Knowledge-Embedded Routing Network for Scene Graph Generation

Tianshui Chen†
Sun Yat-Sen University
DarkMatter AI Research
tianshuichen@gmail.com

Weihao Yu†
Sun Yat-Sen University
weihaoyu6@gmail.com

Riquan Chen
Sun Yat-Sen University
sysucrq@gmail.com

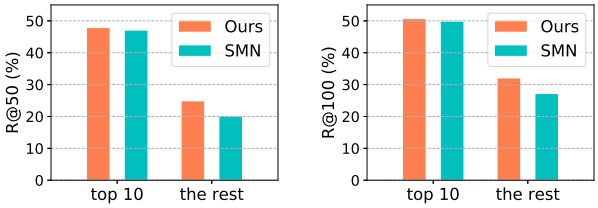
Liang Lin*
Sun Yat-Sen University
DarkMatter AI Research
linliang@ieee.org

Abstract

To understand a scene in depth not only involves locating/recognizing individual objects, but also requires to infer the relationships and interactions among them. However, since the distribution of real-world relationships is seriously unbalanced, existing methods perform quite poorly for the less frequent relationships. In this work, we find that the statistical correlations between object pairs and their relationships can effectively regularize semantic space and make prediction less ambiguous, and thus well address the unbalanced distribution issue. To achieve this, we incorporate these statistical correlations into deep neural networks to facilitate scene graph generation by developing a Knowledge-Embedded Routing Network. More specifically, we show that the statistical correlations between objects appearing in images and their relationships, can be explicitly represented by a structured knowledge graph, and a routing mechanism is learned to propagate messages through the graph to explore their interactions. Extensive experiments on the large-scale Visual Genome dataset demonstrate the superiority of the proposed method over current state-of-the-art competitors.

1. Introduction

Scene graph [13] is a structured representation of image content that not only encodes semantic and spatial information of individual objects in the scene but also represents the relationship between each pair of objects. In recent years, inferring such graph has drawn increasing attention [30, 6] as it provides a deeper understanding for the image and thus facilitates various vision tasks ranging from fundamental recognition and detection [20, 8] to high-level tasks [34, 32].



(a) (b)

Figure 1. (a) Recall@50 and (b) Recall@100 of our proposed method and the SMN [33] on the scene graph classification task on the Visual Genome dataset [14]. Both models are trained on the whole training set and evaluated on the two subsets, respectively. Note that SMN is the previous best-performing method.

Existing methods for scene graph generation rely on the target object regions [19, 6] or further introduce contextual cues [30, 33] to aid recognition. Generally, these methods require large amounts of annotated samples for model optimization. However, the distribution of real-world relationships is seriously uneven, leading to relatively poor performance for the relationships with limited training samples. Take the Visual Genome dataset [14] as an example, we evaluate the performance on samples of top 10 most frequent relationships (namely "top 10" subset) and that on samples of the rest less frequent relationships (namely "the rest" subset), respectively. As shown in Figure 1, current best-performing method (i.e, SMN [33]) can achieve com-

*Tianshui Chen and Weihao Yu share first-authorship. Corresponding author is Liang Lin. This work was supported in part by State Key Development Program under Grant 2016YFB1001004, in part by the National Key Research and Development Program of China under Grant No. 2018YFC0830103, in part by National High Level Talents Special Support Plan (Ten Thousand Talents Program), and in part by National Natural Science Foundation of China (NSFC) under Grant No. 61622214, 61836012, and 61702565.

petitive performance if it has sufficient training samples, but its performance suffers from a severe drop otherwise.

Objects in visual scene commonly have strongly structured regularities [33]. For example, people tend to wear clothes, while cars are inclined to have wheels. The statistical analysis [33] on the Visual Genome dataset [14] revealed that a baseline method, which directly predicts the most frequent relationship of object pairs with given labels, outperforms most existing state-of-the-art methods [23, 30]. Therefore, modeling these statistical correlations between object pairs and relationships can effectively regularize the semantic prediction space, and thus address the uneven distribution issue. On the other hand, the interplay of relationships and objects in the scene also plays a significant role in scene graph generation [30].

We show that the statistical correlations between object pairs and their relationships can be explicitly represented by a structured knowledge graph, and the interplay between these two factors can be captured by propagating node messages through the graph. Similarly, contextual cues can also be represented and explored by another graph with proper message propagation. In this work, we introduce a novel Knowledge-Embedded Routing Network (KERN), which captures the interplay of target objects and their relationships under the explicit guidance of prior statistical knowledge and automatically mines contextual cues to facilitate scene graph generation. Although previous studies [6, 33] have also taken notice of the statistical knowledge, they merely implicitly mine this information by iterative message propagation between relationships and objects [30] or by encoding the global context of objects and relationships [33]. Instead, our model formally represents this statistical knowledge in the form of a structured graph and incorporates the graph into deep propagation network as extra guidance. In this way, it can effectively regularize the distribution of possible relationships of object pairs and thus make prediction less ambiguous. As shown in Figure 1, compared with current best-performing method (i.e., SMN [33]), our model achieves slight improvement for the relationships with sufficient samples, and the improvement is much more evident for the relationships with limited samples.

Our model builds on the Faster RCNN detector [25] to generate a set of object regions. Then, a graph that correlates these regions according to the statistical object co-occurrences is first built, and a propagation network is employed to propagate node messages through the graph to learn contextualized feature representation to predict the class label regarding each region. For each object pair with predicted labels, we build a graph, in which nodes represent the objects and relationships, and edges represent the statistical co-occurrence probabilities between the given object pair and all relationships. Further, we adopt another propagation network to explore the interplay between the relationships and corresponding objects to predict their relationship. This process is performed for all object pairs, and the whole scene graph is generated.

