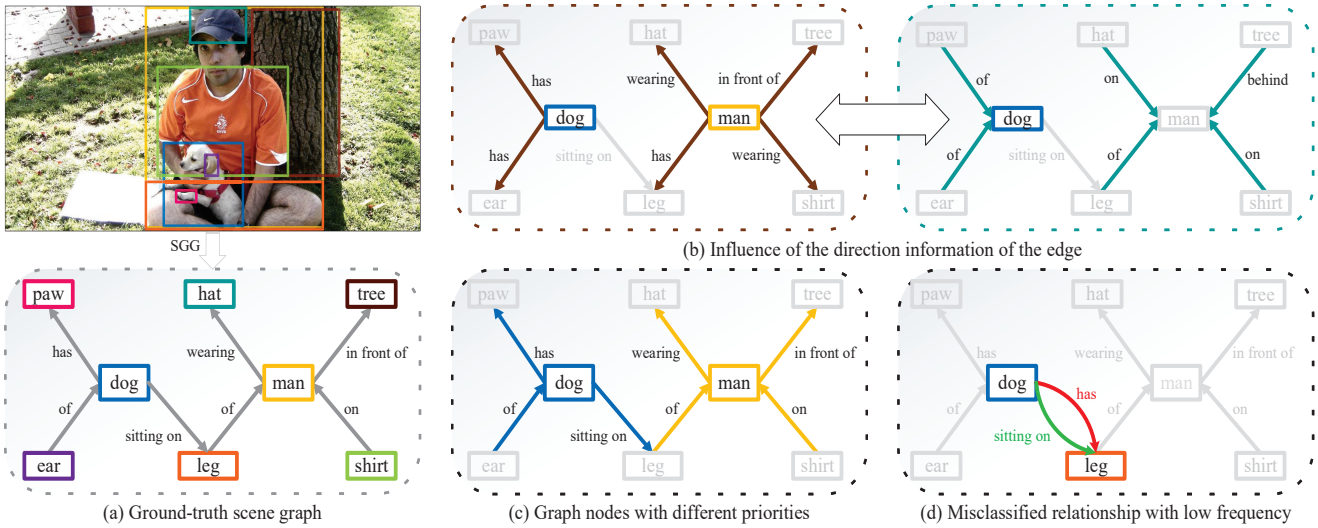
GPS-Net: Graph Property Sensing Network for Scene Graph Generation

Xin Lin[1] Changxing Ding[1] Jinquan Zeng[1] Dacheng Tao[2]
[1] School of Electronic and Information Engineering, South China University of Technology
[2] UBTECH Sydney AI Centre, School of Computer Science, Faculty of Engineering,
The University of Sydney, Darlington, NSW 2008, Australia
{eelinxin,eetakchatsau}@mail.scut.edu.cn chxding@scut.edu.cn dacheng.tao@sydney.edu.au



(a) Ground-truth scene graph (b) Influence of the direction information of the edge (c) Graph nodes with different priorities (d) Misclassified relationship with low frequency

Figure 1: (a) The ground-truth scene graph for one image. (b) The direction of the edge specifies the subject and object, and also affects the relationship type and node-specific context. (c) The priority of nodes varies, according to the number of triplets included in the graph. (d) The long-tailed distribution of relationships causes error for low-frequency relationships, *e.g.*, the failure in recognizing *sitting on*.

Abstract

Scene graph generation (SGG) aims to detect objects in an image along with their pairwise relationships. There are three key properties of scene graph that have been under-explored in recent works: namely, the edge direction information, the difference in priority between nodes, and the long-tailed distribution of relationships. Accordingly, in this paper, we propose a Graph Property Sensing Network (GPS-Net) that fully explores these three properties for SGG. First, we propose a novel message passing module that augments the node feature with node-specific contextual information and encodes the edge direction information via a tri-linear model. Second, we introduce a node priority sensitive loss to reflect the difference in priority between nodes during training. This is achieved by designing a mapping function that adjusts the focusing parameter in the focal loss. Third, since the frequency of relationships is affected by the long-tailed distribution problem, we mitigate this issue by first softening the distribution and then enabling it to be adjusted for each subject-object pair according to their visual appearance. Systematic experiments demonstrate the effectiveness of the proposed techniques. Moreover, GPS-Net achieves state-of-the-art performance on three popular databases: VG, OI, and VRD by significant gains under various settings and metrics. The code and models are available at `https://github.com/taksau/GPS-Net`.

1. Introduction

Scene Graph Generation (SGG) provides an efficient way for scene understanding and valuable assistance for various computer vision tasks, including image captioning [1], visual question answering [2] and 3D scene synthesis [3]. This is mainly because the scene graph [4] not only records the categories and locations of objects in the scene

but also represents pairwise visual relationships of objects.

As illustrated in Figure 1(a), a scene graph is composed of multiple triplets in the form <subject-relationship-object>. Specifically, an object is denoted as a node with its category label, and a relationship is characterized by a directed edge between two nodes with a specific category of predicate. The direction of the edge specifies the subject and object in a triplet. Due to the complexity in relationship characterization and the imbalanced nature of the training data, SGG has emerged as a challenging task in computer vision.

Multiple key properties of the scene graph have been under-explored in the existing research, such as [5, 6, 7]. The first of these is edge direction. Indeed, edge direction not only indicates the subject and object in a triplet, but also affects the class of the relationship. Besides, it influences the context information for the corresponding node, as shown in recent works [8, 9]. An example is described in Figure 1(b), if the direction flow between *man* and the other objects is reversed, the focus of the context will change and thus affects the context information for all the related nodes. This is because that the importance of nodes varies according to the number of triplets they are included in the graph. As illustrated in Figure 1(c), *leg*, *dog* and *man* are involved in two, three, and four triplets in the graph, respectively. Hence, considering the contribution of each node to this scene graph, the priority in object detection should follow the order: $man > dog > leg$. However, existing works usually treat all nodes equally in a scene graph.

Here, we propose a novel direction-aware message passing (DMP) module that makes use of the edge direction information. DMP enhances the feature of each node by providing node-specific contextual information with the following strategies. First, instead of using the popular first-order linear model [10, 11], DMP adopts a tri-linear model based on Tucker decomposition [12] to produce an attention map that guides message passing. In the tri-linear model, the edge direction affects the attention scores produced. Second, we augment the attention map with its transpose to account for the uncertainty of the edge direction in the message passing step. Third, a transformer layer is employed to refine the obtained contextual information.

