RU-Net: Regularized Unrolling Network for Scene Graph Generation

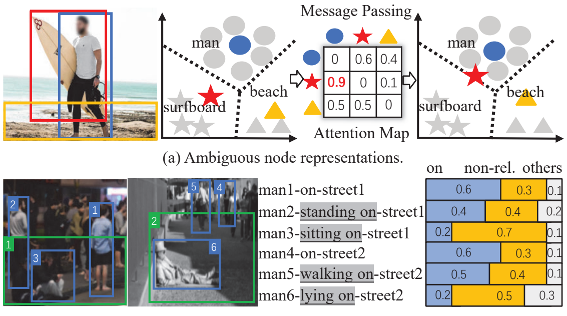
Xin Lin[1]* Changxing Ding[1,2]† Jing Zhang[3] Yibing Zhan[4] Dacheng Tao[4,3]
[1] South China University of Technology [2] Pazhou Lab, Guangzhou [3] The University of Sydney
[4] JD Explore Academy
eelinxin@mail.scut.edu.cn, chxding@scut.edu.cn, jing.zhang1@sydney.edu.au,
zhanyibing@jd.com, dacheng.tao@gmail.com

Abstract

Scene graph generation (SGG) aims to detect objects and predict the relationships between each pair of objects. Existing SGG methods usually suffer from several issues, including 1) ambiguous object representations, as graph neural network-based message passing (GMP) modules are typically sensitive to spurious inter-node correlations, and 2) low diversity in relationship predictions due to severe class imbalance and a large number of missing annotations. To address both problems, in this paper, we propose a regularized unrolling network (RU-Net). We first study the relation between GMP and graph Laplacian denoising (GLD) from the perspective of the unrolling technique, determining that GMP can be formulated as a solver for GLD. Based on this observation, we propose an unrolled message passing module and introduce an ℓ_p-based graph regularization to suppress spurious connections between nodes. Second, we propose a group diversity enhancement module that promotes the prediction diversity of relationships via rank maximization. Systematic experiments demonstrate that RU-Net is effective under a variety of settings and metrics. Furthermore, RU-Net achieves new state-of-the-arts on three popular databases: VG, VRD, and OI. Code is available at https://github.com/siml3/RU-Net.

1. Introduction

Scene Graph Generation (SGG) aims to provide a graphical representation of objects and their relationships in an image. Recently, SGG has emerged as a promising approach that bridges the gap between vision and natural language domains. It has been found to be useful for many vision tasks, including 3D scene understanding [2, 40], visual question answering [8, 32], and image captioning [12, 58].



(a) Ambiguous node representations.

(b) Lack of diversity in relationship prediction.

Figure 1. (a) Spurious correlation between nodes causes ambiguous representations through graph neural network-based message passing. (b) Relationship prediction for the same category of node pairs lacks diversity. Missing relationship annotations are underlined and highlighted in gray. Best viewed in color.

A scene graph comprises a collection of triplets in the form *subject-relationship-object*. The objects and their pairwise relationships are denoted as nodes and edges, respectively. Existing SGG models [6, 19, 22, 39, 46, 47, 60] typically utilize context modeling strategies to learn discriminative representations for node and edge prediction; specifically, most of them adopt graph neural network-based message passing (GMP) mechanisms. In GMP, node representations are iteratively updated through the aggregation of neighboring information according to learnable attention weights, which are typically supervised by node labels.

However, current GMPs are negatively impacted by spurious correlations between nodes. Here, a spurious correlation refers to a relatively large attention weight between a pair of semantically disparate nodes. These spurious correlations frequently occur, as attention weights between spatially proximate nodes tend to be large regardless of whether their object categories are related. In Figure 1(a), it is evident that the attention weights for the *surfboard* are dominated by those for the *man* (*i.e.*, equal to 0.9). As a result, the quality of representations for some nodes may degrade after erroneous message passing. Moreover, relationship prediction diversity among existing SGG models tends to be low. This is mainly due to the long-tailed distribution of relationships and a large number of missing relationship

*Work done during first author's internship at JD Explore Academy.
†Corresponding author.

annotations. As shown in Figure 1(b), the two images contain six triplets related to the *man-street* pair; however, only two of them are annotated, and the relationship categories are both *on*. The trained SGG models, therefore, tend to make biased predictions for the majority classes and the *non-relationship* category.

To address the above issues, we propose a regularized unrolling network (RU-Net) for SGG. First, we study the relation between GMP and graph Laplacian denoising (GLD) [33] from the perspective of the unrolling technique [29]. We show that 1) GMP can be formulated as the solver for GLD, and 2) the quadratic penalty widely adopted in the formulation of GLD is sensitive to outliers (*e.g.*, spurious correlations between nodes). As an alternative, we propose an unrolled message passing (U-MP) module and employ an ℓ_p-based graph regularization term to suppress these spurious connections between nodes, thereby effectively reducing the ambiguity in node representations. Moreover, we determine that the optimization of the ℓ_p-based graph regularization can be efficiently achieved in an end-to-end manner by integrating a reweighting matrix into U-MP, which accounts for the semantic dissimilarity between nodes.

Second, we introduce a group diversity enhancement (GDE) module to promote the diversity of relationship predictions for both labeled and unlabeled samples. More specifically, since score vectors for relationships tend to be linearly independent when predicted as different categories, we formulate the optimization of relationship prediction diversity as a rank maximization problem. Because rank maximization is NP-hard [36], we use the $\ell_{2,1}$-norm to approximate the matrix rank. We also divide the large matrix into several smaller ones, each of which contains relationship predictions for node pairs of the same object categories. By enlarging the $\ell_{2,1}$-norm of the smaller matrices, the relationship prediction diversity is more effectively optimized, as demonstrated in Section 4.3.

