Scene Graph Generation via Convolutional Message Passing and Class-Aware Memory Embeddings

Yidong Zhang[1], Yunhong Wang[1], and Yuanfang Guo[2,3](✉)

[1] Beijing Advanced Innovation Center for Big Data and Brain Computing, Beihang University, Beijing, China
{zhydong,yhwang}@buaa.edu.cn
[2] Laboratory of Intelligent Recognition and Image Processing, School of Computer Science and Engineering, Beihang University, Beijing, China
andyguo@buaa.edu.cn
[3] Science and Technology on Information Assurance Laboratory, Beijing, China

Abstract. Detecting visual relationships between objects in an image still remains challenging, because the relationships are difficult to be modeled and the class imbalance problem tends to jeopardize the predictions. To alleviate these problems, we propose an end-to-end approach for scene graph generations. The proposed method employs the ResNet as the backbone network to extract the appearance features of the objects and relationships. An attention based graph convolutional network is exploited and modified to extract the contextual information. Language and geometric priors are also utilized and fused with the visual features to better describe the relationships. At last, a novel memory module is designed to alleviate the class imbalance problem. Experimental results demonstrate the validity of our model and our superiority compared to our baseline technique.

Keywords: Visual relationship detection · Scene graph generation · Graph convolutional neural networks · LSTM

1 Introduction

Scene graph, whose nodes are object entities in the image and edges depict the pairwise relationships of the objects, represents a high-level abstraction of an image. It not only contains the geometric and semantic information of objects, but also describes the correlations, i.e., the visual relationships, among the objects. As shown in the scene graph example in Fig. 1, a more advanced representation to an image is acquired by visual relationship detection (VRD) compared to the regular computer vision tasks, such as object detection and segmentation [5,12,13,17]. According to the existing literatures, other complex computer vision tasks such as image retrieval [9,24], image generation [8], image captioning [1,11], etc., can be improved based on the VRD result.

I. V. Tetko et al. (Eds.): ICANN 2019, LNCS 11729, pp. 620–633, 2019.
https://doi.org/10.1007/978-3-030-30508-6_49

Different from the typical object classification task [5,19], VRD is usually more challenging, because different pairs of objects may possess identical relationships. The visual appearances of the relationships are highly depending on the object classes. For example, the relationship "on" may appear as a man walking "on" a road, or a cup "on" a desk, etc. Moreover, relationship datasets usually remain class imbalance problem, which increases difficulties of modeling relationships.

Although the existing techniques have explored various features/priors, such as visual features [21], language priors [14], geometric features [2] and object class prior [23], the VRD performances still desire improvements. Therefore, inspired by [22], we propose an end-to-end VRD approach to generate the scene graphs of the input images. Specifically, ResNet is utilized as the backbone network to extract the visual features. After the feature extraction, attention based graph convolutional network (aGCN) [6,22] is exploited to model the visual cues of the relationships and objects. To better preserve the spatial information, which is essential for relationship predictions, convolution layers are exploited instead of the fully connected layers in the existing aGCN. Then, with the fused features, we construct a memory module, which is based on long-short term memory (LSTM) [7], to reduce the interferences from imbalanced number of samples in different relationship classes.

Our contributions are summarized as follows:

(1) We propose an end-to-end VRD method to generate the scene graphs from images by preserving the spatial information and relieving the class imbalance problem.
(2) We exploit the attention based graph convolutional network to describe the relationship patterns and propose to replace the fully connected layers with convolution layers to maintain more spatial information.
(3) We propose an LSTM based memory module to alleviate the class imbalance problem for VRD task.
(4) Experimental results demonstrate that our proposed network can successfully improve the performances compared to the baseline approach while reduce the negative effects caused by the class imbalance problem.

2 Related Work

In the past decade, many researches have been conducted on VRD and various features/priors have been explored. Relationships are modeled as the visual phrases, i.e., the $\langle sub, pred, obj \rangle$ triplet[1] is employed to represent the objects and their relationships, in [18]. Computing the triplets directly [18] gives poor performance as well as high computational complexity. To improve the performance, [14] predicts the classes of *sub*, *obj* and *pred* individually. Besides, language priors (word vector [15,16]) are also explored in [14]. The visual appearance features

[1] In this paper, "*obj*" denotes the object in relationship triplets $\langle sub, pred, obj \rangle$, and "object" denotes the perceived object in the image.

are also extracted in [21] by a message passing scheme. This scheme jointly considers the objects and the relationships between each pair of the objects. Besides of the visual appearance features and language priors, other features can also be exploited. The relative locations and sizes of the bounding boxes are modeled as binary masks in [2]. [23] concludes that the object classes are important priors in predicting the relationships. LinkNet [20] utilizes contextual features by adopting multi-label classification as their subtask and designs relationship and object embedding module.

