and 2.68% and 1.93% higher without constraints. Compared with the VCTree model and GPS-Net, it is 1.60% and 0.82% higher with constraints.

For more comprehensive comparison with existing methods, we also present 415 mR@50 and mR@100 for three tasks in Table 2. Our method achieves the best results on these tasks. Specifically, the mean score is 14.02% with constraints, which is 5.05%, 2.35%, and 1.84% higher than SMN model, KERN model, and

VCTree model, respectively. Without constraints, the mean score is 30.67%, which is 10.07% and 4.17% higher than the SMN model and the KERN model. The significant improvement of mean Recall@K shows that the proposed model can alleviate the problem of uneven relationships distribution. Figure 6 shows the performance comparison of our model and KERN model for mR@50 without constraints on the scene graph generation on the VG dataset. Figure 6(a) shows the mR@50 without constraints of our method and the KERN on the SGGen task, and Figure 6(b) presents that the distribution of different relationships on the VG dataset is extremely uneven. We find that our model achieves better performance on 47 of the 50 relationships, and both the frequent relationships and the relationships with fewer samples get evident improvement from Figure 6(c) and Figure 6(d).

4.3. Ablation study

In this section, to prove the effectiveness of all modules, we analyze the impact of the sub-module on the final performance through ablation study. The core modules mainly include the statistical co-occurrence knowledge learning by spares knowledge graph (indicated by "SCK" in Table 3), relationship measurement network (RelMN), and GAT with new feature aggregation and update way (indicated by "new-GAT" in Table 3). We design some comparative experiments to test the effect of three modules with different branches on the model performance. These experiments are performed on the VG dataset, and the results are shown in Table 3. The three leftmost columns in Table 3 indicate whether the model is equipped with SCK, RelMN, new-GAT or other varieties. The purpose is to explore the impact of: (1) statistical co-occurrence knowledge. The symbol '×' in SCK means no prior statistical probabilities in our model; (2) the sparse graph. The 'R4K' means that the sparse graph is consists of random 4K edges from $n(n-1)$ object pairs; The 'K(F)' means that the sparse graph is consists of all K foreground edges ; The 'K(F)+R3K(B)' means all K foreground edges and random $3K$ background edges; The 'K(F)+T3K(B)' is adopted by our RelMN, which means all K foreground edges and Top $3K$ back-

Table 3: The results of ablation studies on our model on three different tasks.

			SGGen			SGCls			PredCls			
SCK	RelMN	new-GAT	R@20	R@50	R@100	R@20	R@50	R@100	R@20	R@50	R@100	Mean
×	✓	✓	19.9	25.7	28.9	32.2	35.9	37.0	58.2	64.4	65.5	42.90
✓	R4K	✓	17.5	21.7	24.8	27.4	30.5	32.2	50.8	58.8	60.0	38.10
✓	K(F)	✓	22.0	27.5	31.1	36.9	38.9	39.6	61.0	66.5	68.1	45.28
✓	K(F)+R3K(B)	✓	23.0	28.3	32.3	37.2	40.0	40.7	62.4	67.6	69.0	46.32
✓	K(F)+T3K(B)	✓	**23.2**	**28.8**	**32.9**	**37.5**	**40.2**	**41.1**	**62.6**	**67.7**	**69.3**	**46.67**
✓	✓	GCN	21.9	26.7	30.0	34.5	37.5	38.3	59.4	65.8	66.7	44.17
✓	✓	GAT	22.6	27.5	31.4	35.8	38.0	39.0	60.7	67.0	67.5	45.07
✓	✓	new-GAT	**23.2**	**28.8**	**32.9**	**37.5**	**40.2**	**41.1**	**62.6**	**67.7**	**69.3**	**46.67**

ground edges with high foreground scores; (3) graphical message passing. The 'GCN' means the most common graph convolution network model for message passing; The 'GAT' means the general GAT, in which the node features are used for object classification and two nodes features are concatenated for relationship recognition; As described in Section 3.4, The 'new-GAT' is equipped by our model.

The results in Table 3 shows that the model learns the statistical co-occurrence knowledge for object categories and categories-relationships through the spares knowledge graph, which can significantly improve the performance of the scene graph generation, and the mean score increase from 42.90% to 46.67%. The RelMN effectively generates meaningful candidate connections, making the model more accurate and efficient in the features learning. Under the influence of the RelMN, our model increases the mean score by 8.57% compared with randomly selecting object pairs. Specifically, the background edges with high foreground score can enhance the robustness and reduce the risk of filtering out ground truth relationships. Its performance is 1.39% higher than the sparse graph without background edge and is 0.35% higher than random background edges. The GAT with a new feature aggregation and update way gets 2.50% higher than GCN and get 1.60% higher than the general GAT, which shows that the model integrates the context information on the nodes and edges and improves the learning of relationship features in the message passing.