In summary, the contributions of this study are three fold: (1) a novel unrolling framework that interprets GMP as a solver for the GLD problem; (2) the U-MP module for spuriousness-robust message passing via an ℓ_p-based graph regularization, which enhances GMP's robustness against spurious connections between nodes; and (3) the GDE module, which improves the diversity of relationship prediction via the group-wise $\ell_{2,1}$-based regularization term. The efficacy of the proposed RU-Net is systematically evaluated on three popular SGG databases: Visual Genome (VG) [16], OpenImages (OI) [17], and Visual Relationship Detection (VRD) [25]. Experimental results show that our RU-Net consistently outperforms state-of-the-art methods.

2. Related Work

Scene Graph Generation. Existing works in SGG [6, 7, 10, 14, 47, 57, 60] generally focus on context modeling or tackling the class imbalance problem (*i.e.*, the long-tailed distribution). Several context modeling strategies have been proposed to learn discriminative object representation by exploring various message passing mechanisms. Zeller *et al.* [52] represented the global context via a recurrent sequential architecture (*i.e.*, bidirectional long short-term memory (Bi-LSTM) model). Tang *et al.* [39] utilized dynamic tree structures to realize node-specific message passing. Lin *et al.* [22] proposed a direction-aware message passing module that encodes the edge direction information into context modeling. Li *et al.* [19] adopted a relationship prediction confidence-based adaptive message passing strategy to reduce noise in context modeling. Lu *et al.* [26] utilized the transformer encoder to acquire contextual information pertaining to both objects and context. To handle the class imbalance issue, Tang *et al.* [38] proposed an unbiased model that removes the vision-agnostic bias with counterfactual causality, while [4, 7] addressed this problem using positive-unlabeled learning. Some works have additionally explored class imbalance learning strategies [19, 45], re-sampling and cost-sensitive learning, to relieve the long-tailed distribution problem. Our work considers both issues discussed above in a unified framework.

Deep Algorithm Unrolling. In deep algorithm unrolling (DAU), the structure of the model-based iterative optimization algorithms is unrolled into a neural network [11,27,29]. Specifically, each iteration of the algorithm is represented as one layer of a network. Stacking these layers forms a deep neural network with an architectural structure that depends on the optimization method employed. The forward propagation of the network is equivalent to executing the iterative algorithm several times. Compared with fully parameterized neural networks, DAU is advantageous in terms of its interpretability and model complexity [5, 20, 28]; hence, DAU-based networks can be effectively optimized with less training data. For example, Yang *et al.* [49] proposed an unrolled version of the Alternating Direction Method of Multipliers [48] for magnetic resonance imaging. Zhang *et al.* [56] integrated convolutional networks with the iterative shrinkage-thresholding algorithm [3] for compressed image sensing. Moreover, the half-quadratic splitting algorithm [1] has been used in [9, 59] to unfold the minimization problems for image denoising and super-resolution. Inspired by these works, we introduce DAU to the SGG and unify existing GMP modules to solve GLD.

3. Regularized Unrolling Network

This section presents the details of the proposed regularized unrolling network. More specifically, we first introduce the preliminaries, then explain the network details and the training losses. As Figure 2 illustrates, RU-Net comprises a U-MP module and a GDE module. From the perspective of DAU, the U-MP module utilizes ℓ_p-based graph regular-

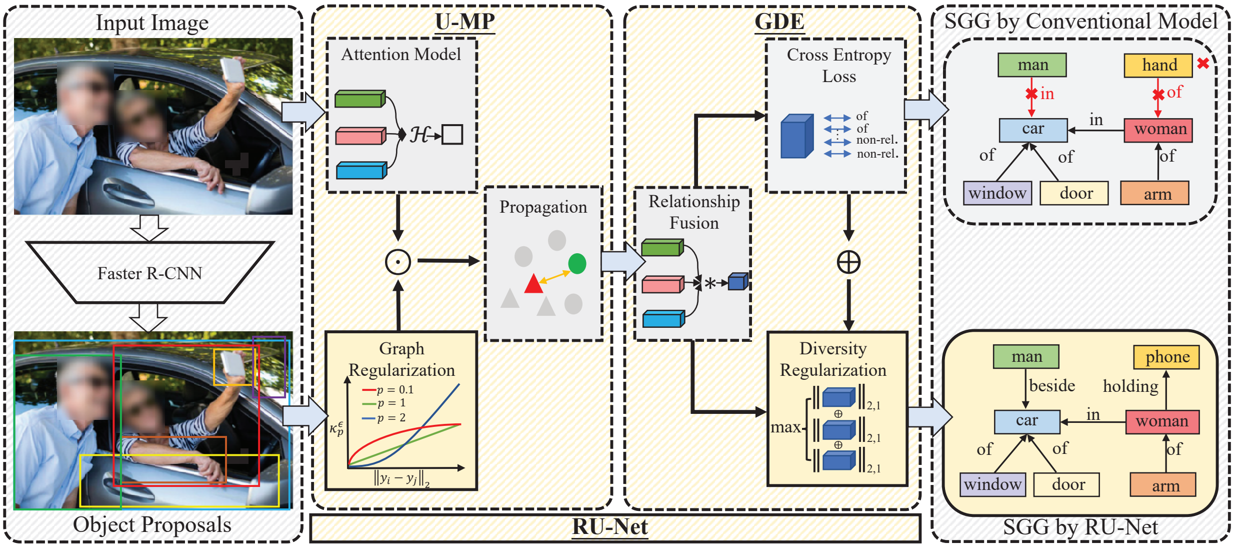


Figure 2. The framework of RU-Net. RU-Net adopts Faster R-CNN [31] to obtain object proposals. Compared with conventional SGG models (highlighted in gray), our RU-Net promotes SGG model optimization with two regularization terms (highlighted in yellow). More specifically, the graph regularization acts as a reweighting matrix to refine the attention maps and reduce ambiguity in the node representations. The diversity regularization is incorporated with the cross-entropy loss and prompts the relationship prediction diversity via rank maximization. $\oplus$ and $\odot$ represent addition and the Hadamard product, respectively. The functions $\mathcal{H}$ and $*$ are defined in Section 3.2.1 and Section 3.3, respectively. Best viewed in color.

ization to improve the robustness of existing GMP modules against spurious connections between nodes. For its part, the GDE module improves relationship prediction diversity via a group-wise $\ell_{2,1}$-based regularization term. In the below, we will describe these two components sequentially.