Recently, more mechanisms are explored. [11] performs a multi-task learning by jointly tackling the object detection, VRD and image captioning tasks, and proves that these three tasks can benefit from each other. [22] predicts the relationships via an attention based graph convolutional network.

In general, existing approaches have discovered various kinds of mechanisms to solve the VRD problems. Unfortunately, current performances are still relatively low, thus more efforts are desired.

3 Proposed Work

In practice, the nodes of a scene graph usually represent the coordinates of object bounding boxes with object class labels. Meanwhile, the edges usually stand for the class labels of relationships. We denote scene graph as a two-tuple $G = \{O, R\}$. O and R denote objects and relationships, respectively. Given an image I, our objective is to predict the probability of the scene graph as Eq. 1 shows.

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

Note that I in Eq. 1 can be omitted for simplicity. $P(O)$ denotes the probabilities of the object class labels and $P(R|O)$ denotes the probability of the relationship labels given the object class labels.

3.1 Pipeline

The pipeline of our network is shown in Fig. 1. Given the object bounding boxes, each two of the object bounding boxes are combined to form a relationship bounding box. With the object and relationship bounding boxes, the object and relationship features are acquired from the top layer of ResNet [5] with *RoIAlign* [4]. Note that the object and relationship feature maps are cropped from the output of the top layer of the backbone network.

Then, the object and relationship features are iteratively updated by an attention based graph convolutional network in the Message Passing (MP) module. Since the visual appearance features may induce inaccurate predictions of the relationships, the language and geometric priors are fused with visual appearance features in the Message Fusion (MF) module. Then, the preliminary relationship classes are generated. To alleviate the class imbalance problem, the Memory (MM) module is proposed by utilizing the sample frequencies to refine the outputs of MF. At last, the outputs of the MF and MM modules are combined to obtain the final triplets.

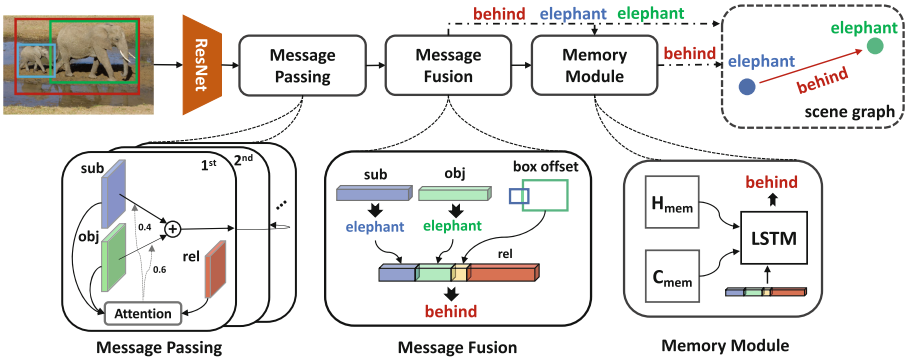

Fig. 1. Pipeline of our network. The "sub", "obj" and "rel" denote subject, object and relationship, respectively. The residual shortcuts are omitted in Message Passing for convenience. The notched arrows denote the softmax outputs.

3.2 Message Passing Module

The message passing scheme has been proven to be efficient in [21]. The model passes the messages, which contains contextual features, among the nodes and edges in a scene graph to refine the object and relationship features. In our work, we exploit an attention based graph convolutional network with residual shortcuts to pass the messages between the objects and relationships. The residual shortcuts can accelerate the training of the network and reduce the degradations [5].

[22] allows the network to pass messages from the relationship features to the object features. Unfortunately, the relationship features usually contain relatively large regions of backgrounds, which are equivalent to noises to the intended object features. Therefore, to avoid potential degradations, we only pass messages from the object features to the relationship features in our MP module.

Let $\boldsymbol{f}_{r,i}^{(t)}$ and $\boldsymbol{f}_{o,j}^{(t)}$ be the i^{th} relationship feature and the j^{th} object feature in the t^{th} iteration. Note that the subscript r represents relationship and o denotes object. $\boldsymbol{W}$ and $\boldsymbol{V}$ stand for the parameters to be trained and α is the attention weight. The operator $\otimes$ stands for the convolution operation. $sub_{r,i}$ and $obj_{r,i}$ respectively denote the sub and obj, which belong to the i^{th} relationship triplet.