Afterward, we devise a node priority-sensitive loss (NPS-loss) to encode the difference in priority between nodes in a scene graph. Specifically, we maneuver the loss contribution of each node by adjusting the focusing parameter of the focal loss [13]. This adjustment is based on the frequency of each node included in the triplets of the graph. Consequently, the network can pay more attention to high priority nodes during training. Comparing with [11] (exploiting a non-differentiable local-sensitive loss function to represent the node priority), the proposed NPS-loss is differentiable and convex, and so it can be easily optimized by gradient descent based methods and deployed to other SGG models.

Finally, the frequency distribution of relationships has proven to be useful as prior knowledge in relationship prediction [7]. However, since this distribution is long-tailed, its effectiveness as the prior is largely degraded. For example, as shown in Figure 1(d), one SGG model tends to misclassify *sitting on* as *has* since the occurrence rate of the latter is relatively high. Accordingly, we propose two strategies to handle this problem. First, we utilize a log-softmax function to soften the frequency distribution of relationships. Second, we propose an attention model to adaptively modify the frequency distribution for each subject-object pair according to their visual appearance.

In summary, the innovation of the proposed GPS-Net is three-fold: (1) DMP for message passing, which enhances the node feature with node-specific contextual information; (2) NPS-loss to encode the difference in priority between different nodes; and (3) a novel method for handling the long-tailed distribution of relationships. The efficacy of the proposed GPS-Net is systematically evaluated on three popular SGG databases: Visual Genome (VG) [14], OpenImages (OI) [15] and Visual Relationship Detection (VRD) [16]. Experimental results demonstrate that the proposed GPS-Net consistently achieves top-level performance.

2. Related Work

Visual Context Modeling: Recent approaches for visual context modeling can be divided into two categories, which model the global and object-specific context, respectively. To model the global context, SENet [17] and PSANet [18] adopt rescaling to different channels in feature maps for feature fusion. In addition, Neural Motif [7] represents the global context via Long Short-term Memory Networks.

To model the object-specific context, NLNet [19] adopts self-attention mechanism to model the pixel-level pairwise relationships. CCNet [20] accelerates NLNet via stacking two criss-cross blocks. However, as pointed out in [21], these methods [22, 23, 24] may fail to learn object-specific context due to the utilization of the first-order linear model. To address this issue, we design a direction-aware message passing module to generate node-specific context via a tri-linear model.

Scene Graph Generation. Existing SGG approaches can be roughly divided into two categories: namely, one-stage methods and two-stage methods. Generally speaking, most one-stage methods focus on object detection and relationship representation [1, 5, 10, 16, 22, 30], but almost ignore the intrinsic properties of scene graphs, *e.g.*, the edge direction and node priority. To further capture the attributes of scene graph, two-stage methods utilize an extra training stage to refine the results produced by the first stage training. For example, [24] utilizes the permutation-invariant

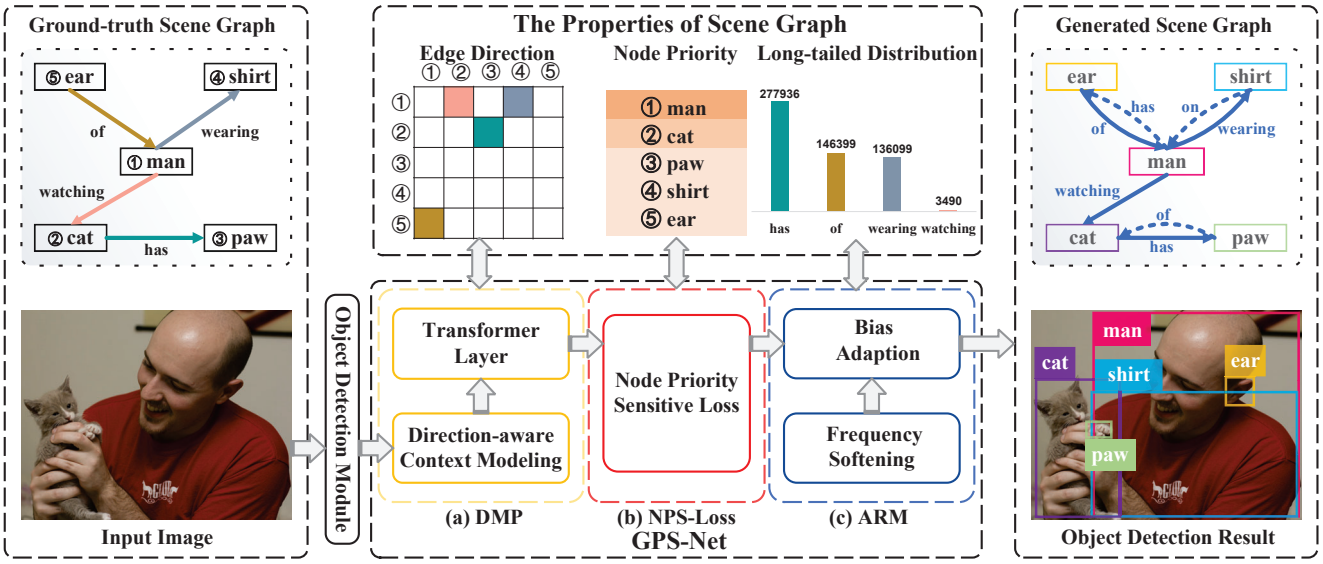


Figure 2: The framework of GPS-Net. GPS-Net adopts Faster R-CNN to obtain the location and visual feature of object proposals. It includes three new modules for SGG: (1) a novel message passing module named DMP that enhances the node feature with node-specific contextual information; (2) a new loss function named NPS-loss that reflects the difference in priority between different nodes; (3) an adaptive reasoning module (ARM) to handle the long-tailed distribution of relationships.

representations of scene graphs to refine the results of [7]. Besides, [2] utilizes dynamic tree structure to characterize the acyclic property of scene graph. Meanwhile, [11] adopts a graph-level metric to learn the node priority of scene graph. However, the adopted loss functions in [2, 11] are non-differentiable and therefore hard to optimize. The proposed approach is a one-stage method but has the following advantages comparing with existing works. First, it explores the properties of the scene graph more appropriately. Second, it is easy to optimize and deploy to existing models.

3. Approach

Figure 2 illustrates the proposed GPS-Net. We employ Faster R-CNN [25] to obtain object proposals for each image. We adopt exactly the same way as [7] to obtain the feature for each proposal. There are O object categories (including background) and R relationship categories (including non-relationship). The visual feature for the i-th proposal is formed by concatenating the appearance features $\boldsymbol{v}_i \in \mathbb{R}^{2048}$, object classification confidence scores $\boldsymbol{s}_i \in \mathbb{R}^O$, and the spatial feature $\boldsymbol{b}_i \in \mathbb{R}^4$. Then, the concatenated feature is projected into a 512-dimensional subspace and denoted as $\boldsymbol{x}_i$. Besides, we further extract features from the union box of one pair of proposal i and j, denoted as $\boldsymbol{u}_{ij} \in \mathbb{R}^{2048}$. To better capture properties of scene graph, we make contributions from three perspectives. First, a direction-aware message passing (DMP) module is introduced in Section 3.1. Second, a node priority sensitive loss (NPS-loss) is introduced in Section 3.2. Third, an adaptive reasoning module (ARM) is designed in Section 3.3.