3.1. Preliminaries

Notations. To obtain the appearance feature for each proposal, we adopt the same approach used in [52]. There are O object categories (including background) and R relationship categories (including non-relationship). The representation for the i-th node is denoted as $\boldsymbol{x}_i \in \mathbb{R}^d$. Specifically, $\boldsymbol{x}_i$ is obtained via linear projection from the concatenation of the appearance feature, object classification probabilities, and the spatial feature. For an image that includes n nodes, we can obtain a node representation matrix $\boldsymbol{X} \in \mathbb{R}^{n\times d}$, where d is the feature dimension. In addition, we extract features from the union box of one pair of nodes i and j, denoted as $\boldsymbol{u}_{ij} \in \mathbb{R}^d$. $|\cdot|$, $\|\cdot\|_2$, and $\|\cdot\|_F$ denote the absolute value of a number, the ℓ_2-norm of a vector, and the Frobenius norm of a matrix, respectively. $[;]$ represents the concatenation operation. $\odot$ is the Hadamard product. For a matrix $\boldsymbol{S} \in \mathbb{R}^{m\times n}$, $[\boldsymbol{S}]_{ij}$ and $\boldsymbol{s}_i$ represent the ij-th entry and the i-th row of $\boldsymbol{S}$, respectively.

Smoothed ℓ_p-norm Distance Metric. To improve the robustness against spurious correlations between nodes, we utilize a smoothed ℓ_p-norm distance metric [35] as follows:

$$\kappa_p^\epsilon(x) \triangleq \begin{cases} \epsilon^{p-2}|x|^2, & |x| \le \epsilon \\ \frac{2}{p}|x|^p - \frac{2-p}{p}\epsilon^p, & |x| > \epsilon \end{cases}, \qquad (1)$$

where $\epsilon > 0$ and $0 < p \le 2$. As depicted in the coordinate plane of Figure 2, with a smaller value of p (*e.g.*, $p = 0.1$), Eq. (1) places far less emphasis on large $|x|$ and is more robust against outliers than the ℓ_2-based distance function. More details regarding the properties of Eq. (1) can be found in Appendix A.

3.2. Unrolled Message Passing

Existing SGG methods [19, 22, 24, 47] typically utilize a sequence of GMP layers to iteratively refine node representations with contextual information. However, these GMP modules may be sensitive to spurious correlations between nodes, which may lead to more ambiguous node representations. To clarify and address this issue, we will discuss two key aspects in what follows: the relation between GMP and GLD [33] and spuriousness-robust graph regularization.

3.2.1 The Relation between GMP and GLD

In each GMP layer, a function is utilized to compute the attention weight for each node pair. The node representation is then updated by aggregating neighboring information according to the learnable attention weights. The output of the

$k+1$-th GMP layer can be represented as follows:

$$\left\{\begin{array}{l} \boldsymbol{A}^{(k+1)} = \text{Normalize}(\mathcal{H}(\boldsymbol{Y}^{(k)})) \\ \boldsymbol{Y}^{(k+1)} = \text{ReLU}(\boldsymbol{Y}^{(k)} + \boldsymbol{A}^{(k+1)}\boldsymbol{Y}^{(k)}) \end{array}\right. , \tag{2}$$

where $\boldsymbol{Y} \in \mathbb{R}^{n\times d}$ represents the node representations refined by GMP. $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ stands for the learned attention matrix. $\mathcal{H}(\boldsymbol{Y})$ is a trainable attention function with $\boldsymbol{Y}$ as input. "Normalize" denotes the row-wise normalization via softmax function.

Next, we will prove that the GMP module defined in Eq. (2) essentially solves the GLD [33] problem in the SGG context. Specifically, the GLD problem can be defined as:

$$\mathcal{L}_{\text{GLD}}(\boldsymbol{Y},\boldsymbol{L}) \triangleq \|\boldsymbol{Y}-\boldsymbol{X}\|_F^2 + \mathcal{G}_{\text{GLR}}(\boldsymbol{Y},\boldsymbol{L}) , \tag{3}$$

where

$$\mathcal{G}_{\text{GLR}}(\boldsymbol{Y},\boldsymbol{L}) = \left\|\boldsymbol{L}^{\frac{1}{2}}\boldsymbol{Y}\right\|_F^2 = \sum\nolimits_{(i,j)\in\mathcal{E}} [\boldsymbol{A}]_{ij} \left\|\boldsymbol{y}_i - \boldsymbol{y}_j\right\|_2^2 . \tag{4}$$

Here, Eq. (4) is well known as the Graph Laplacian Regularization (GLR) [30]. $\mathcal{E}$ denotes the entire set of node pairs in the scene graph. Unlike the standard GLD problem [33] where the Laplacian matrix $\boldsymbol{L}$ is already known, this matrix needs to be learnt in SGG. Specifically, the Laplacian matrix is defined as follows: $\boldsymbol{L} = \boldsymbol{D}-\boldsymbol{A}$, where $[\boldsymbol{D}]_{ii} = \sum_j [\boldsymbol{A}]_{ij}$.