With the symbols defined above, the relationship features can be updated by Eq. 2.

$$\boldsymbol{f}_{r,i}^{(t+1)} = \boldsymbol{f}_{r,i}^{(t)} + \boldsymbol{W}_r \otimes \boldsymbol{f}_{r,i}^{(t)} + \sum_{p \in \{sub_{r,i}, obj_{r,i}\}} \alpha_{i,p} \boldsymbol{W}_{o,p} \otimes \boldsymbol{f}_{o,p}^{(t)} \tag{2}$$

Similarly, the object features can be updated by Eq. 3.

$$\boldsymbol{f}_{o,i}^{(t+1)} = \boldsymbol{f}_{o,i}^{(t)} + \boldsymbol{V}_o \otimes \boldsymbol{f}_{o,i}^{(t)} \tag{3}$$

Since the spatial information usually serves effectively in the prediction of the relationships (e.g. "on" and "next to"), we utilize the convolution ($conv$) layers

rather than the fully-connected (fc) layers to process the graph, because the feature maps tend to contain more spatial information than the feature vectors. Note that the initial feature maps $\boldsymbol{f}_r^{(0)}$ and $\boldsymbol{f}_o^{(0)}$ are obtained from the backbone network.

3.3 Message Fusion Module

In a typical relationship triplets $\langle sub, pred, obj \rangle$, the class labels of obj and sub are usually strong priors for $pred$. Meanwhile, the relative locations and sizes between the objects, which usually serve as the geometric features, are vital for VRD.

In MF module, the object feature maps $\boldsymbol{f}_o$ and the relationship feature maps $\boldsymbol{f}_r$ are transformed into 2048-dim feature vectors. We denote the object feature vectors as $\boldsymbol{v}_o$, and the relationship feature vectors as $\boldsymbol{v}_r$.

The object classes are directly predicted according to $\boldsymbol{v}_o$ with a softmax layer. With these predicted labels, language priors can be constructed in the form of word vectors. The bounding box offsets $\boldsymbol{e}$, which is proposed in [3], are adopted to represent the geometric priors as shown in Eq. 4.

$$\boldsymbol{e} = \left(\frac{x_o - x_s}{w_s}, \frac{y_o - y_s}{h_s}, \log\left(\frac{w_o}{w_s}\right), \log\left(\frac{h_o}{h_s}\right) \right) \tag{4}$$

where the subscript s and o represent the sub and obj in relationship triplets, respectively. After the word vectors and bounding box offset vectors are obtained, they are mapped to higher dimensions with fc layers and concatenated with $\boldsymbol{v}_r$.

The initial relationship class labels can be then calculated from the concatenated features, and the prediction will be refined in the next module.

3.4 Memory Module

The samples with high appearing frequencies usually dominate the network training process when the networks are trained on a dataset with imbalanced classes. Under such circumstances, the learned parameters are less capable of classifying the samples with less appearing frequencies.

To tackle this problem, a Memory (MM) module, which manages memory vectors, is proposed. Each of these vectors, which are essentially the memory embeddings, is updated on one specific class of samples.

These memory embeddings are independent from the parameters which will be trained in the ordinary back propagation process. The memory embedding of class i^{th} intends to remember the information from the samples of class i^{th} in the training stage. Inspired by the hidden and cell states in LSTM, which serve similarly to our proposed memory embeddings, we construct our Memory module based on the standard LSTM structure, as shown in Fig. 2.

Assume the size of each memory embedding is d, and the number of the relationship classes is m. Note that the 0^{th} relationship is denoted as $irrelevant$. In MM module, two matrices $\boldsymbol{H}_{mem} \in \mathbb{R}^{(m+1)\times d}$ and $\boldsymbol{C}_{mem} \in \mathbb{R}^{(m+1)\times d}$ are constructed for storing the memory embeddings. In our design, $\boldsymbol{H}_{mem}$ serves similarly to the hidden state in LSTM, while $\boldsymbol{C}_{mem}$ contributes as the cell state.

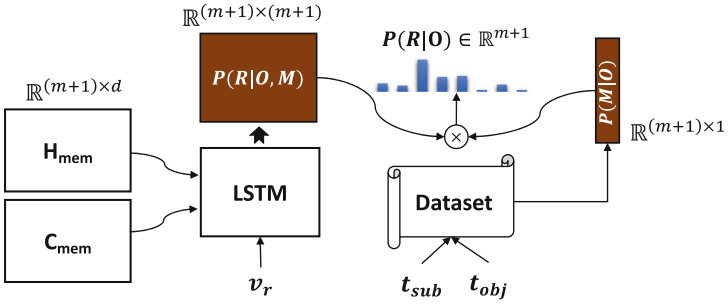

Fig. 2. The inference scheme of our Memory module. $\boldsymbol{v}_r$ denotes the relationship feature adopted from MF module. $\boldsymbol{t}_{sub}$ and $\boldsymbol{t}_{obj}$ denote the classes of subject and object predicted in MF module. m is the number of relationship classes and d is the size of each memory embedding.