On the other hand, existing works utilize the recall@K (short as R@K) [19] as the evaluation metric. However, this metric is easily dominated by the performance of the relationships with a large proportion of samples. As the distribution of different relationships is severely uneven, if one method performs well on several most frequent relationships, it can achieve a high R@K score. Thus, it can not well measure the performance of all relationships. To address this issue, we further propose a mean recall@K (short as mR@K) as a complimentary evaluation metric. It first computes the R@K for samples of each relationship and then averages over all relationships to obtain mR@K. Compared with R@K, mR@K can give a more comprehensive performance evaluation for all relationships.

To the best of our knowledge, this work is the first to explicitly unify the statistical knowledge with the deep architecture to facilitate scene graph generation. Compared with existing methods, our model incorporates this knowledge to regularize the semantic space of relationship prediction and thus improves the performance of scene graph generation. We conduct experiments on the most widely used and challenging Visual Genome dataset [14], and demonstrate our model can achieve best R@K performance than existing leading competitors. Notably, by explicitly regularizing the semantic space of relationship prediction, our model can well address the issue of uneven distribution of real-world relationships and achieves much more obvious improvement on the mR@K metric. For example, our model improves the mR@50 and mR@100 from 15.4% and 20.6% to 19.8% and 26.2% on the scene graph classification task, with relative improvements of 28.6% and 27.2%, respectively. Code and trained models are available at `https://github.com/HCPLab-SYSU/KERN`.

2. Related Work

2.1. Visual relationship detection

Visual relationship detection involves detecting semantic objects that occur in the images and inferring the relationship between each object pair (i.e., a subject and an object). Over the past decade, a series of works were dedicated to recognizing spatial relationships [9, 11, 5] like "above", "below", "inside", and "around", and to exploring using these relationships to improve various vision tasks such as object recognition [9], detection [8], and segmentation [11]. Some other works also attempted to learn human-object interactions [31, 1], in which the subject was a person.

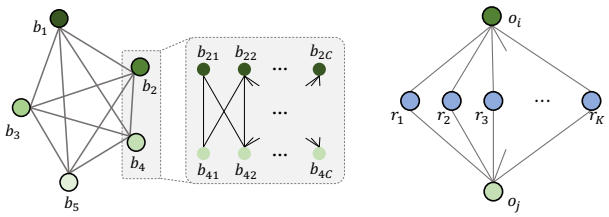
Latterly, lots of attention [19, 30, 6, 16, 23, 33] was drawn to the visual relationship detection task under a more

general and practical setting, where the subject and object can be any objects in the scene and their relationships cover a wide range of relationship types including spatial (e.g., above, below), actions (e.g., ride, wear), affiliations (e.g., part of), etc. As a pioneer work, Lu et al. [19] trained visual models of subject, relationship, and object individually to tackle the problem of the long-tail distribution of relationship triplets and leveraged language prior from semantic word embedding to further improve the predicted performance. Xu et al. [30] introduced an end-to-end model that learned to iteratively refine relationship and object prediction via message passing based on the RNNs [21]. Li et al. [16] formulated a multi-task framework to explore semantic associations over three tasks of object detection, scene graph generation, and image caption generation, and found that jointly learning the three tasks could bring about mutual improvements. More recently, Dai et al. [6] designed a deep relational network that exploited both spatial configuration and statistical dependency to resolve the ambiguities during relationship recognition. Zeller et al. [33] presented an analysis of statistical co-occurrences between relationships and object pairs on the Visual Genome dataset [14] and came to a conclusion that these statistical co-occurrences provided strong regularization for relationship prediction. They encoded the global context of objects and relationships by LSTM sequential architectures [12] to facilitate scene graph parsing.

The works [6, 33] also took notice of the statistical co-occurrences between object pair and their relationship, but they devised deep models to implicitly mine this information via message passing. Different from these works, our model formally represents this information and explicitly incorporates them into graph propagation network to help scene graph generation.

2.2. Knowledge representation

It has been extensively studied to incorporate prior knowledge to aid numerous vision tasks [20, 8, 15, 7, 2, 18]. For example, Marino et al. [20] constructed a knowledge graph based on the WordNet [22] and the Visual Genome dataset [14], and learned the representation of this graph to enhance image feature representation to promote multi-label recognition. Lee et al. [15] further extended this method to multi-label zero-shot learning. Some works also utilized the knowledge graph as extra constraints for model training. Fang et al. [8] incorporated semantic consistency into object detection systems with the constraint that more semantically consistent concepts were more likely to occur in an image. Deng et al. [7] introduced semantic relations including mutual exclusion, overlap, and subsumption, as constraints in the loss function to train the classifiers. These methods learned graph representation for feature enhancement or use graph as extra constraints on the loss functions.



(a) (b)

Figure 2. (a) A graph correlating the detected regions appearing in an image; (b) A graph correlating given object pair o_i and o_j with all the relationships.

Differently, our model introduces the graph that correlates target object pair and their possible relationships to explicitly regularize the semantic space of relationship prediction, and thus addresses the uneven distribution issue.