3.1. Direction-aware Message Passing

The message passing (MP) module takes a node features $\boldsymbol{x}_i$ as input. Its output for the i-th node is denoted as $\boldsymbol{z}_i$, and the neighborhood of this node is represented as $\mathcal{N}_i$. For all MP modules in this section, $\mathcal{N}_i$ includes all nodes but the i-th node itself. Following the definition in graph attention network [8], given two nodes i and j, we represent the direction of $i \rightarrow j$ as forward and $i \leftarrow j$ as backward for the i-th node. In the following, we first review the design of the one representative MP module, which is denoted as Global Context MP (GCMP) in this paper. GCMP adopts the softmax function for normalization. Its structure is illustrated in Figure 3(a) and can be formally expressed as

$$\boldsymbol{z}_i = \boldsymbol{x}_i + \boldsymbol{W}_z \sigma\left(\sum\nolimits_{j\in\mathcal{N}_i} \frac{\exp(\boldsymbol{w}^T[\boldsymbol{x}_i,\boldsymbol{x}_j])}{\sum_{m\in\mathcal{N}_i}\exp(\boldsymbol{w}^T[\boldsymbol{x}_i,\boldsymbol{x}_m])}\boldsymbol{W}_v\boldsymbol{x}_j\right), \tag{1}$$

where σ represents the ReLU function. $\boldsymbol{W}_v$ and $\boldsymbol{W}_z \in \mathbb{R}^{512\times512}$ are linear transformation matrices. $\boldsymbol{w} \in \mathbb{R}^{1024}$ is a projection vector, and $[,]$ represents the concatenation operation. For simplicity, we define $c_{ij} = \frac{\exp(\boldsymbol{w}^T[\boldsymbol{x}_i,\boldsymbol{x}_j])}{\sum_{m\in\mathcal{N}_i}\exp(\boldsymbol{w}^T[\boldsymbol{x}_i,\boldsymbol{x}_m])}$ as the pairwise contextual coefficient between nodes i and j in the forward direction. However, it has been revealed that utilizing the concatenation operation in Equation (1) may not obtain node-specific contextual information [21]. In fact, it is more likely that $\boldsymbol{x}_i$ in Equation (1) is ignored by $\boldsymbol{w}$. Therefore, GCMP actually generates the same contextual information for all nodes.

Inspired by this observation, Equation (1) can be simpli-

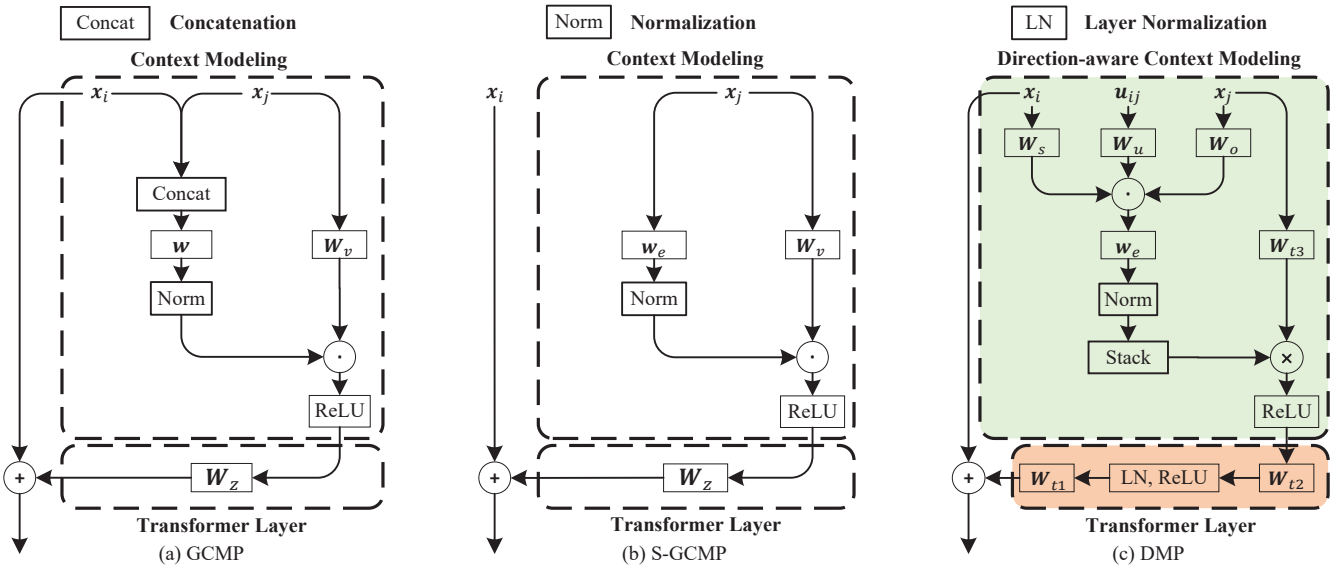


Figure 3: Architecture of the three MP modules in Section 3.1. $\odot$, $\oplus$, $\otimes$, represent Hadamard product, element-wise addition, and Kronecker product, respectively.

fied as follows [21]:

$$\boldsymbol{z}_i = \boldsymbol{x}_i + \boldsymbol{W}_z \sigma \left(\sum\nolimits_{j \in \mathcal{N}_i} \frac{\exp(\boldsymbol{w}_e^T \boldsymbol{x}_j)}{\sum_{m \in \mathcal{N}_i} \exp(\boldsymbol{w}_e^T \boldsymbol{x}_m)} \boldsymbol{W}_v \boldsymbol{x}_j \right), \tag{2}$$

where $\boldsymbol{w}_e \in \mathbb{R}^{512}$ is a projection vector. As depicted in Figure 3(b), we denote this model as Simplified Global Context MP (S-GCMP) module. The above two MP modules may not be optimal for SGG because they ignore the edge direction information and cannot provide node-specific contextual information. Accordingly, we propose the DMP module to solve the above problems. As illustrated in Figure 3(c), DMP consists of two main components: **direction-aware context modeling** and one **transformer layer**.