Motivated by the algorithm unrolling strategy [29], we can unfold a sequence of gradient steps to form an unrolled message passing (U-MP) module and optimize Eq. (3). Specifically, given $\boldsymbol{L}$, we have

$$\frac{\partial \mathcal{L}_{\text{GLD}}(\boldsymbol{Y})}{\partial \boldsymbol{Y}} = 2\boldsymbol{L}\boldsymbol{Y} + 2\boldsymbol{Y} - 2\boldsymbol{Y}^{(0)}, \tag{5}$$

where $\boldsymbol{Y}^{(0)} = \boldsymbol{X}$. Therefore, the $k+1$-th step in the gradient descent can be written as follows:

$$\boldsymbol{Y}^{(k+1)} = \boldsymbol{Y}^{(k)} - 2\alpha[(\boldsymbol{L}+\boldsymbol{I})\boldsymbol{Y}^{(k)} - \boldsymbol{Y}^{(0)}], \tag{6}$$

where α is the step size and $\boldsymbol{I}$ denotes an identity matrix. If we replace $\boldsymbol{L}$ with the random-walk normalized Laplacian [15] version $\boldsymbol{L} = \boldsymbol{I} - \boldsymbol{D}^{-1}\boldsymbol{A}$ and set α as 1/6, we have:

$$\boldsymbol{Y}^{(k+1)} = \tfrac{1}{3}(\boldsymbol{D}^{-1}\boldsymbol{A}\boldsymbol{Y}^{(k)} + \boldsymbol{Y}^{(k)} + \boldsymbol{Y}^{(0)}) . \tag{7}$$

Given $\boldsymbol{Y}$, rather than updating $\boldsymbol{L}$, we can instead directly update $\boldsymbol{A}$ with any $\mathcal{H}(\boldsymbol{Y})$ proposed in previous SGG works [6, 46, 47]. In this paper, we define the $\mathcal{H}(\boldsymbol{Y})$ as: $[\mathcal{H}(\boldsymbol{Y})]_{ij} = \boldsymbol{w}_a^T[\boldsymbol{y}_i;\boldsymbol{y}_j;\boldsymbol{u}_{ij}]$, where $\boldsymbol{w}_a \in \mathbb{R}^{3d}$ represents a fusion vector. This enables us to solve the GLD problem, defined in Eq. (3), with a GMP-like procedure as follows:

$$\left\{\begin{array}{l} \tilde{\boldsymbol{A}}^{(k+1)} = \text{Normalize}(\mathcal{H}(\boldsymbol{Y}^{(k)})) \\ \boldsymbol{Y}^{(k+1)} = \tfrac{1}{3}(\boldsymbol{Y}^{(k)} + \tilde{\boldsymbol{A}}^{(k+1)}\boldsymbol{Y}^{(k)} + \boldsymbol{Y}^{(0)}) \end{array}\right. , \tag{8}$$

where $\tilde{\boldsymbol{A}} = \boldsymbol{D}^{-1}\boldsymbol{A}$ can be viewed as a row-normalized attention matrix. It is worth noting that nonlinear activation can be incorporated into Eq. (8) by solving the revised version of Eq. (3) as: $\mathcal{L}_{\text{GLD}} + \sum_i \eta(\boldsymbol{y}_i)$. Here, $\eta(\boldsymbol{y}_i)$ represents an indicator function that assigns infinite penalty to any element of $\boldsymbol{y}_i$ is less than zero. According to the proximal gradient method [18], the proximal descent version of Eq. (8) can be written as follows:

$$\left\{\begin{array}{l} \tilde{\boldsymbol{A}}^{(k+1)} = \text{Normalize}(\mathcal{H}(\boldsymbol{Y}^{(k)})) \\ \boldsymbol{Y}^{(k+1)} = \text{ReLU}(\tfrac{1}{3}(\boldsymbol{Y}^{(k)} + \tilde{\boldsymbol{A}}^{(k+1)}\boldsymbol{Y}^{(k)} + \boldsymbol{Y}^{(0)})) \end{array}\right. . \tag{9}$$

Regardless of the scalar term (*i.e.*, $\frac{1}{3}$), the only difference, between the GMP layer defined in Eq. (2) and the solver for the GLD problem defined in Eq. (9), is the skip connection with original node representation $\boldsymbol{Y}^{(0)}$. Therefore, *existing GMP modules can be utilized as means of solving the GLD problem in SGG*. This conclusion enables us to solve the problem of spurious inter-node correlations in the GLD framework.

3.2.2 Spuriousness-robust Graph Regularization

As a quadratic penalty, the Frobenius norm in GLR (Eq. (4)) is known to be sensitive to outliers as errors accumulate quadratically [43]. For GMP-based SGG models, this implies that spurious correlations between nodes could dominate the loss, resulting in ambiguous node representations. To address this issue, we propose the following ℓ_p-based graph regularization to replace GLR in Eq. (4):

$$\mathcal{G}_p(\boldsymbol{Y},\boldsymbol{L}) = \sum\nolimits_{(i,j)\in\mathcal{E}} [\boldsymbol{A}]_{ij}\kappa_p^\epsilon\left(\|\boldsymbol{y}_i - \boldsymbol{y}_j\|_2\right) . \tag{10}$$

Accordingly, we can define a general GLD problem as follows:

$$\mathcal{L}_{\text{GLD}}^p(\boldsymbol{Y},\boldsymbol{L}) \triangleq \|\boldsymbol{Y}-\boldsymbol{X}\|_F^2 + \mathcal{G}_p(\boldsymbol{Y},\boldsymbol{L}). \tag{11}$$