3. Proposed Model

A scene graph is a structured representation of content in an image. It consists of the class labels and locations of individual objects and the relationship between each object pair, which can be defined as a 3-tuple set $\mathcal{G} = \{B, O, R\}$:

- $B = \{b_1, b_2, \ldots, b_n\}$ is the region candidate set, with element $b_i \in \mathbb{R}^4$ denoting the bounding box of the i-th region.
- $O = \{o_1, o_2, \ldots, o_n\}$ is the object set, with element $o_i \in \mathbb{N}$ denoting the corresponding class label regarding region b_i.
- $R = \{r_{1\to2}, r_{1\to3}, \ldots, r_{n\to n-1}\}$ is the corresponding relationship triplet set, where $r_{i\to j}$ is a triplet of a subject $(b_i, o_i) \in B \times O$, an object $(b_j, o_j) \in B \times O$, and a relationship label $x_{i\to j} \in \mathcal{R}$.

$\mathcal{R}$ is the set of all relationships including *no-relationship* that indicates no relationship between the given object pair.

Given an image I, we decompose the probability distribution of the scene graph $p(\mathcal{G}|I)$ into three components similar to [33]:

$$p(\mathcal{G}|I) = p(B|I)p(O|B, I)p(R|O, B, I). \quad (1)$$

In this equation, the bounding box component $p(B|I)$ generates a set of candidate regions that cover most of the key objects directly from the input image. Similar to previous scene graph works [6, 33], this component is implemented by the widely used Faster RCNN detector [25]. The object component $p(O|B, I)$ then predicts the class label regarding each detected region. Here, we construct a graph that correlates the detected regions based on the statistical object co-occurrence information (see Figure 2(a)). Then, our model adopts a graph neural network [27, 17] to propagate

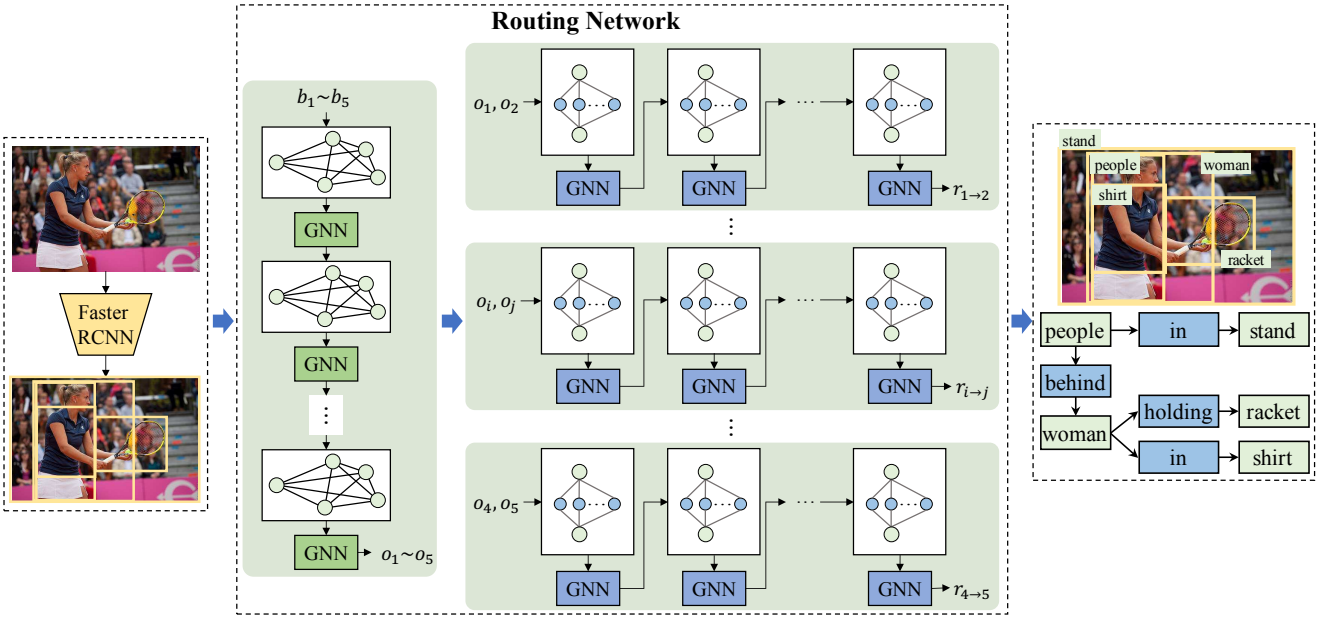


Figure 3. An overall pipeline of the knowledge-embedded routing network. Given an image, we first adopt the Faster RCNN to detect a set of regions. Then, a graph is built to correlate the regions, and a graph neural network is employed to learn contextualized representation to predict the class label for each region. For each object pair with predicted labels, we build another graph to correlate the given object pair with all the possible relationships and employ a graph neural network to infer their relationship. The process is repeated for all object pairs and the scene graph is generated.

messages through the graph to learn contextualized representation for each region and achieves better label prediction under the constraint of statistical information of object co-occurrences. Conditioned on the predicted labels, the relationship component $p(R|O, B, I)$ infers the relationship of each object pair and finally generates the whole scene graph. For each object pair with predicted labels, we construct a graph, in which nodes refer to the objects and relationships, and edges represent the statistical co-occurrences between the corresponding object pair and all the relationships (see Figure 2(b)). Similarly, another graph neural network is learned to explore the interplay between relationships and objects, and finally, the features from all nodes are aggregated to predict the relationship. Our model performs this process for all object pairs and generates the whole scene graph. Figure 3 illustrates an overall pipeline of the proposed model.

3.1. Bounding box localization

Given an image, the model first obtains a set of candidate regions. In this work, we utilize the Faster RCNN [25] to automatically generate the region set $B = \{b_1, b_2, \ldots, b_n\}$ directly from input image I. For each region, besides a bounding box $b_i \in \mathbb{R}^4$ denoting its position, our model also extracts a feature vector $\mathbf{f}_i$ using the ROI pooling layer [10]. These feature vectors are then fed into the propagation networks for subsequent inference.