Updating. In the training stage, $\boldsymbol{H}_{mem}$ and $\boldsymbol{C}_{mem}$ are indexed with the ground truth labels of the relationships. When a sample $\boldsymbol{v}_r$ is being processed by the MM module, according to the ground truth labels of $\boldsymbol{v}_r$, the MM module will fetch the corresponding rows of the memory embeddings and feed them into LSTM. After $\boldsymbol{v}_r$ is processed, the hidden and cell states are assigned back to the corresponding locations in the matrices.

With the mechanism of the proposed MM module, the i^{th} row of the memory embeddings will only be updated by the samples of the i^{th} class. Thus the MM module can prevent the information, which is learned from the rare samples, from being corrupted by the information from the frequently appearing samples.

Inference. With the memories of each classes stored in the memory embeddings, we designed a scheme to infer the final relationship predictions as Fig. 2 shows.

After the labels of the objects are acquired, the class labels of the relationships can be computed based on $P(R|O) \in \mathbb{R}^{m+1}$ as shown in Eq. 5.

$$P(R|O) = P(R,O)/P(O) \tag{5}$$

Let M_i represents the memory embeddings of the i^{th} relationship class. Then, $P(R,O)$ can be computed via Eq. 6.

$$P(R,O) = \sum_{i=0}^{m} P(R,O,M_i) = \sum_{i=0}^{m} P(R|O,M_i)P(O,M_i) \tag{6}$$

where $P(O,M_i)$ can be decomposed by the Bayesian formula via Eq. 7.

$$P(O,M_i) = P(M_i|O)P(O) \tag{7}$$

By substituting Eqs. 6 and 7 into Eq. 5, we obtain

$$P(R|O) = \sum_{i=0}^{m} P(R|O,M_i)P(M_i|O), \tag{8}$$

where $P(R|O, M_i)$ denotes the probability of the relationship when objects O are detected and the memory embedding of the i^{th} relationship class is fetched. $P(M_i|O)$ is regarded as the probability of selecting the i^{th} row of the memory embeddings, given the objects O are detected. The $P(R|O)$ in Eq. 8 is the final relationship prediction in MM module.

Obviously, $P(R|O, M_i) \in \mathbb{R}^{m+1}$ is the output probability of LSTM when the i^{th} rows of $\boldsymbol{H}_{mem}$ and $\boldsymbol{C}_{mem}$ are fetched to be the hidden and cell states.

The frequencies of relationships, which are usually provided in the datasets and can be estimated in practice, are adopted to estimate $P(M_i|O)$. We count the probabilities of all possible relationships given two object classes t_{sub} and t_{obj} predicted from the MF module, which are denoted as $\hat{P}(R|t_{sub}, t_{obj}) \in \mathbb{R}^{m+1}$. Then, $P(M_i|O)$ can be approximated by $P(M_i|O) \approx \hat{P}(R_i|t_{sub}, t_{obj})$.

Note that there may not exist any relationship label for some object class pairs t_{sub} and t_{obj} in practice, i.e. $\hat{P}(R|t_{sub}, t_{obj}) \equiv \mathbf{0}$, because certain pairs tend to possess no correlations intuitively. Under such circumstances, the initial predicted relationship from the MF module will be considered as the final relationship prediction result instead of $P(R|O)$ in Eq. 8.

4 Experimental Results

Our proposed method was evaluated on the Visual Genome (VG) dataset [10]. Since the original VG dataset was sparsely annotated, we adopted the preprocessed VG dataset, which was filtered and split by [21], as [20–22]. This preprocessed dataset selects the most frequently appearing 150 object classes and 50 relationship classes and contains 75k images for training, 5k images for validating and 32k images for testing. In the experiments, the scene graph classification (**SGCls**) and predicate classification (**PredCls**) tasks were solved and the **Recall@K** metric [14] was employed for assessment. A single edge between each two object nodes was predicted in the scene graph in our experiments.

In **SGCls**, the VRD model is usually given an image with certain object bounding boxes, and classifies the objects and the relationships between these objects. On the other hand, **PredCls** only requires the VRD model to classify the relationships between the objects, because the image, object bounding boxes and the object labels are all given. The **PredCls** metric evaluates the ability to detect relationships among the known objects, while **SGCls** judges the ability to generate the relationship triplets.