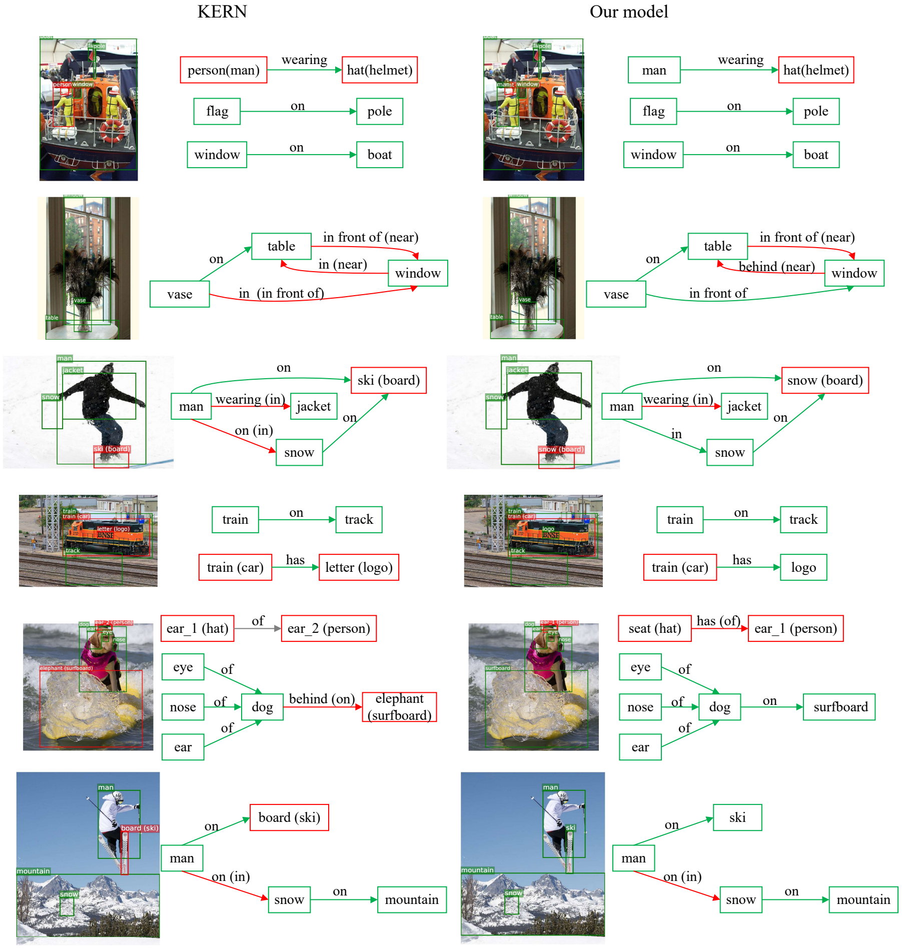


Figure 7: The qualitative results of our model and KERN method on SGCls task. Green boxes are the correctly predicted object categories, red boxes are the wrongly predicted object categories and the correct categories are in brackets. Green edges are true predictions for relationships, red edges are false predictions for relationships and the proper relationships are in brackets, and gray edges are relationships that are not predicted.

4.4. Qualitative results

In this section, to more intuitively show that the model can better identify objects and relationships, and reduce computational complexity, we visualize several SGCls results from KERN and our model in Figure 7, and display some sparse graph examples and the execution time in Figure 8. Figure 7 presents the

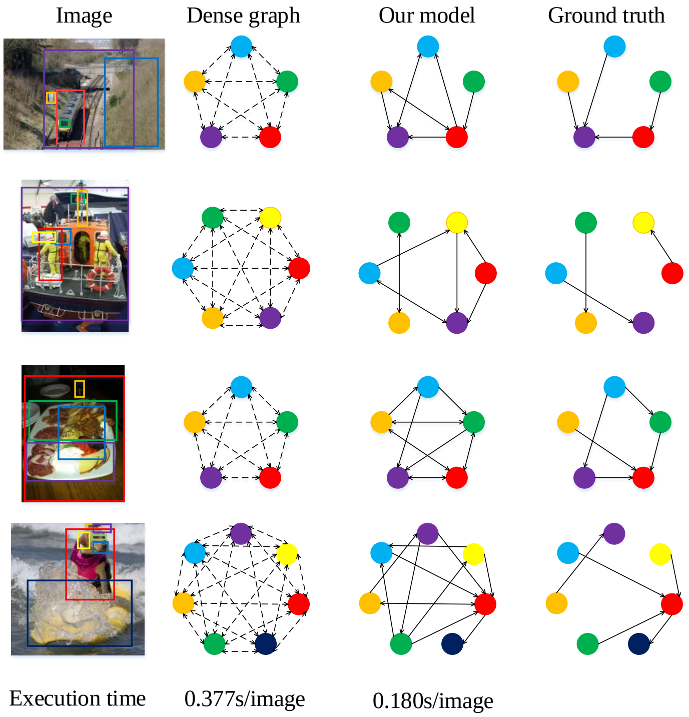


Figure 8: The visualization results for our sparse graph structures and the execution time.