3.2. Knowledge-embedded routing network

Object. Statistical information of object co-occurrence is a crucial cue to correlate objects in an image and regularizes object label prediction. In this work, we build a graph to associate the regions detected in the image according to these statistical correlations and employ a graph neural network to propagate messages through the graph that can learn contextualized representation to predict the class label regarding each region.

To this end, we first count the statistical co-occurrence probabilities of objects from different categories on the training set of the target dataset (e.g., Visual Genome [14]). More specifically, for two categories of c and c', we count the probability $m_{cc'}$ of the existence of object belonging to category c in the presence of object belonging to category c'. We count these co-occurrence probabilities for all category pair and obtain a matrix $M_c \in \mathbb{R}^{C\times C}$, where C is the number of object categories. We then correlate the regions from B based on the matrix M_c. Given two regions of b_i and b_j, we duplicate b_i C times to obtain C nodes $\{b_{i1}, b_{i2}, \ldots, b_{iC}\}$, with node b_{ic} denoting the correlation of region b_i with category c. The same process is performed for b_j. Intuitively, $m_{cc'}$ can be used to correlate node $b_{jc'}$ to b_{ic}, and thus M_c can be used to correlate nodes of region b_i and nodes of b_j. In this way, we can correlate all regions and construct the graph.

Inspired by the Graph Gated Neural Networks [17, 3,

29], we adopt a gated recurrent update mechanism to iterative propagate node messages through the graph. Specifically, at timestep t, each node b_{ic} has a hidden state $\mathbf{h}_{ic}^t$. As each node corresponds to a specific region, we use the feature vector of this region to initialize the hidden state at $t = 0$, which can be expressed as

$$\mathbf{h}_{ic}^0 = \varphi_o(\mathbf{f}_i), \tag{2}$$

where φ_o is a transformation that maps $\mathbf{f}_i$ to a feature vector of low dimension, and it is implemented by a fully connected layer. At each timestep t, each node aggregates messages from its neighbors according to the graph structure, formulated as

$$\mathbf{a}_{ic}^t = \left[\sum_{j=1,j\neq i}^{n} \sum_{c'=1}^{C} m_{c'c} \mathbf{h}_{jc'}^{t-1}, \sum_{j=1,j\neq i}^{n} \sum_{c'=1}^{C} m_{cc'} \mathbf{h}_{jc'}^{t-1} \right]. \tag{3}$$

Then, the model take a_{ic}^t and its previous hidden state as input to update its hidden state by a gated mechanism similar to the Gated Recurrent Unit [4, 17]

$$\begin{aligned} \mathbf{z}_{ic}^t &= \sigma(\mathbf{W}_o^z \mathbf{a}_{ic}^t + \mathbf{U}_o^z \mathbf{h}_{ic}^{t-1}) \\ \mathbf{r}_{ic}^t &= \sigma(\mathbf{W}_o^r \mathbf{a}_{ic}^t + \mathbf{U}_o^r \mathbf{h}_{ic}^{t-1}) \\ \widetilde{\mathbf{h}_{ic}^t} &= \tanh\left(\mathbf{W}_o \mathbf{a}_{ic}^t + \mathbf{U}_o(\mathbf{r}_{ic}^t \odot \mathbf{h}_{ic}^{t-1})\right) \\ \mathbf{h}_{ic}^t &= (1 - \mathbf{z}_{ic}^t) \odot \mathbf{h}_{ic}^{t-1} + \mathbf{z}_{ic}^t \odot \widetilde{\mathbf{h}_{ic}^t} \end{aligned} \tag{4}$$

In this way, each node can aggregate messages from its neighbors and meanwhile transfer its message to its neighbors, enabling interactions among all nodes in the graph. After T_o steps, the node messages have been propagated through the graph and we obtain the final hidden state for each region i, i.e., $\{\mathbf{h}_{i1}^{T_o}, \mathbf{h}_{i2}^{T_o}, \ldots, \mathbf{h}_{iC}^{T_o}\}$. We use an output network that takes the initial hidden state and final hidden state as input to compute the output feature for each node

$$\mathbf{f}_{ic}^o = o_o(\mathbf{h}_{ic}^0, \mathbf{h}_{ic}^{T_o}), \tag{5}$$

where $o_o(\cdot)$ is implemented by a fully connected layer. Finally, for each region, we aggregate all correlated output feature vectors to predict its class label

$$\mathbf{o}_i = \phi_o(\mathbf{f}_{i1}^o, \mathbf{f}_{i2}^o, \ldots, \mathbf{h}_{iC}^o) \tag{6}$$

The predicted class label $o_i = \text{argmax}(\mathbf{o}_i)$ are then used for relationship inference.

Relationship. Given the categories of object pair, the probability distribution of their relationships is highly skewed. For example, given a subject "man" and an object "horse", their relationship is likely to be "riding". Here, we represent the correlations of object pair and their relationships in the form of a structured graph and adopt another graph neural network to explore the interplay of these two factors to infer the relationship.

To this, we also count the statistical co-occurrence probability on the training part of the target dataset to obtain these correlations. Concretely, we count the probabilities of all possible relationships given a subject of the category c and an object of the category c', which are denoted as $\{m_{cc'1}, m_{cc'2}, \ldots, m_{cc'K}\}$. Here, K is the relationship number. For a subject o_i and an object o_j taken from the object set O, we construct a graph with a subject node, an object node, and K relationship nodes. We use $m_{o_i o_j k}$ to denote the correlations between o_i and relationship node k as well as between o_j and relationship node k. In this way, a graph with statistic co-occurrences embedded is built.