Recall@K computes the ratio of true positive triplets $\langle sub, pred, obj \rangle$ to the overall ground truth triplets in the top-K probable relationship outputs as

$$Recall@K = \sum_{i=1}^{N} \frac{|TP_i[:k]|}{|GT_i|}, \tag{9}$$

where N denotes the total number of images, TP_i and GT_i represent the true positive triplets and ground truth triplets, respectively, and $[:k]$ stands for the top-k probable values.

4.1 Implementation Details

The cross-entropy loss was adopted in both the object and relationship classification tasks. The overall loss of our model during the training process is shown as

$$L = L_{object} + L_{rel\_mf} + L_{rel\_mm}, \tag{10}$$

where L_{object} is the object classification loss in MF module, and L_{rel_mf} and L_{rel_mm} represent relationship classification losses in the MF and MM module, respectively.

The input images were reshaped to 592×592. Our network was trained with batch size 6 on a single Nvidia 1080Ti GPU. The learning rate was 1e-3 at the beginning and decayed by the PyTorch ReduceLROnPlateau rule.

We pre-trained the ResNet in faster RCNN [17] on the VG dataset on object detection task. Then, our end-to-end network was trained without fixing any parameters.

4.2 A Study of the MP Module

Number of Iterations. Our model was assessed with different number of iterations of MP on the validation set. As shown in Table 1(left), different numbers of iterations of the message passing tend to give different performances. Inadequate iterations only result in less capability of modeling the contexts and correlations between the objects and relationships. Meanwhile, too many iterations tend to induce overfitting, and thus jeopardize both the performance and the convergence speed of our model. Since 2 iterations of MP introduced the best performance, the latter experiments were performed with this setting.

Table 1. (left) Results of different MP iteration numbers. "iter" stands for the number of iterations of the MP module. (right) Results of different layers employed in the attention based graph convolutional network.

	SGCls		
iter	R@20	R@50	R@100
0	22.56	26.98	29.05
1	25.55	28.82	29.94
2	**26.45**	**30.11**	**31.26**
3	25.91	29.72	31.15

	SGCls		
	R@20	R@50	R@100
fc	24.94	28.11	29.20
conv	**26.45**	**30.11**	**31.26**

Conv vs. FC. In previous attention based graph convolutional network, nodes were processed with the fc layers. On the contrary, the *conv* layers are adopted in our model. Therefore, we compared these two types of layers on the validation set in **SGCls** task to verify the effectiveness of our modification.

In this experiment, only the MP module and the backbone network were employed. Table 1(right) indicates that the *conv* layers outperform the fc layers in our MP module, because the *conv* layers can maintain more spatial information for the relationship predictions.

4.3 Ablation Study

Here, the ablation study was conducted to validate each module in our network on the testing set. The results are reported in Table 2.

Table 2. Results of ablation study.

MP	MF	MM	**SGCls**			**PredCls**		
			R@20	R@50	R@100	R@20	R@50	R@100
			17.97	21.79	23.29	30.18	38.78	42.44
√			21.09	24.08	25.02	38.85	46.54	49.32
√	√		31.12	**34.41**	35.81	**55.92**	**63.61**	66.34
√	√	√	**31.67**	34.40	**35.90**	55.61	63.54	**66.61**

As can be observed, the MP module gives about 3% gains in **SGCls** and 8% gains in **PredCls** compared to the prediction results of the backbone network. These improvements indicate that the modified attention based graph convolutional network can effectively model the relationships. Note that the gains in **PredCls** are more than that in **SGCls**, which proves that the gains achieved by the MP module are mainly contributed by a more precise modeling of the relationship patterns, rather than optimizing the object classification results. After the MF module is added, the performance raised 8%–15%, which clearly demonstrates the effectiveness of the language priors and geometric information.

Since our MM module only refines the existing prediction results to alleviate the class imbalance problem, the performance gains of the MM module in Table 2 are less obvious, we will further study the validity of our MM module in Sect. 4.4.

4.4 A Study of the Memory Module

We tested our model with/without the MM module on the **PredCls** task, and compared their **R@50** values. The results of the specific relationship prediction improvement are reported in Fig. 3. As shown in Fig. 3, the recall rate of some relationships with certain appearing frequencies decreases while the recall rate of more relationships with relatively less appearing frequencies achieves obvious improvements, with the MM module.