qualitative results of our model and KERN method on the SGCls task. These figures show that our model can better judge meaningful relationships and identify them, such as “vase-in front of-window”. In VG dataset, an image contains 11.5 objects and 6.2 relationships on average, which indicates the dense graph need compute features and classify them on about 120 edges for an image. Ideally, our model constructs the sparse graph with about 25 edges, which reduces the computational complexity. Figure 8 shows the visualization results for our sparse graph structures and the execution time in training on SGCls task. We can find that the execution time of sparse graph is 0.180s per image, with a relative reduction of 52.3% compared with the dense graph. The sparse graph in our model basically covers almost meaningful relationships, and also includes some background edges which contains numerous unannotated relationships and filters out unnecessary search candidates.

5. Conclusion

In this paper, we propose a knowledge learning framework of constructing sparse graph and graphical message passing for scene graph generation. First, the relationship measurement network (RelMN) classifies the relationships into foreground and background, and automatically constructs the sparse graph in the scene. Based on the graph structure, the graph attention network with a new feature aggregation and update way is used to update the node features and edge features and explores context clues. Then our model learns the statistical co-occurrence knowledge of object categories and categories-relationships on the structured knowledge model. Finally, we have conducted experiments on the most widely used Visual Genome benchmark to prove the superiority of our method. In the ablation study, we analyzed each component of the model in detail. To dig out fine-grained semantic relationships between object pairs and prune dense graphs more reasonably for high-level vision applications is an interesting future work. Besides, with the fine-grained semantic relationships, the scene graph can infer better scene structures for down-stream tasks, such as search by image and VQA.

Acknowledgment

This work was partially supported by the National Natural Science Foundation of China [Grant No. 61671459, 61906208, and 61906201] and National Natural Science Foundation for Young Scholars of Hunan Province [Grant No. 2020JJ5669].

References

[1] J. Johnson, R. Krishna, M. Stark, L.-J. Li, D. Shamma, M. Bernstein, L. Fei-Fei, Image retrieval using scene graphs, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 3668–3678.

[2] W. Ma, Y. Wu, F. Cen, G. Wang, Mdfn: Multi-scale deep feature learning network for object detection, Pattern Recognition 100 (2020) 107149.

[3] J. Peng, G. Estrada, M. Pedersoli, C. Desrosiers, Deep co-training for semi-supervised image segmentation, Pattern Recognition 107 (2020) 107269.

[4] J. Zhang, M. Elhoseiny, S. Cohen, W. Chang, A. Elgammal, Relationship proposal networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 5678–5686.

[5] K. Tang, Y. Niu, J. Huang, J. Shi, H. Zhang, Unbiased scene graph generation from biased training, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2020, pp. 3716–3725.

[6] D. Xu, Y. Zhu, C. B. Choy, L. Fei-Fei, Scene graph generation by iterative message passing, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 5410–5419.

[7] J. Luo, J. Zhao, B. Wen, Y. Zhang, Explaining the semantics capturing capability of scene graph generation models, Pattern Recognition 110 (2021) 107427.

[8] J. Johnson, A. Gupta, L. Fei-Fei, Image generation from scene graphs, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 1219–1228.

[9] A. Prest, V. Ferrari, C. Schmid, Explicit modeling of human-object interactions in realistic videos, IEEE transactions on pattern analysis and machine intelligence 35 (4) (2012) 835–848.

[10] X. Yin, V. Ordonez, Obj2text: Generating visually descriptive language from object layouts, in: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, 2017.

[11] D. A. Hudson, C. D. Manning, Gqa: A new dataset for real-world visual reasoning and compositional question answering, in: Proceedings of the

IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 6700–6709.

[12] X. Yang, K. Tang, H. Zhang, J. Cai, Auto-encoding scene graphs for image captioning, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 10685–10694.

[13] S. Chen, Q. Jin, P. Wang, Q. Wu, Say as you wish: Fine-grained control of image caption generation with abstract scene graphs, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 9962–9971.

[14] C. Lu, R. Krishna, M. Bernstein, L. Fei-Fei, Visual relationship detection with language priors, in: Proceedings of the European Conference on Computer Vision, Springer, 2016, pp. 852–869.

[15] B. Dai, Y. Zhang, D. Lin, Detecting visual relationships with deep relational networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 3076–3086.

[16] K. Tang, H. Zhang, B. Wu, W. Luo, W. Liu, Learning to compose dynamic tree structures for visual contexts, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 6619–6628.