When p is 2, Eq. (11) is equivalent to Eq. (3), which is the traditional GLD problem. Conventional optimization strategies, *e.g.*, gradient-based or Hessian-based methods, are computationally expensive when optimizing Eq. (11), especially when n is a large number. Motivated by the majorization-minimization algorithm [37], we utilize a quadratic upper-bound function to approximate Eq. (10) (Proof is provided in Appendix B). Specifically,

$$\hat{\mathcal{G}}_p(\boldsymbol{Y},\boldsymbol{L}) = \sum\nolimits_{(i,j)\in\mathcal{E}} [\boldsymbol{A}]_{ij}[\Omega]_{ij}\|\boldsymbol{y}_i - \boldsymbol{y}_j\|_2^2, \tag{12}$$

where

$$[\Omega]_{ij} \triangleq \begin{cases} \epsilon^{p-2}, & \|\boldsymbol{y}_i - \boldsymbol{y}_j\|_2 \le \epsilon \\ \|\boldsymbol{y}_i - \boldsymbol{y}_j\|_2^{p-2}, & \text{otherwise} \end{cases} . \tag{13}$$

Here, $[\Omega]_{ij}$ acts as a reweigting factor for $[\boldsymbol{A}]_{ij}$. Accordingly, we modify the architecture of U-MP as follows:

$$\begin{cases} \tilde{\boldsymbol{A}}^{(k+1)} = \text{Normalize}(\Omega^{(k)} \odot \mathcal{H}(\boldsymbol{Y}^{(k)})) \\ \boldsymbol{Y}^{(k+1)} = \text{ReLU}(\frac{1}{3}(\boldsymbol{Y}^{(k)} + \tilde{\boldsymbol{A}}^{(k+1)}\boldsymbol{Y}^{(k)} + \boldsymbol{Y}^{(0)})) \end{cases} . \tag{14}$$

More details of the U-MP can be found in Appendix C.

Finally, the classification score vector of the i-th node can be obtained as follows: $\boldsymbol{t}_i = \text{softmax}(\boldsymbol{W}_t \hat{\boldsymbol{y}}_i)$. Here, $\boldsymbol{W}_t \in \mathbb{R}^{O\times d}$ denotes the object classifier, while $\hat{\boldsymbol{y}}_i$ is the output node representation obtained by the final U-MP layer.

3.3. Group Diversity Enhancement

Entropy minimization has been widely adopted for optimization in previous SGG models. However, it may also reduce relationship prediction diversity due to issues related to class imbalance and missing annotations; since there are significantly more samples in majority categories, relationship prediction tends to exhibit a bias towards majority categories. In this part, we propose the GDE module to promote relationship prediction diversity. More specifically, the prediction score vector for the relationship between the i-th and j-th nodes can be expressed as follows:

$$\boldsymbol{p}_{ij} = \text{softmax}(\boldsymbol{W}_r(\hat{\boldsymbol{y}}_i * \hat{\boldsymbol{y}}_j * \boldsymbol{u}_{ij}) + \boldsymbol{f}_{ij}), \tag{15}$$

where $\boldsymbol{W}_r \in \mathbb{R}^{R\times d}$ denotes the relationship classifier. $*$ denotes a fusion function defined in [39]: $\boldsymbol{x} * \boldsymbol{y} = \text{ReLU}(\boldsymbol{W}_x\boldsymbol{x} + \boldsymbol{W}_y\boldsymbol{y}) - (\boldsymbol{W}_x\boldsymbol{x} - \boldsymbol{W}_y\boldsymbol{y}) \odot (\boldsymbol{W}_x\boldsymbol{x} - \boldsymbol{W}_y\boldsymbol{y})$, where $\boldsymbol{W}_x$ and $\boldsymbol{W}_y$ project $\boldsymbol{x}$, $\boldsymbol{y}$ to d-dimensional space, respectively. $\boldsymbol{f}_{ij}$ indicates the relationship distribution vector between the object categories of the i-th and j-th nodes in the training set, which functions in the same way as frequency bias and has been widely adopted in existing works [22, 39, 46, 52]. By gathering all prediction score vectors in the same image, we obtain a relationship prediction matrix $\boldsymbol{P} \in \mathbb{R}^{N\times R}$, which satisfies:

$$\begin{gathered} \sum\nolimits_{j=1}^{R} [\boldsymbol{P}]_{ij} = 1 \\ \text{s.t. } [\boldsymbol{P}]_{ij} \geq 0, \ \ \forall i \in 1 \dots N, \ \ j \in 1 \dots R, \end{gathered} \tag{16}$$

where N is the total number of node pairs in the image.

Considering that row-vectors in $\boldsymbol{P}$ are linearly independent when predicting different relationship categories, we can utilize the rank of $\boldsymbol{P}$ to measure the prediction diversity. However, maximizing the rank of a matrix is known to be an NP-hard problem [36]. We propose two strategies to address this issue.

First, inspired by [23, 55], we adopt the $\ell_{2,1}$-norm based regularization to approximate the rank of $\boldsymbol{P}$ as follows:

$$\|\boldsymbol{P}\|_{2,1} = \sum\nolimits_{j=1}^{R} \sqrt{\sum\nolimits_{i=1}^{N} [\boldsymbol{P}]_{ij}^2} , \tag{17}$$

which encourages a column-sparse structure for $\boldsymbol{P}$, and therefore promotes relationship prediction diversity.

Second, rather than promoting prediction diversity for all node pairs, we find it is more effective to encourage prediction diversity within pairs that share the same object categories. This is mainly because the rank maximization of $\mathcal{P}$ is hard to optimize when the number of nodes n is large. Accordingly, we divide the node pairs into several groups, each of which contains correlated node pairs. In practice, we find that selecting node pairs of the same object categories for each group is helpful to the optimization of Eq. (17).

Finally, by extending to the whole batch, we can utilize the following loss function to prompt the relationship prediction diversity:

$$\mathcal{L}_e = \frac{1}{\mathcal{M}_B}\mathcal{L}_{cls}^e - \frac{\tau}{B}\sum\nolimits_{b=1}^{B} \frac{1}{\mathcal{N}_b}\|\boldsymbol{P}_b\|_{2,1} , \tag{18}$$

where $\mathcal{L}_{cls}^e$ denotes the cross-entropy (CE) loss for relationship classification, τ is a weight, and B represents the number of groups in a mini-batch. Each group contains prediction score vectors for node pairs that share the same object categories. $\boldsymbol{P}_b$ denotes the relationship prediction matrix for the b-th group, $\mathcal{M}_B$ denotes the number of score vectors in the same batch, while $\mathcal{N}_b$ represents the number of score vectors in the b-th group.