Our model learns to explore the node interaction using the identical graph gated recurrent update mechanism [17]. Similarly, each node $v \in V = \{o_i, o_j, 1, 2, \ldots, K\}$ has a hidden state $\mathbf{h}_v^t$ at timestep t. At timestep $t = 0$, we initialize the object nodes with the feature vectors of corresponding regions and the relationship nodes with the feature vector from the union region of the two objects together with their spatial information

$$\mathbf{h}_v^0 = \begin{cases} \varphi_{o'}(\mathbf{f}_i) & \text{if } v \text{ is the object node } o_i \\ \varphi_r(\mathbf{f}_{ij}) & \text{if } v \text{ is a relationship node} \end{cases}, \tag{7}$$

where $\varphi_{o'}$ and φ_r are two transformations, and both are implemented by a fully-connected layer, respectively. $\mathbf{f}_{ij}$ is a feature vector that encodes the visual feature of the union region of b_i and b_j as well as the spatial information following [33]. At each timestep t, the relationship nodes aggregate messages from the object nodes while object nodes aggregate messages from the relationship nodes

$$\mathbf{a}_v^t = \begin{cases} \sum_{k=1}^{K} m_{o_i o_j k} \mathbf{h}_k^{t-1} & \text{if } v \text{ is a object node} \\ m_{o_i o_j k}(\mathbf{h}_{o_i}^{t-1} + \mathbf{h}_{o_j}^{t-1}) & \text{if } v \text{ is the relationship node } k \end{cases}. \tag{8}$$

Then, the model incorporates these aggregated features with the previous hidden states to update the hidden state for each node using the gated mechanism as Eq. 4. The model repeats the iterations T_r times and generates the final hidden state of each node, i.e., $\{\mathbf{h}_{o_i}^{T_r}, \mathbf{h}_{o_j}^{T_r}, \mathbf{h}_1^{T_r}, \ldots, \mathbf{h}_K^{T_r}, \}$. Similar to [17], our model use an output sub-network implemented by a fully-connected layer to compute node-level features and aggregates these features to infer the relationship

$$\begin{aligned} \mathbf{f}_v^o &= o_r([\mathbf{h}_v^{T_r}, \mathbf{h}_v^0]) \\ \mathbf{x}_{i\to j} &= \phi_r([\mathbf{f}_{o_i}^o, \mathbf{f}_{o_j}^o, \mathbf{f}_1^o, \ldots, \mathbf{f}_K^o]). \end{aligned} \tag{9}$$

ϕ_r is the relationship classifier implemented by a fully connected layer.

4. Experiments

4.1. Experiment setting

Implementation details. Similar to prior works [30, 33] for scene graph generation, we adopt the Faster RCNN detec-

	Method	SGGen		SGCls		PredCls		
		mR@50	mR@100	mR@50	mR@100	mR@50	mR@100	Mean
Constraint	IMP [30]	0.6	0.9	3.1	3.8	6.1	8.0	3.8
	IMP+ [30, 33]	3.8	4.8	5.8	6.0	9.8	10.5	6.8
	FREQ [33]	4.3	5.6	6.8	7.8	13.3	15.8	8.9
	SMN [33]	5.3	6.1	7.1	7.6	13.3	14.4	9.0
	Ours	**6.4**	**7.3**	**9.4**	**10.0**	**17.7**	**19.2**	**11.7**
Unconstraint	AE [23]	1.6	2.5	6.0	7.8	15.1	19.5	8.8
	IMP+ [30, 33]	5.4	8.0	12.1	16.9	20.3	28.9	15.3
	FREQ [33]	5.9	8.9	13.5	19.6	24.8	37.3	18.3
	SMN [33]	9.3	12.9	15.4	20.6	27.5	37.9	20.6
	Ours	**11.7**	**16.0**	**19.8**	**26.2**	**36.3**	**49.0**	**26.5**

Table 1. Comparison of the mR@50 and mR@100 in % with and without constraint on the three tasks of the VG dataset. We compute Mean mR by averaging mR@50 and mR@100 over the three tasks. As existing works do not present the mR@K metric, we utilize the released models (IMP, FREQ, SMN, AE) or train the model using the released code (IMP+) to generate the results to compute the metric.

tor [25] to generate the candidate region set. The detector utilizes VGG16-ConvNet [28] pretrained on ImageNet [26] as its backbone network as in [30, 33]. The iteration step of both GGNNs is set to 3. We follow [33] to set the input image size as 592×592, and use anchor scales and aspect ratios similar to YOLO-9000 [24]. Then, we train the detector on the target dataset using the SGD algorithm with a batch size of 18, momentum of 0.9, and weight decay of 0.0001. The learning rate is initialized as 0.001 and is divided by 10 when the mAP of the validation set plateaus. After that, we freeze the weights of all the convolution layers and train the fully-connected layers as well as the stacked graph neural networks using the Adam algorithm with a batch size of 2, and momentums of 0.9 and 0.999. In this process, we initialize the learning rate as 0.00001 and divide it by 10 when the recall of the validation set plateaus.

Dataset. We evaluate the proposed method and existing state-of-the-art competitors on the Visual Genome (VG) [14] benchmark. VG contains 108,077 images with average annotations of 38 objects and 22 relationships per image. It is a challenging and most widely used benchmark for scene graph generation. In the experiments, we follow previous works [33, 30] to use the most frequent 150 object categories and 50 relationships and use the training/test split in [30] for evaluation.