Direction-aware Context Modeling: This component aims to learn node-specific context and guide message passing via the edge direction information. Inspired by the multi-modal low rank bilinear pooling method [34], we formulate the contextual coefficient e_{ij} between two nodes i and j as follows:

$$e_{ij} = \boldsymbol{w}_e^T \left(\boldsymbol{W}_s \boldsymbol{x}_i \odot \boldsymbol{W}_o \boldsymbol{x}_j \odot \boldsymbol{W}_u \boldsymbol{u}_{ij} \right), \tag{3}$$

where $\odot$ represents Hadamard product. $\boldsymbol{W}_s$, $\boldsymbol{W}_o$, and $\boldsymbol{W}_u \in \mathbb{R}^{512 \times 512}$ are projection matrices for fusion. Equation (3) can be considered as a tri-linear model based on Tucker decomposition [12].

Compared with the first two MP modules, Equation (3) has four advantages. First, it employs union box features to expand the receptive field in context modeling. Second, the tri-linear model is a more powerful way to model high-order interactions between three types of features. Third, since features for the two nodes and the union box are coupled together by Hadamard product in Equation (3), they jointly affect context modeling. In this way, we obtain node-specific contextual information. Fourth, Equation (3) specifies the position of subject and object; therefore, it considers the edge direction information of the edge.

However, the direction of the edge is unclear in the MP step of SGG, since the relationship between two nodes is still unknown. Therefore, we consider the contextual coefficient for both the forward and backward directions by stacking them as a two-element-vector $[\alpha_{ij} \alpha_{ji}]^T$, where α_{ij} denotes the normalized contextual coefficient. Finally, the output of the first component of DMP for the i-th node can be denoted as

$$\sum\nolimits_{j \in \mathcal{N}_i} \begin{bmatrix} \alpha_{ij} \\ \alpha_{ji} \end{bmatrix} \otimes \boldsymbol{W}_{t3} \boldsymbol{x}_j, \tag{4}$$

where $\otimes$ denotes Kronecker product. $\boldsymbol{W}_{t3} \in \mathbb{R}^{256 \times 512}$ is a learnable projection matrix.

Transformer Layer: The contextual information obtained above may contain redundant information. Inspired by [21], we employ a transformer layer to refine the obtained contextual information. Specifically, it is consisted of two fully-connected layers with ReLU activation and layer normalization (LN) [33]. Finally, residual connection is applied to fuse the original feature and the contextual information. Our whole DMP module can be expressed as

$$\boldsymbol{z}_i = \boldsymbol{x}_i + \boldsymbol{W}_{t1} \sigma \left(\mathrm{LN} \left(\boldsymbol{W}_{t2} \sum\nolimits_{j \in \mathcal{N}_i} \begin{bmatrix} \alpha_{ij} \\ \alpha_{ji} \end{bmatrix} \otimes \boldsymbol{W}_{t3} \boldsymbol{x}_j \right) \right), \tag{5}$$

where $\boldsymbol{W}_{t1} \in \mathbb{R}^{512 \times 128}$ and $\boldsymbol{W}_{t2} \in \mathbb{R}^{128 \times 512}$ denote linear transformation matrices.

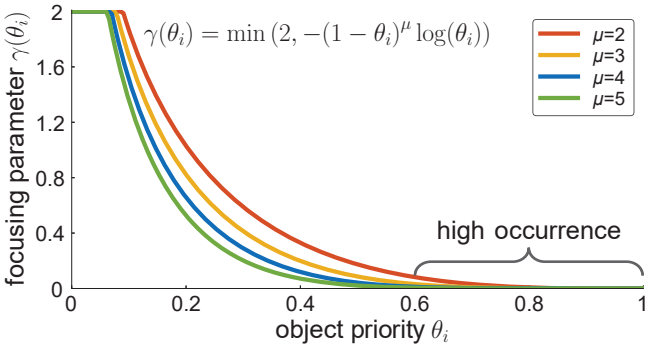


Figure 4: Mapping function $\gamma(\theta_i)$ with different controlling factors μ.

3.2. Node Priority Sensitive Loss

Existing works for SGG tend to utilize cross-entropy loss as objective function for object classification, which implicitly regards the priority of all nodes is equal for the scene graph. However, their priority varies according to the number of triplets they are involved. Recently, a local-sensitive loss has been proposed to address this problem in [11]. As the loss is non-differentiable, the authors in [11] adopt a two-stage training strategy, where the second stage is realized by a complicated policy gradient method [46].

To handle this problem, we propose a novel NPS-loss that not only captures the node priority in scene graph but also has the benefit of differentiable and convex formulation. NPS-loss is inspired by focal loss that reduces weights of well-classified objects using a focusing parameter, which is denoted as γ in this paper. Compared with focal loss, NP-loss has the following key differences: (1) it is mainly used to solve the node-priority problem in SGG. In comparison, focal loss is designed to solve the class imbalance problem in object detection; (2) γ is fixed in [13]. In NPS-loss, it depends on the node priority. Specifically, we first calculate the priority θ_i for the i-th node according to its contribution to the scene graph:

$$\theta_i = \frac{t_i}{\|T\|}, \tag{6}$$

where t_i denotes the number of triplets that include the i-th node and $\|T\|$ is the total number of triplets in one graph. Given θ_i, one intuitive way to obtain the focusing parameter γ is a linear transformation, *e.g.*, $\gamma(\theta_i) = -2\theta_i + 2$. However, this transformation exaggerates the difference between nodes of high-priority and middle-level priority, and narrows the difference between nodes of middle-level priority and low-priority. To solve this problem, we design a nonlinear mapping function that transforms θ_i to γ:

$$\gamma(\theta_i) = \min\left(2, -(1-\theta_i)^{\mu}\log(\theta_i)\right), \tag{7}$$

where μ denotes a controlling factor, which controls the influence of θ_i to the value of γ. As depicted in Figure 4, curve for the mapping function changes quickly for nodes with low priority, and slowly for nodes of high priority. Moreover, a larger μ leads to more nodes to be highlighted during training. Finally, we obtain the NPS-loss that guides the training process according to node priority:

$$\mathcal{L}_{nps}(p_i) = -(1-p_i)^{\gamma(\theta_i)}\log(p_i), \tag{8}$$

where p_i denotes the object classification score on the ground-truth object class for the i-th node.

3.3. Adaptive Reasoning Module

After obtaining the refined node features by DMP and the object classification scores by NPS-loss, we further propose an adaptive reasoning module (ARM) for relationship classification. Specifically, ARM provides prior for classification by two steps: frequency softening and bias adaptation for each triplet. In what follows, we introduce the two steps in detail.