This imbalanced dataset resembles the relationships in reality. Typically, people may employ “on” in a relationship description for thousands of times while “laying on” may only be selected for a few times. If a model simply predicts

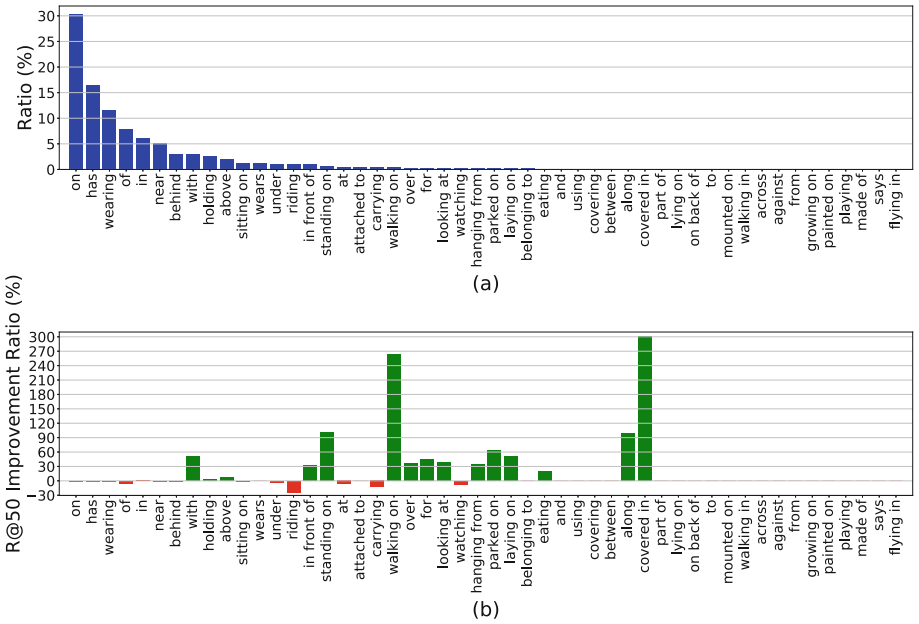

Fig. 3. Results of the specific relationship prediction improvement. (a) Sample distribution of the relationships in VG dataset. (b) The **R@50** improvement ratio (%) for the individual relationship classes with the MM module.

all the relationships to be the frequently appearing relationships such as "on", "above" or "beside", it may achieve a high recall value in assessments. However, these frequently appearing relationships usually contain less semantic information and tend to give superficial descriptions of the images. On the contrary, the less frequently appearing relationships, such as "laying on", "walking in", tend to give a more accurate descriptions of the images.

To generate the scene graph more precisely, we tried to boost the recall rate of our model to enhance the ability of our network to fit the data distribution of the dataset. But we believe that a VRD model should not only achieve a high recall rate in the experiments, but also generate relationships containing more semantic information. It motivated us to design a model to generate more meaningful relationships. As can be observed from the experimental results, we propose a particular memory module to alleviate the class imbalance problem and extract deeper semantic relationship information among the objects, which can aid the VRD problem in the future.

4.5 Objective Results

Since our proposed model is developed based on Graph RCNN [22], we chose Graph RCNN as our baseline technique. We also compared our method to [2,14,21], because our model exploits the language priors [14], message passing

structure [21] and geometric features [2]. Note that DR-Net [2] was not evaluated on the same version of VG dataset as ours and it allows multiple-edge predictions between each two of the object nodes. Predicting multiple edges usually gives higher performance as proven in [23]. Thus, we reimplemented DR-Net and evaluated it on the preprocessed VG dataset [21] for fair comparisons. Following the experimental protocols in [22], our method was also compared to motifs frequency model [23], which predicts the relationships according to the frequencies of the potential relationships. In addition, the results of the state-of-the-art method LinkNet [20] are shown in Table 3.

Table 3. Quantitative comparisons. All the numbers are in %. The omitted values indicate that the results are not presented in the original paper. The results of [14] is adopted from [21] because the original paper did not evaluate the model on VG. Results of DR-Net was reproduced by us.

Methods	**SGCls**			**PredCls**		
	R@20	R@50	R@100	R@20	R@50	R@100
Language prior [14]	-	11.8	14.1	-	27.9	35.0
IMP [21]	-	21.7	24.4	-	44.8	53.0
DR-Net [2]	27.8	32.0	33.4	37.5	45.4	48.3
Motif-Freq [23]	27.7	32.4	34.0	49.4	59.9	64.1
Graph RCNN [22]	-	29.6	31.6	-	54.2	59.1
Our method	31.7	34.4	35.9	55.6	63.5	66.6
(**SOTA**) LinkNet [20]	38.3	41	41.7	61.8	67.0	68.5

As can be observed, our proposed model obviously outperforms our baseline Graph RCNN. The results of our model also surpass [2,14,21] and the frequency model [23] in both the **SGCls** and **PredCls** task. These results indicate that our model can generate a more precise scene graph comparing with the most of the existing methods which exploit similar structures and features.