Tasks. Scene graph generation aims to predict a set of *subject-relationship-object* triplets. Following [30], we evaluate the proposed model with three task setups as below:

- Predicate classification (PredCls) predicts the relationship label of given object pair from a set of objects with ground truth annotations of class labels and bounding boxes.
- Scene graph classification (SGCls) predicts the class labels for the set of objects with ground truth bounding boxes and predicts the relationship label of each object pair.
- Scene graph generation (SGGen) simultaneously detects objects appearing in the image and predicts the relationship label of each object pair.

Evaluation metrics. All the methods are evaluated using the recall@K (short as R@K) metric that measures the fraction of the ground truth relationship triplets that appear among the top K most confident triplet predictions in an image. However, as shown in Figure 4(a), the distribution of different relationships is seriously uneven, and this metric is easily dominated by the performance of the most frequent relationships. To evaluate the performance of each relationship more comprehensively, we further propose a new metric, i.e., mean recall@K (short as mR@K). This metric computes the R@K for the samples of each relationship, respectively, and then averages R@K over all relationships to obtain mR@K.

Some previous works [30] compute R@K with the constraint that merely one relationship is obtained for a given object pair. Some other works [23] omit this constraint so that multiple relationships can be obtained, leading to higher values. In this work, we report both the R@K and mR@K with and without constraint respectively for comprehensive comparisons.

4.2. Comparison with state-of-the-art methods

VG [14] is the largest and most widely used benchmark for evaluating the scene graph generation task. In this part, we compare our proposed method with the existing state-of-the-art methods, including Visual Relationship Detection (VRD) [14], Iterative Message Passing (IMP) [30] and its improved version by using a better detector (IMP+) [30, 33], Associative Embedding (AE) [23], FREQuency baseline (FREQ) [33], and Stacked Motif Networks (SMN) [33].

We first present the mR@50 and mR@100 on three tasks on the VG dataset in Table 1. As shown, the FREQ baseline method, which directly predicts the most frequent re-

	Methods	SGGen		SGCls		PredCls		
		R@50	R@100	R@50	R@100	R@50	R@100	Mean
Constraint	VRD [19]	0.3	0.5	11.8	14.1	27.9	35.0	14.9
	IMP [30]	3.4	4.2	21.7	24.4	44.8	53.0	25.3
	IMP+ [30, 33]	20.7	24.5	34.6	35.4	59.3	61.3	39.3
	FREQ [33]	23.5	27.6	32.4	34.0	59.9	64.1	40.3
	SMN [33]	**27.2**	**30.3**	35.8	36.5	65.2	67.1	43.7
	Ours	27.1	29.8	**36.7**	**37.4**	**65.8**	**67.6**	**44.1**
No constraint	AE [23]	9.7	11.3	26.5	30.0	68.0	75.2	36.8
	IMP+ [30, 33]	22.0	27.4	43.4	47.2	75.2	83.6	49.8
	FREQ [33]	25.3	30.9	40.5	43.7	71.3	81.2	48.8
	SMN [33]	30.5	35.8	44.5	47.7	81.1	88.3	54.7
	Ours	**30.9**	**35.8**	**45.9**	**49.0**	**81.9**	**88.9**	**55.4**

Table 2. Comparison of the R@50 and R@100 in % with and without constraint on the three tasks of the VG dataset. We compute Mean R by averaging R@50 and R@100 over the three tasks.

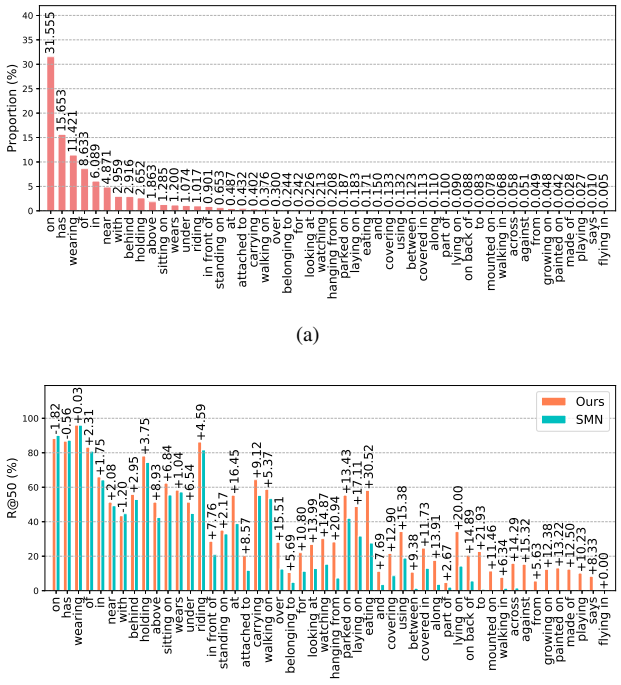
Methods	SGGen		SGCls		PredCls		
	mR@50	mR@100	mR@50	mR@100	mR@50	mR@100	Mean
Ours w/o rk & w/o ok	5.1	5.8	6.1	6.5	10.5	11.5	7.6
Ours w/o rk	5.2	5.9	6.5	6.9	11.1	12.0	7.9
Ours	**6.4**	**7.3**	**9.4**	**10.0**	**17.7**	**19.2**	**11.7**
	R@50	R@100	R@50	R@100	R@50	R@100	Mean
Ours w/o rk & w/o ok	25.2	27.9	33.9	34.8	58.7	61.0	40.3
Ours w/o rk	25.5	28.0	34.3	35.2	59.2	61.5	40.6
Ours	**27.1**	**29.8**	**36.7**	**37.4**	**65.8**	**67.6**	**44.1**