Frequency Softening: Inspired by the frequency baseline introduced in [7], we employ the frequency of relationships as prior to promote the performance of relationship classification. However, the original method in [7] suffers from the long-tailed distribution problem of relationships. Therefore, it may fail to recognize relationships of low frequency. To handle this problem, we first adopt a log-softmax function to soften the original frequency distribution of relationships as follows:

$$\tilde{\boldsymbol{p}}^{i\to j} = \log\mathrm{softmax}\left(\boldsymbol{p}^{i\to j}\right), \tag{9}$$

where $\boldsymbol{p}^{i\to j} \in \mathbb{R}^R$ denotes the original frequency distribution vector between the i-th and the j-th nodes. The same as [7], this vector is determined by the object class of the two nodes. $\tilde{\boldsymbol{p}}^{i\to j}$ is the normalized vector of $\boldsymbol{p}^{i\to j}$.

Bias Adaptation: To enable the frequency prior adjustable for each node pair, we further propose an adpative attention mechanism to modify the prior according to the visual appearance of the node pair. Specifically, a sigmoid function is applied to obtain attention on the frequency prior: $\boldsymbol{d} = \mathrm{sigmoid}\left(\boldsymbol{W}_p\boldsymbol{u}_{ij}\right)$, where $\boldsymbol{W}_p \in \mathbb{R}^{R\times 2048}$ is transformation matrix. Then, the classification score vector of relationships can be obtained as follows:

$$\boldsymbol{p}_{ij} = \mathrm{softmax}\left(\boldsymbol{W}_r(\boldsymbol{z}_i * \boldsymbol{z}_j * \boldsymbol{u}_{ij}) + \boldsymbol{d}\odot\tilde{\boldsymbol{p}}^{i\to j}\right), \tag{10}$$

where $\boldsymbol{W}_r \in \mathbb{R}^{R\times 1024}$ denotes the classifier, and $\boldsymbol{d}\odot\tilde{\boldsymbol{p}}^{i\to j}$ is the bias. $*$ represents a fusion function defined in [47]: $\boldsymbol{x} * \boldsymbol{y} = \mathrm{ReLU}\left(\boldsymbol{W}_x\boldsymbol{x} + \boldsymbol{W}_y\boldsymbol{y}\right) - \left(\boldsymbol{W}_x\boldsymbol{x} - \boldsymbol{W}_y\boldsymbol{y}\right)\odot\left(\boldsymbol{W}_x\boldsymbol{x} - \boldsymbol{W}_y\boldsymbol{y}\right)$, where $\boldsymbol{W}_x$ and $\boldsymbol{W}_y$ project $\boldsymbol{x}, \boldsymbol{y}$ to 1024-dimensional space, respectively.

Relationship Prediction: During testing, the category of relationship between i-th and j-th nodes is predicted by:

$$r_{ij} = \arg\max_{r\in\mathcal{R}}(\boldsymbol{p}_{ij}(r)), \tag{11}$$

where $\mathcal{R}$ represents the set of relationship categories.

	Model	SGDET			SGCLS			PREDCLS			Mean
		R@20	R@50	R@100	R@20	R@50	R@100	R@20	R@50	R@100	
Two-Stage	GPI$^{\diamond}$ [24]	-	-	-	-	36.5	38.8	-	65.1	66.9	-
	VCTREE-HL$^{\diamond}$ [2]	22.0	27.9	31.3	35.2	38.1	38.8	60.1	66.4	68.1	45.1
	CMAT$^{\diamond}$ [11]	22.1	27.9	31.2	35.9	39.0	39.8	60.2	66.4	68.1	45.4
One-Stage	IMP$^{\diamond}$ [5]	14.6	20.7	24.5	31.7	34.6	35.4	52.7	59.3	61.3	39.3
	FREQ$^{\diamond}$[7]	20.1	26.2	30.1	29.3	32.3	32.9	53.6	60.6	62.2	40.7
	MOTIFS$^{\diamond}$ [7]	21.4	27.2	30.3	32.9	35.8	36.5	58.5	65.2	67.1	43.7
	Graph-RCNN [22]	-	11.4	13.7	-	29.6	31.6	-	54.2	59.1	33.2
	KERN$^{\diamond}$ [23]	-	27.1	29.8	-	36.7	37.4	-	65.8	67.6	44.1
	VCTREE-SL$^{\diamond}$ [2]	21.7	27.7	31.1	35.0	37.9	38.6	59.8	66.2	67.9	44.9
	CMAT-XE$^{\diamond}$ [11]	-	-	-	34.0	36.9	37.6	-	-	-	-
	RelDN‡ [6]	21.1	28.3	32.7	36.1	36.8	36.8	66.9	68.4	68.4	45.2
	GPS-Net$^{\diamond}$	**22.6**	**28.4**	**31.7**	**36.1**	**39.2**	**40.1**	**60.7**	**66.9**	**68.8**	**45.9**
	GPS-Net‡	**22.3**	**28.9**	**33.2**	**41.8**	**42.3**	**42.3**	**67.6**	**69.7**	**69.7**	**47.7**

Table 1: Comparisons with state-of-the-arts on VG. Since some works do not evaluate on R@20, we compute the mean on all tasks over R@50 and R@100. $^{\diamond}$ and ‡ denote the methods using the same Faster-RCNN detector and evaluation metric as [7] and [6], respectively.

Model	SGDET mR@100	SGCLS mR@100	PREDCLS mR@100
IMP$^{\diamond}$ [5]	4.8	6.0	10.5
FREQ$^{\diamond}$ [7]	7.1	8.5	16.0
MOTIFS$^{\diamond}$ [7]	6.6	8.2	15.3
KERN$^{\diamond}$ [23]	7.3	10.0	19.2
VCTREE-HL$^{\diamond}$ [2]	8.0	10.8	19.4
GPS-Net$^{\diamond}$	**9.8**	**12.6**	**22.8**

Table 2: Comparison on the mR@100 metric between various methods across all the 50 relationship categories.

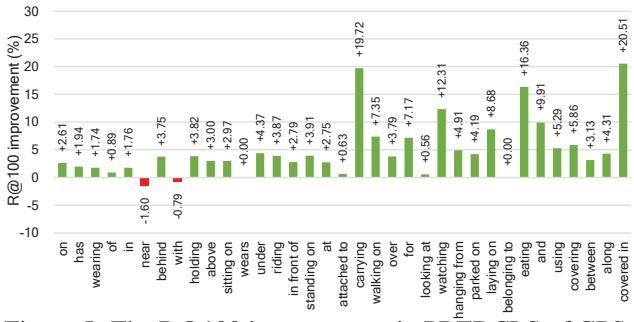


Figure 5: The R@100 improvement in PREDCLS of GPS-Net compared with the VCTREE [2]. The Top-35 categories of relationship are selected according to their occurrence frequency.