The critical insight of Eq. (18) is to decrease a certain level of prediction hit rate on majority categories to enhance the prediction hit rate on minority categories. When the prediction diversity increases, one key concern is that some samples belonging to the majority classes may be classified as the minority class. Fortunately, the classification loss on the labeled samples will penalize incorrect predictions caused by encouraging diversity. Consequently, by selecting an appropriate value of τ, the model can generate diverse predictions while ensuring that the vast majority of labeled samples are correctly predicted.

3.4. SGG by RU-Net

During training, the overall loss function $\mathcal{L}$ for RU-Net can be expressed as follows:

$$\mathcal{L} = \frac{1}{n_b}\mathcal{L}_{cls}^o + \mathcal{L}_e, \tag{19}$$

where n_b represents the number of nodes in the batch. $\mathcal{L}_{cls}^o$ denotes the CE loss for object classification.

During testing, the object category for the i-th node is predicted by the following equation:

$$e_i = \arg\max\nolimits_{o\in\mathcal{O}}(\boldsymbol{t}_i(o)), \tag{20}$$

where $\mathcal{O}$ represents the set of object categories. The relationship category of the edge between the i-th and j-th nodes can be obtained as follows:

$$q_{ij} = \arg\max\nolimits_{r\in\mathcal{R}}(\boldsymbol{p}_{ij}(r)), \tag{21}$$

		SGDET			SGCLS			PREDCLS			
Backbone	Method	R@20	R@50	R@100	R@20	R@50	R@100	R@20	R@50	R@100	Mean
VGG-16	IMP$^\diamond$ [10]	14.6	20.7	24.5	31.7	34.6	35.4	52.7	59.3	61.3	39.3
	MOTIFS$^\diamond$ [52]	21.4	27.2	30.3	32.9	35.8	36.5	58.5	65.2	67.1	43.7
	KERN$^\diamond$ [6]	-	27.1	29.8	-	36.7	37.4	-	65.8	67.6	44.1
	GPI$^\diamond$ [14]	-	-	-	-	36.5	38.8	-	65.1	66.9	-
	VCTREE$^\diamond$ [39]	22.0	27.9	31.3	35.2	38.1	38.8	60.1	66.4	68.1	45.1
	GPS-Net$^\diamond$ [22]	22.6	28.4	31.7	36.1	39.2	40.1	60.7	66.9	68.8	45.9
	R-CAGCN$^\diamond$ [46]	22.1	28.1	31.3	35.4	38.3	39.0	60.2	66.6	68.3	45.3
	RelDN‡ [57]	-	-	32.7	-	-	36.8	-	-	68.4	-
	Seq2Seq-RL‡ [26]	22.1	30.9	34.4	34.5	38.3	39.0	60.3	66.4	68.5	46.3
	RU-Net$^\diamond$	**22.9**	28.7	32.0	37.2	39.8	40.9	61.6	67.8	69.8	46.6
	RU-Net ‡	22.6	**31.3**	**34.8**	**38.2**	**41.2**	**42.1**	**61.9**	**68.1**	**70.1**	**48.0**
RX-101	VTransE* [38]	23.0	29.7	34.3	35.4	38.6	39.4	59.0	65.7	67.6	45.9
	VCTREE* [39]	24.7	31.5	36.2	37.0	40.5	41.4	59.8	66.2	68.1	47.3
	MOTIFS* [52]	25.1	32.1	36.9	35.8	39.1	39.9	59.5	66.0	67.9	47.0
	SGGNLS* [60]	24.6	31.8	36.3	36.5	40.0	40.8	58.7	65.6	67.4	47.0
	RU-Net*	**25.7**	**32.9**	**37.5**	**38.7**	**42.4**	**43.3**	**61.2**	**67.7**	**69.6**	**48.9**

Table 1. Performance comparisons with state-of-the-art methods on the VG dataset. We compute the mean over all tasks on R@50 and R@100. $^\diamond$, ‡, and * denote using the same Faster-RCNN detector as [52], [57], and [38], respectively.

	SGDET	SGCLS	PREDCLS
Model	mR@100	mR@100	mR@100
IMP $^\diamond$ [10]	4.8	6.0	10.5
FREQ$^\diamond$ [52]	7.1	8.5	16.0
MOTIFS $^\diamond$ [52]	6.6	8.2	15.3
KERN$^\diamond$ [6]	7.3	10.0	19.2
VCTREE [39]	8.0	10.8	19.4
R-CAGCN$^\diamond$ [46]	8.8	11.1	19.9
MOTIFS* [38]	6.8	8.5	15.8
VCTREE* [38]	6.9	7.9	16.1
Transformer* [13]	8.8	10.2	17.5
RU-Net$^\diamond$	10.1	13.9	**24.7**
RU-Net*	**10.8**	**14.6**	24.2

Table 2. Performance comparisons on mean recall $(\%)$ across all 50 relationship categories in the VG dataset.

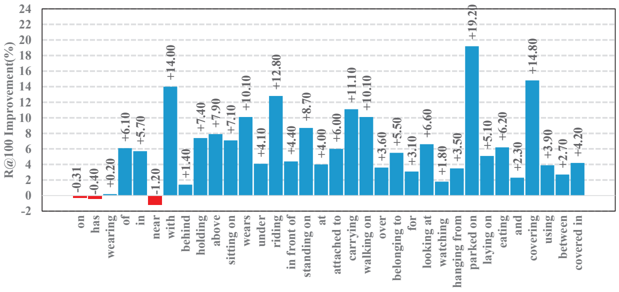


Figure 3. Absolute R@100 improvement in PREDCLS by RU-Net compared with R-CAGCN [46] on the VG dataset. We use the same backbone and evaluation metric as [39]. The Top-35 relationship categories are selected according to their occurrence frequency.