Table 3. Comparison of the mR@50, mR@100 (above) and the R@50, R@100 (below) with constraint in % of our full model, our model without relationship correlation (w/o rk), and our model without relationship correlation and object correlation (w/o rk & ok). We compute Mean mR by averaging mR@50 and mR@100 over the three tasks and mean R in the same way.

lationship of object pairs with given labels, performs better than most existing works. This comparison suggests that the statistical correlations between object pairs and their relationships play an equally or even more important role than other information like contextual cues [30]. SMN is the best-performing method among existing works, which implicitly captures these statistical correlations by encoding global context. It achieves the mean mR of 9.0% and 20.6% under the evaluation settings with and without constraint. By explicitly incorporating the statistical correlations, our method can make better use of them, leading to notable performance improvement. Specifically, it consistently outperforms existing methods on all three tasks under the two settings. For example, it obtains the mean mR of 11.7% and 26.5%, with a relative improvement of 30.0% and 28.6% compared with the previous best-performing method (i.e., SMN). Note that we use prior statistical correlations to aid scene graph generation. But these correlations are obtained merely based on the annotations of samples from the training set, and no additional supervision is introduced. Thus, the preceding comparisons are fair.

For more comprehensive comparison with existing methods, we also present the R@50 and R@100 on the three tasks on the VG dataset in Table 2. Still, our method achieves best results on these metrics. Concretely, the mean R is 44.1% and 55.4% under the settings with and without constraint, with an improvement of 0.4% and 0.7% compared with SMN.

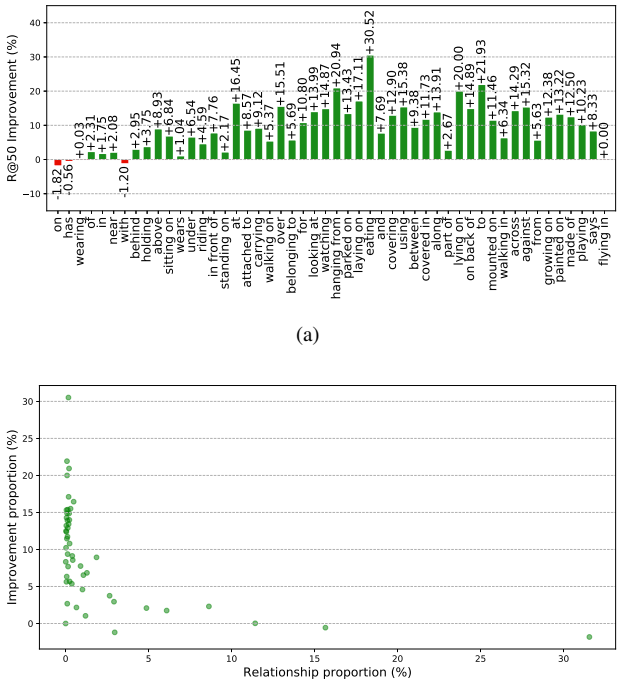
As shown in the above discussion and comparison, our method exhibits an improvement compared with existing state-of-the-art methods, both on the mR@K and R@K metrics. However, we find that the improvement on the mR@K metric is much more obvious than that on the R@K metric. Here, we give a deeper and more comprehensive analysis for this phenomenon. We first present the distribution of different relationships on the VG dataset in Figure 4(a), and the corresponding distributions on the training and test splits are basically the same to this distribution. As shown, the distribution is extremely uneven. The samples of the top 10 most frequent relationships account for almost 90% samples, while those of the rest 40 relationships merely account for about 10%. Thus, the R@K metric is dominated by the performance of these most frequent relationships. As shown in Figure 4(b), current state-of-the-art method (i.e., SMN) performs quite well for these relationships such as "on", "has"; thus it can achieve a good R@K. However, SMN performs quite poorly for the relationships that have fewer samples (e.g., "make of", "to"). The mR@K metric measures the overall performance over all relationships; thus these poor results lead to an obvi-



(a)

(b)

Figure 4. (a) The distribution of different relationships on the VG dataset. The training and test splits share similar distribution. (b) The R@50 without contraint of our method and the SMN on the predicate classification task on the VG dataset.



(a)

(b)

Figure 5. (a) The R@50 improvement of different relationships of our methods to the SMN and (b) the relation between the R@50 improvement and sample proportion on the predicate classification task on the VG dataset. The R@50 are computed without contraint.

ous drop on this metric. Different from existing methods, our model integrates prior knowledge to explicitly regularize the semantic space; thus it also performs well for these less frequent relationships. In this way, our model can well address the issue of uneven distribution of relationships.

To present a more direct comparison of the relation between the performance improvement and sample number, we further present the R@50 improvement for each relationship and sample proportion in Figure 5(a) and 5(b). As shown, our model achieves evident improvement in almost all relationships (47/50). Besides, the improvement is more obvious for the relationships with fewer samples.

4.3. Ablative study

The core of our method is the explicit incorporation of statistical correlation of object pair and their relationship. To better verify its effectiveness, we replace the statistical probabilities with uniform distribution, i.e., assigning each $m_{cc'k}$ to $\frac{1}{K}$, leaving other components unchanged. Then, we retrain the model in similar way. The experiment is conducted on the VG dataset and the results are presented in Table 3. We find that the mean mR decreases from 11.7% to 7.9% and the mean R decreases from 44.1% to 40.6%. This obvious performance drop clearly indicates incorporating statistical correlations significantly helps scene graph generation.

It is another important module that our method propagates messages through regions appearing in the image to learn contextualized representation. Similarly, we analyze its significance by replacing the statistical probabilities with a uniform distribution, and retrain the model on the VG dataset. As shown in Table 3, both the mean mR and mean R suffer from 0.3% drop.