4. Experiments

We present experimental results on three datasets: Visual Genome (VG) [14], OpenImages (OI) [15], and Visual Relationship Detection (VRD) [16]. We first report evaluation settings, followed by comparisons with state-of-the-art methods and the ablation studies. Besides, qualitative comparisons between GPS-Net and other approaches are provided in the supplementary file.

4.1. Evaluation Settings

Visual Genome: We use the same data and evaluation metrics that have been widely adopted in recent works [22, 10, 1, 24, 30, 11]. Specifically, the most frequent 150 object categories and 50 relationship categories are utilized for evaluation. After preprocessing, the scene graph for each image consists of 11.6 objects and 6.2 relationships on average. The data is divided into one training set and one testing set. The training set includes 70% images, with 5K images as a validation subset. The testing set is composed of the remaining 30% images. In the interests of fair comparisons, we also adopt Faster R-CNN [25] with VGG-16 backbone to obtain the location and features of object proposals. Moreover, since SGG performance highly depends on the pre-trained object detector, we utilize the same set of hyper-parameters as [7] and [6] respectively. We follow three conventional protocols for evaluation: (1) Scene Graph Detection (SGDET): given an image, detect object bounding boxes and their categories, and predict their pair-wise relationships; (2) Scene Graph Classification (SGCLS): given ground-truth object bounding boxes, predict the object categories and their pair-wise relationships; (3) Predicate Classification (PREDCLS): given the object categories and their bounding boxes, predict their pair-wise relationships only. All algorithms are evaluated by Recall@K metrics, where K=20, 50, and 100, respectively. Considering that the distribution of relationships is highly imbalanced in VG, we further utilize mean recall@K (mR@K) to evaluate the performance of each relationship [2, 23].

OpenImages: The training and testing sets contain 53,953 images and 3,234 images respectively. We utilize Faster R-CNN associated with the pre-trained ResNeXt-101-FPN [6] as the backbone. We also follow the same data

Model	R@50	wmAP_{rel}	wmAP_{phr}	score_{wtd}	AP_{rel} per class								
					at	on	holds	plays	interacts with	wears	hits	inside of	under
RelDN, L_0[6]	74.67	34.63	37.89	43.94	32.40	36.51	41.84	36.04	40.43	5.70	55.40	44.17	25.00
RelDN[6]	74.94	35.54	38.52	44.61	32.90	37.00	43.09	41.04	44.16	7.83	51.04	44.72	50.00
GPS-Net	**77.27**	**38.78**	**40.15**	**47.03**	**35.10**	**38.90**	**51.47**	**45.66**	**44.58**	**32.35**	**71.71**	**47.21**	**57.28**

Table 3: Comparisons with state-of-the-arts on OI. We adopt the same evaluation metrics as [6]

Model	Pre.	Rel.		Phr.	
	R@50	R@50	R@100	R@50	R@100
VTransE [37]	44.8	19.4	22.4	14.1	15.2
ViP-CNN [39]	-	17.3	20.0	22.8	27.9
VRL [40]	-	18.2	20.8	21.4	22.6
KL distilation [43]	55.2	19.2	21.3	23.1	24.0
MF-URLN [44]	58.2	23.9	26.8	31.5	36.1
Zoom-Net*[42]	50.7	18.9	21.4	24.8	28.1
CAI + SCA-M*[42]	56.0	19.5	22.4	25.2	28.9
GPS-Net* (ImageNet)	**58.7**	**21.5**	**24.3**	**28.9**	**34.0**
RelDN† [6]	-	25.3	28.6	31.3	36.4
GPS-Net† (COCO)	**63.4**	**27.8**	**31.7**	**33.8**	**39.2**

Table 4: Comparisons with state-of-the-arts on VRD (− denotes unavailable). Pre., Phr., and Rel. represent predication detection, phrase detection, and relation detection, respectively. † and * denote using the same object detector.

processing and evaluation metrics as in [6]. More specifically, the results are evaluated by calculating Recall@50 (R@50), weighted mean AP of relationships (wmAP_{rel}), and weighted mean AP of phrase (wmAP_{phr}). The final score is given by $\mathrm{score}_{wtd} = 0.2 \times R@50 + 0.4 \times \mathrm{wmAP}_{rel} + 0.4 \times \mathrm{wmAP}_{phr}$. Note that the wmAP_{rel} evaluates the AP of the predicted triplet where both the subject and object boxes have an IoU of at least 0.5 with ground truth. The wmAP_{phr} is similar, but utilized for the union area of the subject and object boxes.

Visual Relationship Detection: We apply the same object detectors as in [6]. More specifically, two VGG16-based backbones are provided, which were trained on ImgaeNet and COCO, respectively. The evaluation metric is the same as in [16], which reports R@50 and R@100 for relationship, predicate, and phrase detection.

4.2. Implementation Details

To ensure compatibility with the architectures of previous state-of-the-art methods, we utilize ResNeXt-101-FPN as our OpenImages backbone on OI and VGG-16 on VG and VRD. During training, we freeze the layers before the ROIAlign layer and optimize the model jointly considering the object and relationship classification losses. Our model is optimized by SGD with momentum, with the initial learning rate and batch size set to 10^{-3} and 6 respectively. For the SGDET task, we follow [7] that we only predict the relationship between proposal pairs with overlapped bounding boxes. Besides, the top-64 object proposals in each image are selected after per-class non-maximal suppression (NMS) with an IoU of 0.3. Moreover, the ratio between pairs without any relationship (background pairs) and those with relationship during training is sampled to 3:1.

4.3. Comparisons with State-of-the-Art Methods

Visual Genome: Table 1 shows that GPS-Net outperforms all state-of-the-arts methods on various metrics. Specifically, GPS-Net outperforms one very recent one-stage model, named KERN [23], by 1.8% on average at R@50 and R@100 over the three protocols. In more detail, it outperforms KERN by 1.9%, 2.7% and 1.2% at R@100 on SGDET, SGCLS, and PRECLS, respectively. Even when compared with the best two-stage model CMAT [11], GPS-Net still demonstrates a performance improvement of 0.5% on average over the three protocols. Meanwhile, compared with the one-stage version of VCTREE [2] and CMAT [11], GPS-Net respectively achieves 1.5% and 2.5% performance gains on SGCLS at Recall@100. Another advantage of GPS-Net over VCTREE and CMAT is that GPS-Net is much more efficient, as the two methods adopt policy gradient for optimization, which is time-consuming [46]. Moreover, when compare with RelDN using the same backbone, the performance gain by GPS-Net is even more dramatic, namely, 5.5% promotion on SGCLS at Recall@100 and 2.5% on average over three protocols.