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

4. Experiments

4.1. Dataset and Evaluation Settings

Visual Genome (VG): We follow the same data cleaning strategy [10] that has been widely used in recent works. The most frequently occurring 150 object categories and 50 relationship categories are utilized for evaluation. We further adopt three conventional protocols for evaluation: (1) Scene Graph Detection (SGDET): Given an image, the model detects objects and predict relationship categories between each pair of objects. (2) Scene Graph Classification (SGCLS): Given the ground-truth location of objects, the model predicts both the object and relationship categories. (3) Predicate Classification (PREDCLS): Given the ground-truth object location and categories, the model predicts only the relationship categories. All algorithms are evaluated using the Recall@K metrics, where K=20, 50, and 100, respectively. Considering that the distribution of relationships in VG is highly imbalanced, we further utilize mean recall@K (mR@K) to evaluate the average performance on relationships [6].

Open Images (OI): We conduct experiments on both Open Images V4 and V6. we follow the same data processing and evaluation protocols utilized in [19, 22, 57]. The results are

Daraset	Model	R@50	WmAP		score$_{wtd}$
			rel	phr	
V4	RelDN [57]	74.9	35.5	38.5	44.6
	BGNN [19]	75.5	37.8	41.7	46.9
	RU-Net	**78.3**	**38.9**	**42.4**	**48.2**
V6	RelDN [57]	73.1	32.2	33.4	40.8
	VCTREE [39]	74.1	34.2	33.1	40.2
	G-RCNN [47]	74.5	33.2	34.2	41.8
	MOTIFS [52]	71.6	29.9	31.6	38.9
	GPS-Net [22]	74.8	32.9	34.0	41.7
	BGNN [19]	75.0	33.5	34.2	42.1
	RU-Net	**76.9**	**35.4**	**34.9**	**43.5**

Table 3. Comparisons with state-of-the-art methods on OI. We adopt the same evaluation metric as in [57].

	Relation Detection		Phrase Detection	
Model	R@50	R@100	R@50	R@100
VTransE [54]	19.4	22.4	14.1	15.2
KL distilation [51]	19.2	21.3	23.1	24.0
Zoom-Net [50]	18.9	21.4	24.8	28.1
CAI + SCA-M [50]	19.5	22.4	25.2	28.9
GPS-Net [22]	21.5	24.3	28.9	34.0
MF-URLN [53]	23.9	26.8	31.5	36.1
RelDN [57]	25.3	28.6	31.3	36.4
HetH [42]	22.4	24.8	30.6	35.5
Seq2Seq-RL [26]	26.1	30.2	33.4	39.1
RU-Net	**27.4**	**31.4**	**33.8**	**39.5**

Table 4. Comparisons with state-of-the-arts on VRD.

	Module		SGCLS		PREDCLS	
Exp	U-MP	GDE	R@50	R@100	R@50	R@100
1	✗	✗	40.3	41.2	66.0	67.8
2	✗	✓	40.7	41.6	67.3	69.2
3	✓	✗	42.2	43.1	66.3	68.1
4	✓	✓	**42.4**	**43.3**	**67.7**	**69.6**

Table 5. Ablation studies of the proposed method. We use the same object detection backbone as in [38].

evaluated by calculating Recall@50 (R@50), the weighted mean AP of relationships (wmAP_{rel}), and the weighted mean AP of phrase (wmAP_{phr}). The last metric is given by score$_{wtd}$ $= 0.2\times\text{R@50}+0.4\times\text{wmAP}_{rel}+0.4\times\text{wmAP}_{phr}$. Note that wmAP_{rel} requires the IoUs between the predicted and ground-truth bounding boxes to be larger than 0.5 for both objects. The wmAP_{phr} metric is similar but only requires the IoU between the predicted and ground-truth union boxes of the *subject* and *object* to be over 0.5.

Visual Relationship Detection (VRD): We adopt the same dataset split used in [25] and the same object detector from [57]. The evaluation metrics are the same as those in [57], which reports R@50 and R@100 for relationship detection and phrase detection, respectively.

Implementation Details. To facilitate a fair comparison with the majority of existing works, we utilize ResNeXt-101-FPN [21, 44] as the backbone for the OI benchmark. We further adopt ResNeXt-101-FPN [21, 44] and VGG-16 [34] as the backbones for the VG benchmark. For VRD benchmark, we utilize the VGG-16 [34] as the backbone. During training, we freeze the layers before the ROIAlign layer and optimize the remaining layers in the model using the loss functions described in Section 3.4. We optimize RU-Net via Stochastic Gradient Descent with momentum, using an initial learning rate of 10^{-3} and a batch size of 6. The top-64 object proposals in each image are chosen using per-class non-maximal suppression (NMS) with an IoU of 0.3. Additionally, the sampling ratio between pairs that do not have any relationship (background pairs) and pairs that do have relationships during training is set to 3:1. In all experiments, ϵ is set to 0.5.

4.2. Comparisons with State-of-the-art Methods

Visual Genome: As Table 1 shows, RU-Net achieves superior performance relative to the current state-of-the-art methods across various metrics. In more detail, RU-Net outperforms the recent GMP-based SGG model, named R-CAGCN [46], by 1.3% on average at R@50 and R@100 over the three protocols. It also outperforms R-CAGCN [46] by 0.7 %, 2.2 %, and 1.5 % on SGDET, SGCLS, and PREDCLS at Recall@100, respectively. Moreover, RU-Net outperforms VCTREE [39] with the same ResNeXt-101-FPN backbone by 1.3%, 1.9%, and 1.5% on SGCLS, SGDET, and PREDCLS at Recall@100, respectively. Furthermore, to demonstrate RU-Net's robustness to the class imbalance problem on VG, we also compare its performance with state-of-the-art methods using the Mean Recall metric. As shown in Table 2, RU-Net delivers a notable absolute performance gain, indicating its advantages in handling the class imbalance problem in SGG. To illustrate this advantage more vividly, we present the R@100 improvement of each predicate category compared with R-CAGCN [46] under the PREDCLS setting in Figure 3. These improvements are much larger for minority relationship categories. We owe this advantage to the power of the GDE module.