5. Conclusion

The prior knowledge of statistical correlations between object pair and their relationship can help regularize the semantic space of relationship prediction given target object pair, and thus effectively address the issue of the uneven distribution over different relationships. In this work, we show these correlations can be explicitly represented by a knowledge graph, in which a routing mechanism is learned to propagate node messages through the graph under the explicit guidance of the structured knowledge. We conduct experiments on the most widely used Visual Genome benchmark and demonstrate the superiority of the proposed method.

References

[1] Y.-W. Chao, Y. Liu, X. Liu, H. Zeng, and J. Deng. Learning to detect human-object interactions. *arXiv preprint arXiv:1702.05448*, 2017.

[2] T. Chen, R. Chen, L. Nie, X. Luo, X. Liu, and L. Lin. Neural task planning with and-or graph representations. *TMM*, 2018.

[3] T. Chen, L. Lin, R. Chen, Y. Wu, and X. Luo. Knowledge-embedded representation learning for fine-grained image recognition. In *IJCAI*, pages 627–634, 2018.

[4] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. In *EMNLP*, 2014.

[5] W. Choi, Y.-W. Chao, C. Pantofaru, and S. Savarese. Understanding indoor scenes using 3d geometric phrases. In *CVPR*, pages 33–40, 2013.

[6] B. Dai, Y. Zhang, and D. Lin. Detecting visual relationships with deep relational networks. In *CVPR*, pages 3298–3308, 2017.

[7] J. Deng, N. Ding, Y. Jia, A. Frome, K. Murphy, S. Bengio, Y. Li, H. Neven, and H. Adam. Large-scale object classification using label relation graphs. In *ECCV*, pages 48–64, 2014.

[8] Y. Fang, K. Kuan, J. Lin, C. Tan, and V. Chandrasekhar. Object detection meets knowledge graphs. In *IJCAI*, pages 1661–1667, 2017.

[9] C. Galleguillos, A. Rabinovich, and S. Belongie. Object categorization using co-occurrence, location and appearance. In *CVPR*, pages 1–8, 2008.

[10] R. Girshick. Fast r-cnn. In *ICCV*, pages 1440–1448, 2015.

[11] S. Gould, J. Rodgers, D. Cohen, G. Elidan, and D. Koller. Multi-class segmentation with relative location prior. *IJCV*, 80(3):300–316, 2008.

[12] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[13] J. Johnson, R. Krishna, M. Stark, L.-J. Li, D. Shamma, M. Bernstein, and L. Fei-Fei. Image retrieval using scene graphs. In *CVPR*, pages 3668–3678, 2015.

[14] 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. *IJCV*, 123(1):32–73, 2017.

[15] C.-W. Lee, W. Fang, C.-K. Yeh, and Y.-C. F. Wang. Multi-label zero-shot learning with structured knowledge graphs. *arXiv preprint arXiv:1711.06526*, 2017.

[16] Y. Li, W. Ouyang, B. Zhou, K. Wang, and X. Wang. Scene graph generation from objects, phrases and region captions. In *CVPR*, pages 1261–1270, 2017.

[17] Y. Li, D. Tarlow, M. Brockschmidt, and R. Zemel. Gated graph sequence neural networks. In *ICLR*, 2016.

[18] L. Lin, L. Huang, T. Chen, Y. Gan, and H. Cheng. Knowledge-guided recurrent neural network learning for task-oriented action prediction. In *ICME*, pages 625–630, 2017.

[19] C. Lu, R. Krishna, M. Bernstein, and L. Fei-Fei. Visual relationship detection with language priors. In *ECCV*, pages 852–869, 2016.

[20] K. Marino, R. Salakhutdinov, and A. Gupta. The more you know: Using knowledge graphs for image classification. In *CVPR*, pages 2673–2681, 2017.

[21] T. Mikolov, M. Karafiát, L. Burget, J. Černockỳ, and S. Khudanpur. Recurrent neural network based language model. In *Eleventh Annual Conference of the International Speech Communication Association*, 2010.

[22] G. A. Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.

[23] A. Newell and J. Deng. Pixels to graphs by associative embedding. In *NIPS*, pages 2168–2177, 2017.

[24] J. Redmon and A. Farhadi. Yolo9000: better, faster, stronger. *arXiv preprint*, 2017.

[25] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*, pages 91–99, 2015.

[26] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. *IJCV*, 115(3):211–252, 2015.

[27] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini. The graph neural network model. *TNN*, 20(1):61–80, 2009.

[28] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.

[29] Z. Wang, T. Chen, J. Ren, W. Yu, H. Cheng, and L. Lin. Deep reasoning with knowledge graph for social relationship understanding. In *IJCAI*, pages 1021–1028, 2018.

[30] D. Xu, Y. Zhu, C. B. Choy, and L. Fei-Fei. Scene graph generation by iterative message passing. In *CVPR*, 2017.

[31] B. Yao and L. Fei-Fei. Grouplet: A structured image representation for recognizing human and object interactions. In *CVPR*, pages 9–16. IEEE, 2010.

[32] M. Yatskar, L. Zettlemoyer, and A. Farhadi. Situation recognition: Visual semantic role labeling for image understanding. In *CVPR*, pages 5534–5542, 2016.

[33] R. Zellers, M. Yatskar, S. Thomson, and Y. Choi. Neural motifs: Scene graph parsing with global context. *arXiv preprint arXiv:1711.06640*, 2017.

[34] C. L. Zitnick, D. Parikh, and L. Vanderwende. Learning the visual interpretation of sentences. In *ICCV*, pages 1681–1688, 2013.