Due to the class imbalance problem in VG, previous works usually achieve low performance for less frequent categories. Hence, we conduct an experiment utilizing the Mean Recall as evaluation metric [23, 2]. As shown in Table 2 and Figure 5, GPS-Net shows a large absolute gain for both the Mean Recall and Recall metrics, which indicates that GPS-Net has advantages in handling the class imbalance problem of SGG.

OpenImages: We present results compared with RelDN [6] in Table 3. RelDN is an improved version of the model that won the Google OpenImages Visual Relationship Detection Challenge, with the same object detector, GPS-Net outperforms RelDN by 2.4% on the overall metric $score_{wtd}$. Moreover, despite of the severe class imbalance problem, GPS-Net still achieves outstanding performance in AP_{rel} for each category of relationships. The largest gap between GPS-Net and RelDN in AP_{rel} is 24.5% for *wears* and 20.6% for *hits*.

Visual Relationship Detection: Table 4 presents comparisons on VRD with state-of-the-art methods. To facil-

Exp	Module DMP	NPS	ARM	SGDET R@20	R@50	R@100	SGCLS R@20	R@50	R@100	PREDCLS R@20	R@50	R@100
1				21.1	26.3	29.4	32.7	35.4	36.3	58.8	65.6	67.3
2	✓			22.3	28.1	31.4	35.2	38.3	39.3	59.6	66.1	67.9
3		✓		21.5	26.6	29.8	33.2	36.3	37.1	59.1	65.9	67.7
4			✓	21.3	26.5	29.6	32.9	35.8	36.8	60.5	66.7	68.5
5	✓	✓	✓	**22.6**	**28.4**	**31.7**	**36.1**	**39.2**	**40.1**	**60.7**	**66.9**	**68.8**

Table 5: Ablation studies on the proposed methods. We consistently use the same backbone as [7].

		w. stack	w.o. stack
SGCLS	R@20	35.7	**36.1**
	R@50	38.8	**39.2**
	R@100	39.6	**40.1**
SGDET	R@20	22.4	**22.6**
	R@50	28.3	**28.4**
	R@100	31.5	**31.7**

		GCMP	S-GCMP	DMP
SGCLS	R@20	34.3	34.8	**36.1**
	R@50	37.2	37.7	**39.2**
	R@100	37.9	38.4	**40.1**
SGDET	R@20	21.7	22.1	**22.6**
	R@50	27.5	28.0	**28.4**
	R@100	30.8	31.2	**31.7**

		Focal	$\mu = 3$	$\mu = 4$	$\mu = 5$
SGCLS	R@20	35.8	36.0	**36.1**	35.8
	R@50	39.0	38.9	**39.2**	39.1
	R@100	39.8	39.9	**40.1**	39.9
SGDET	R@20	22.4	22.5	**22.6**	22.5
	R@50	28.2	28.2	**28.4**	28.3
	R@100	31.5	31.6	**31.7**	31.6

Table 6: The **left sub-table** shows the effectiveness of the stacking operation in DMP. The **middle sub-table** compares the performance of the three MP modules in Section 3.1 with the same transformer layer. The **right sub-table** compares NPS-loss and the focal loss, and shows the influence of the controlling factor μ.

itate fair comparison, we adopt the two backbone models provided in RelDN [6] to train GPS-Net, respectively. It is shown that GPS-Net consistently achieves superior performance with both backbone models.

4.4. Ablation Studies

To prove the effectiveness of our proposed methods, we conduct four ablation studies. Results of the ablation studies are summarized in Table 5 and Table 6, respectively.

Effectiveness of the Proposed Modules. We first perform an ablation study to validate the effectiveness of DMP, NPS-loss, and ARM. Results are summarized in Table 5. We add the above modules one by one to the baseline model. In Table 5, Exp 1 demotes our baseline that is based on the MOTIFNET-NOCONTEXT method [7] with our feature construction strategy for relationship prediction. From Exp 2-5, we can clearly see that the performance improves consistently when all the modules are used together. This shows that each module plays a critical role in inferring object labels and their pair-wise relationships.

Effectiveness of the Stacking Operation in DMP. We conduct additional analysis on the stacking operation in DMP. The stacking operation accounts for the uncertainty in the edge direction information. As shown in the left sub-table of Table 6, the stacking operation consistently improves the performance of DMP over various metrics. Therefore, its effectiveness is justified.

Comparisons between Three MP Modules. We compare the performance of three MP modules in Section 3.1: GCMP, S-GCMP, and DMP. To facilitate fair comparison, we implement the same transformer layer as DMP to the other two modules. As shown in the middle sub-table in Table 6, the performance of DMP is much better than the other two modules. This is because DMP encodes the edge direction information and provides node-specific contextual information for each node involved in message passing.

Design Choices in NPS-loss. The value of the controlling factor μ determines the impact of node priority on object classification. As shown in the right sub-table of Table 6, we show the performance of NPS-loss with three different values of μ. We also compare NPS-loss with the focal loss [13]. NPS-loss achieves the best performance when μ equals to 4. Moreover, NPS-loss outperforms the focal loss, justifying its effectiveness to solve the node priority problem for SGG.

5. Conclusion

In this paper, we devise GPS-Net to address the main challenges in SGG by capturing three key properties of scene graph. Specifically, (1) edge direction is encoded when calculating the node-specific contextual information via the DMP module; (2) the difference in node priority is characterized by a novel NPS-loss; and (3) the long-tailed distribution of relationships is alleviated by improving the usage of relationship frequency through ARM. Through extensive comparative experiments and ablation studies, we validate the effectiveness of GPS-Net on three datasets.

Acknowledgment. Changxing Ding was supported in part by NSF of China under Grant 61702193, in part by the Science and Technology Program of Guangzhou under Grant 201804010272, in part by the Program for Guangdong Introducing Innovative and Entrepreneurial Teams under Grant 2017ZT07X183. Dacheng Tao was supported in part by ARC FL-170100117 and DP-180103424.

References

[1] Y. Li, W. Ouyang, B. Zhou, K. Wang, and X. Wang. Scene graph generation from objects, phrases and region captions. In *ICCV*, 2017.

[2] K. Tang, H. Zhang, B. Wu, W. Luo, and W. Liu. Learning to compose dynamic tree structures for visual contexts. In *CVPR*, 2019.

[3] S. Qi, Y. Zhu, S. Huang, C. Jiang, S. Zhu. Human-centric Indoor Scene Synthesis Using Stochastic Grammar. In *ICLR*, 2018.