Unfortunately, our model generates less decent objective results compared to the state-of-the-art LinkNet. As can be observed, the performance gap between our method and LinkNet is smaller for the **PredCls** task than the **SGCls** task. A potential explanation to this phenomenon is that LinkNet extracts context features from the overall image [20] to improve the object classification while we only focused on refining the relationship features. Besides, since our main objective is to alleviate the class imbalance problem, we spent less efforts on enhancing the overall recall value, which may not be optimized as illustrated in Sect. 4.4.

4.6 Subjective Results

Some subjective results of our model for the **SGCls** task are given in Fig. 4. As can be observed from Figs. 4(a), (b), (c), our model gives excellent predic-

tions when the object classes are accurately classified. On the other hand, as Figs. 4(d) and (e) show, the inaccurate predictions are mainly induced by the misclassifications of the object classes.

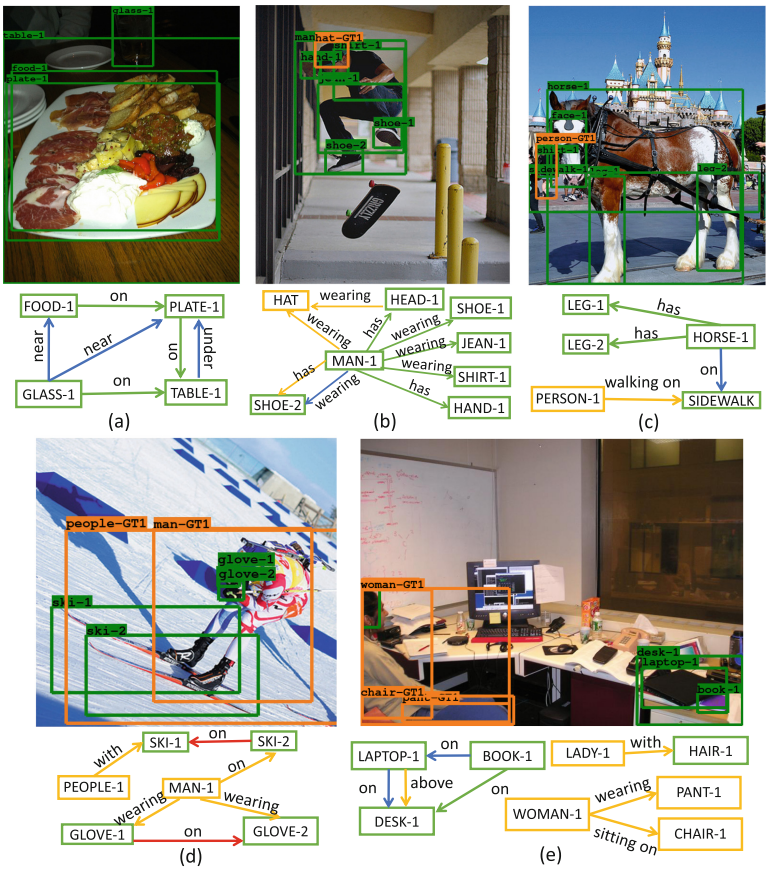

Fig. 4. Subjective results of our method. (a) (b) (c) demonstrate good predictions while (d) and (e) give failure cases. The green color indicate a correct and labelled prediction. The blue color means a correct prediction yet the label does not exist in the dataset. The yellow color represents unpredicted labels in the dataset. The red color is an inaccurate and unlabelled prediction. (Color figure online)

The evaluation results are highly depending on the annotations of the dataset. For example, in Fig. 4(b), our model predicts $\langle man-1, wearing, shoe-2\rangle$ while the ground truth label is $\langle man-1, has, shoe-2\rangle$. Although our prediction predicts a relationship which is different yet more accurate for the descriptions of the image, our recall value decreases because of the mismatch between the output and ground truth label. These ambiguous annotations are one of the main reasons of the class imbalance problem. Since the evaluations are usually depending

on the ground truth labels, these ambiguous annotations also jeopardize the correctiveness of the evaluations. Besides, these ambiguous annotations also serve as the obstacles in the training process by deteriorating the convergences of the cross-entropy loss.

5 Conclusion

In this paper, we proposed an end-to-end scheme to tackle the VRD problem. By replacing the fc layers with the *conv* layers in aGCN, we managed to maintain more spatial information. The proposed model also benefited from the extracted geometric features and the language priors for generating precise scene graphs. At last, the class imbalance problem could be alleviated by the proposed memory module. Extensive experiments verified the validity of the proposed method and demonstrated its superiority against our baseline method.