[17] J. Jiang, Z. He, S. Zhang, X. Zhao, J. Tan, Learning to transfer focus of graph neural network for scene graph parsing, Pattern Recognition 112 (2021) 107707.

[18] R. Zellers, M. Yatskar, S. Thomson, Y. Choi, Neural motifs: Scene graph parsing with global context, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 5831–5840.

[19] J. Yang, J. Lu, S. Lee, D. Batra, D. Parikh, Graph r-cnn for scene graph generation, in: Proceedings of the European Conference on Computer Vision, 2018, pp. 670–685.

[20] T. Chen, W. Yu, R. Chen, L. Lin, Knowledge-embedded routing network for scene graph generation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 6163–6171.

[21] S. Ren, K. He, R. Girshick, J. Sun, Faster r-cnn: Towards real-time object detection with region proposal networks, in: Advances in Neural Information Processing Systems, 2015, pp. 91–99.

[22] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma, et al., Visual genome: Connecting language and vision using crowdsourced dense image annotations, International Journal of Computer Vision 123 (1) (2017) 32–73.

[23] X. Lin, C. Ding, J. Zeng, D. Tao, Gps-net: Graph property sensing network for scene graph generation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2020, pp. 3746–3753.

[24] W. Che, X. Fan, R. Xiong, D. Zhao, Visual relationship embedding network for image paragraph generation, IEEE Transactions on Multimedia.

[25] Y. Zhan, J. Yu, T. Yu, D. Tao, Multi-task compositional network for visual relationship detection, International Journal of Computer Vision 128 (8) (2020) 2146–2165.

[26] C. Desai, D. Ramanan, Detecting actions, poses, and objects with relational phraselets, in: Proceedings of the European Conference on Computer Vision, Springer, 2012, pp. 158–172.

[27] M. A. Sadeghi, A. Farhadi, Recognition using visual phrases, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2011, pp. 1745–1752.

[28] Y. Li, W. Ouyang, X. Wang, X. Tang, Vip-cnn: Visual phrase guided convolutional neural network, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 1347–1356.

[29] H. Zhang, Z. Kyaw, S.-F. Chang, T.-S. Chua, Visual translation embedding network for visual relation detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 5532–5540.

[30] J. Wang, W. Wang, L. Wang, Z. Wang, D. D. Feng, T. Tan, Learning visual relationship and context-aware attention for image captioning, Pattern Recognition 98 (2020) 107075.

[31] X. Yang, H. Zhang, J. Cai, Shuffle-then-assemble: Learning object-agnostic visual relationship features, in: Proceedings of the European Conference on Computer Vision, 2018, pp. 36–52.

[32] R. Yu, A. Li, V. I. Morariu, L. S. Davis, Visual relationship detection with internal and external linguistic knowledge distillation, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 1974–1982.

[33] G. Yin, L. Sheng, B. Liu, N. Yu, X. Wang, J. Shao, C. Change Loy, Zoomnet: Mining deep feature interactions for visual relationship recognition, in: Proceedings of the European Conference on Computer Vision, 2018, pp. 322–338.

[34] A. Oliva, A. Torralba, The role of context in object recognition, Trends in Cognitive Sciences 11 (12) (2007) 520–527.

[35] D. Parikh, C. L. Zitnick, T. Chen, From appearance to context-based recognition: Dense labeling in small images, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2008, pp. 1–8.

[36] Y. Li, W. Ouyang, B. Zhou, K. Wang, X. Wang, Scene graph generation from objects, phrases and region captions, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 1261–1270.

[37] X. Liang, L. Lee, E. P. Xing, Deep variation-structured reinforcement learning for visual relationship and attribute detection, in: Proceedings of the

IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 848–857.

[38] C. Galleguillos, A. Rabinovich, S. Belongie, Object categorization using co-occurrence, location and appearance, in: Proceedings of the IEEE International Conference on Computer Vision, 2008, pp. 1–8.

[39] L. Ladicky, C. Russell, P. Kohli, P. H. Torr, Graph cut based inference with co-occurrence statistics, in: Proceedings of the European Conference on Computer Vision, Springer, 2010, pp. 239–253.

[40] C. Desai, D. Ramanan, C. C. Fowlkes, Discriminative models for multi-class object layout, International Journal of Computer Vision 95 (1) (2011) 1–12.

[41] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, Y. Bengio, Graph attention networks, in: ICLR, 2018.

[42] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, Computer Science.

[43] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, Imagenet: A large-scale hierarchical image database, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 248–255.

[44] J. Redmon, A. Farhadi, Yolo9000: better, faster, stronger, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 7263–7271.

[45] A. Newell, J. Deng, Pixels to graphs by associative embedding, in: Advances in Neural Information Processing Systems, 2017, pp. 2171–2180.