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

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

[6] J. Zhang, K. Shih, A. Elgammal, A. Tao, and B. Catanzaro. Graphical contrastive losses for scene graph parsing. In *CVPR*, 2019.

[7] R. Zellers, M. Yatskar, S. Thomson, and Y. Choi. Neural motifs: Scene graph parsing with global context. In *CVPR*, 2018.

[8] P. Velickovic, G. Cucurull, A. Casanova, and A. Romero. Graph Attention Networks. In *ICLR*, 2018.

[9] L. Gong and Q. Cheng. Exploiting Edge Features in Graph Neural Networks. In *CVPR*, 2019.

[10] M. Qi, W. Li, Z. Yang, Y. Wang and J. Luo. Attentive Relational Networks for Mapping Images to Scene Graphs. In *CVPR*, 2019.

[11] L. Chen, H. Zhang, J. Xiao, X. He, S. Pu, and S. F. Chang, Counterfactual Critic Multi-Agent Training for Scene Graph Generation. In *CVPR*, 2019.

[12] H. Ben-Younes, R. Cadene, M. Cord, and N. Thome. Mutan: Multimodal tucker fusion for visual question answering. In *CVPR*, 2017.

[13] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar. Focal loss for dense object detection. In *ICCV*, 2017.

[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. In *IJCV*, 2017.

[15] A. Kuznetsova, H. Rom, N. Alldrin, J. Uijlings, I. Krasin, J. Tuset, S. Kamali, S. Popov, M. Malloci, T. Duerig, et al. The open imagesdataset v4: Unified image classification, object detection, and visual relationship detection at scale. In *arXiv:1811.00982*, 2018.

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

[17] J. Hu, L. Shen, and G. Sun. Squeeze-and-excitation networks. In *CVPR*, 2018.

[18] H. Zhao, Y. Zhang, S. Liu, J. Shi, C. Change Loy, D. Lin, and J. Jia. Psanet: Point-wise spatial attention network for scene parsing. In *ECCV*, 2018.

[19] X.Wang, R. Girshick, A. Gupta, and K. He. Non-local neural networks. In *CVPR*, 2018.

[20] Z. Huang, X. Wang, L. Huang, C. Huang, Y. Wei, and W. Liu. Ccnet: Criss-cross attention for semantic segmentation. In *arXiv preprint arXiv:1811.11721*, 2018.

[21] Y. Cao and J. Xu. GCNet: Non-local Networks Meet Squeeze-Excitation Networks and Beyond. In *arXiv preprint arXiv:1904.11492*, 2019.

[22] J. Yang, J. Lu, S. Lee, D. Batra, and D. Parikh. Graph r-cnn for scene graph generation. In *ECCV*, 2018.

[23] T. Chen, W. Yu, R. Chen, and L. Lin. Knowledge-embedded routing network for scene graph generation. In *CVPR*, 2019.

[24] R. Herzig, M. Raboh, G. Chechik, J. Berant, and A. Globerson. Mapping images to scene graphs with permutation-invariant structured prediction. In *NeurIPS*, 2018.

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

[26] A. Newell and J. Deng. Pixels to graphs by associative embedding. In *NIPS*, 2017.

[27] S. Hwang, S. Ravi, Z. Tao, H. Kim, M. Collins, and V. Singh. Tensorize, factorize and regularize: Robust visual relationship learning. In *CVPR*, 2018.

[28] J. Zhang, Y. Kalantidis, M. Rohrbach, M. Paluri, A. Elgammal, and M. Elhoseiny. Large-scale visual relationship un- derstanding. In *AAAI*, 2019.

[29] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In *ICCV*, 2017.

[30] Y. Li, W. Ouyang, B. Zhou, Y. Cui, J. Shi, and X. Wang. Factorizable net: An efficient subgraph-based framework for scene graph generation. In *ECCV*, 2018.

[31] Y. Li, W. Ouyang, B. Zhou, K. Wang, and X. Wang. Scene graph generation from objects, phrases and region captions. In *ICCV*, 2017.

[32] L. Gong, and Q. Cheng. Exploiting Edge Features for Graph Neural Networks. In *CVPR*, 2019.

[33] J. Ba, J. R. Kiros, and G. E. Hinton, Layer normalization. In *arXiv preprint arXiv:1607.06450*, 2016.

[34] J. Kim, K. On, W. Lim, J. Kim, J. Ha, and B. Zhang. Hadamard product for low-rank bilinear pooling. In *ICLR*, 2017.

[35] H. Zhang, Z. Kyaw, J. Yu, and S.-F. Chang. Ppr-fcn: Weakly supervised visual relation detection via parallel pairwise r-fcn. In *CVPR*, 2017.

[36] J. Peyre, I. Laptev, C. Schmid, and J. Sivic. Weakly-supervised learning of visual relations. In *ICCV*, 2017.

[37] H. Zhang, Z. Kyaw, S. Chang, and T. Chua. Visual translation embedding network for visual relation detection. In *CVPR*, 2017.

[38] B. Dai, Y. Zhang, and D. Lin. Detecting visual relationships with deep relational networks. In *CVPR*, 2017.

[39] Y. Li, W. Ouyang, and X. Wang. Vip-cnn: A visual phrase reasoning convolutional neural network for visual relationship detection. In *CVPR*, 2017.

[40] X. Liang, L. Lee, and E. P. Xing. Deep variation-structured reinforcement learning for visual relationship and attribute detection. In *CVPR*, 2017.

[41] B. Zhuang, L. Liu, C. Shen, and I. Reid. Towards context-aware interaction recognition for visual relationship detec- tion. In *ICCV*, 2017.

[42] G. Yin, L. Sheng, B. Liu, N. Yu, X. Wang, J. Shao, and C. Change Loy. Zoom-net: Mining deep feature interactions for visual relationship recognition. In *ECCV*, 2018.

[43] R. Yu, A. Li, V. I. Morariu, and L. S. Davis. Visual relationship detection with internal and external linguistic knowl- edge distillation. In *ICCV*, 2017.

[44] Y. Zhan, J. Yu, T. Yu, D. Tao. On Exploring Undetermined Relationships for Visual Relationship Detection. In *CVPR*, 2019.

[45] T. N. Kipf and M. Welling. Semi-Supervised Classification with Graph Convolutional Networks. In *ICLR*, 2017.

[46] R. Houthooft, R. Y. Chen, P. Isola, B. C. Stadie, F. Wolski, J. Ho, and P. Abbeel. Evolved policy gradients. In *NeurIPS*, 2018.

[47] Y. Zhang, J. Hare, and A. Prugel-Bennett. Learning to count objects in natural images for visual question answering. In *ICLR*, 2018.