Acknowledgement. This work was supported in part by the National Natural Science Foundation of China under Grant 61573045 and Grant 61802391, in part by the Foundation of Science and Technology on Information Assurance Laboratory under Grant KJ-17-006, and in part by the Fundamental Research Funds for the Central Universities.

References

1. Anderson, P., Fernando, B., Johnson, M., Gould, S.: SPICE: semantic propositional image caption evaluation. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9909, pp. 382–398. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46454-1_24
2. Dai, B., Zhang, Y., Lin, D.: Detecting visual relationships with deep relational networks. In: IEEE CVPR, pp. 3076–3086 (2017). https://doi.org/10.1109/CVPR.2017.352
3. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: IEEE CVPR, pp. 580–587 (2014). https://doi.org/10.1109/CVPR.2014.81
4. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask R-CNN. In: IEEE ICCV, pp. 2961–2969 (2017). https://doi.org/10.1109/ICCV.2017.322
5. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: IEEE CVPR, pp. 770–778 (2016). https://doi.org/10.1109/CVPR.2016.90
6. Henaff, M., Bruna, J., LeCun, Y.: Deep convolutional networks on graph-structured data. arXiv preprint arXiv:1506.05163 (2015)
7. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997). https://doi.org/10.1007/978-3-642-24797-2_4
8. Johnson, J., Gupta, A., Fei-Fei, L.: Image generation from scene graphs. In: IEEE CVPR, pp. 1219–1228 (2018). https://doi.org/10.1109/CVPR.2018.00133
9. Johnson, J., et al.: Image retrieval using scene graphs. In: IEEE CVPR, pp. 3668–3678 (2015). https://doi.org/10.1109/CVPR.2015.7298990
10. Krishna, R., et al.: Visual genome: connecting language and vision using crowdsourced dense image annotations. IJCV **123**(1), 32–73 (2017). https://doi.org/10.1007/s11263-016-0981-7

11. Li, Y., Ouyang, W., Zhou, B., Wang, K., Wang, X.: Scene graph generation from objects, phrases and region captions. In: IEEE ICCV, pp. 1261–1270 (2017)
12. Lindh, A., Ross, R.J., Mahalunkar, A., Salton, G., Kelleher, J.D.: Generating diverse and meaningful captions. In: Kůrková, V., Manolopoulos, Y., Hammer, B., Iliadis, L., Maglogiannis, I. (eds.) ICANN 2018. LNCS, vol. 11139, pp. 176–187. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01418-6_18
13. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: IEEE CVPR, pp. 3431–3440 (2015). https://doi.org/10.1109/TPAMI.2016.2572683
14. Lu, C., Krishna, R., Bernstein, M., Fei-Fei, L.: Visual relationship detection with language priors. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9905, pp. 852–869. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46448-0_51
15. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: NeurIPS, pp. 3111–3119 (2013)
16. Pennington, J., Socher, R., Manning, C.: Glove: global vectors for word representation. In: EMNLP, pp. 1532–1543 (2014). https://doi.org/10.3115/v1/D14-1162
17. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. In: NeurIPS, pp. 91–99 (2015). https://doi.org/10.1109/TPAMI.2016.2577031
18. Sadeghi, M.A., Farhadi, A.: Recognition using visual phrases. In: IEEE CVPR, pp. 1745–1752 (2011). https://doi.org/10.1109/CVPR.2011.5995711
19. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
20. Woo, S., Kim, D., Cho, D., Kweon, I.S.: LinkNet: relational embedding for scene graph. In: NeurIPS, pp. 560–570 (2018)
21. Xu, D., Zhu, Y., Choy, C.B., Fei-Fei, L.: Scene graph generation by iterative message passing. In: IEEE CVPR, pp. 5410–5419 (2017). https://doi.org/10.1109/CVPR.2017.330
22. Yang, J., Lu, J., Lee, S., Batra, D., Parikh, D.: Graph R-CNN for scene graph generation. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) ECCV 2018. LNCS, vol. 11205, pp. 690–706. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01246-5_41
23. Zellers, R., Yatskar, M., Thomson, S., Choi, Y.: Neural motifs: scene graph parsing with global context. In: IEEE CVPR, pp. 5831–5840 (2018). https://doi.org/10.1109/CVPR.2018.00611
24. Zhang, W., Cao, X., Wang, R., Guo, Y., Chen, Z.: Binarized mode seeking for scalable visual pattern discovery. In: IEEE CVPR, pp. 6827–6835 (2017). https://doi.org/10.1109/CVPR.2017.722