Open Images: We compare the performance of RU-Net with state-of-the-art methods in Table 3. Using the same object detector, RU-Net outperforms RelDN [57] by 3.6% and 2.7% in terms of overall metric score$_{wtd}$ for OI V4 and V6, respectively. More specifically, in OI V4, RU-Net outperforms RelDN by 3.4%, 3.4%, and 3.9% on R@50, wmAP_{rel}, and wmAP_{phr}, respectively. Furthermore, when

	p	0	0.1	0.3	1	2
SGCLS	R@20	38.1	**38.3**	38.0	38.7	37.1
	R@50	41.9	**42.1**	41.8	41.4	40.8
	R@100	43.1	**43.3**	43.0	42.6	41.9

(a) Evaluation on the value of p in Eq. (13).

	K	2	3	4	5
SGCLS	R@20	38.1	38.4	38.5	**38.7**
	R@50	41.8	42.0	42.2	**42.4**
	R@100	42.7	43.0	43.1	**43.3**

(b) Evaluation on the number of U-MP layers K.

	τ	0.05	0.1	0.15	0.2
PREDCLS	R@20	60.8	**61.2**	60.5	59.9
	R@50	67.3	**67.7**	67.0	66.4
	R@100	69.2	**69.6**	68.9	68.3

(c) Evaluation on the value of τ in Eq. (18).

Table 6. The impact of hyperparameters on the U-MP and GDE modules, respectively.

	Group	$\mathcal{I}$	$\mathcal{B}$+$\mathbb{G}$	$\mathcal{B}$+$\mathbb{G}^*$
PREDCLS	R@50	66.7	67.2	**67.7**
	R@100	68.5	69.1	**69.6**
	mR@100	22.5	23.8	**24.2**

Table 7. The Design Choices for the GDE modules.

compared with other approaches for OI V6, RU-Net consistently achieves the best performance.

Visual Relationship Detection: In Table 4, we compare the performance of RU-Net with state-of-the-art methods on the VRD dataset. It can be seen that RU-Net consistently achieves superior performance under both relation detection and phrase detection metrics.

4.3. Ablation Studies

Effectiveness of the Proposed Modules. We first perform an ablation study to justify the effectiveness of U-MP and GDE. The results are summarized in Table 5. Exp 1 in Table 5 shows the performance of the baseline, which adopt neither U-MP or GDE modules. It employs the GMP module defined in Eq. (2) for message passing. To facilitate fair comparison, all the other settings remain the same as RU-Net. Exps 2-4 show that each module helps to promote the performance of SGG. The best performance is achieved when both modules are involved. Note that U-MP and GDE are designed to refine object and relationship representations, respectively. Therefore, U-MP helps the model achieve outstanding SGCLS performance, which heavily depends on the object classification ability. Meanwhile, GDE enables the model to achieve a significant performance gain on the PREDCLS task, mainly relying on relationship prediction power.

Evaluation on hyperparameters for U-MP and GDE. We go on to verify the impact of the hyperparameters of the U-MP and GDE modules. As shown in Table 6(a), RU-Net achieves the best performance when p is set to 0.1 in the ℓ_p-based graph regularization. In Table 6(b), we show the performance of RU-Net with different numbers of U-MP layers, ranging from two to five. The model performance improves consistently as the number of U-MP layers increases. However, due to limitations on GPU memory size, we only conduct experiments up to five U-MP layers. Finally, the value of the weight τ determines the impact of the $\ell_{2,1}$-based regularization on relationship prediction. As shown in Table 6(c), the model achieves the best performance when τ equals 0.1.

Design Choices for the GDE module. In Table 7, we compare the performance of GDE with and without the grouping strategy described in Eq. (18). "$\mathcal{I}$" denotes that we impose the diversity regularization on relationship predictions for each training image. "$\mathcal{B} + \mathbb{G}$" represents that we divide all node pairs in batch into several groups according to the object categories of the nodes. For each group, we impose an $\ell_{2,1}$-based regularization term. Besides, "$\mathcal{B}$+$\mathbb{G}^*$" means we remove small groups that contain less than three elements. Experimental results in Table 7 show that the grouping strategy consistently achieves better performance.

4.4. Conclusion and Limitations

In this paper, we propose the RU-Net model, which adopts scene graph-based regularizations to handle two critical issues in SGG: ambiguous node representations and low relationship prediction diversity. From the perspective of the unrolling technique, we first prove that GMP can be interpreted as a solver for GLD. We then address the ambiguous node representation problem with the U-MP module, which utilizes an ℓ_p-based graph regularization to suppress spurious correlations between nodes. We further enhance the diversity in the relationship prediction through a group-wise $\ell_{2,1}$-based regularization term. Extensive experimental results justify the effectiveness of RU-Net on three popular SGG datasets. Like most SGG models, one limitation of our method is its dependency on pre-trained object detectors [31, 41]. In the future, we will apply the proposed techniques to end-to-end SGG models. We hope this study will provide valuable insights for future research to design interpretable and robust SGG models.

Broader Impacts. SGG is able to simultaneously provide object and relationship predictions. This merit enables more in-depth scene understanding and can potentially benefit many real-world applications, like intelligent service robot and autonomous driving. We do not foresee any negative societal consequences arising specifically from our contributions in this paper.

Acknowledgment. This work is supported by the NSF of China (Nos: 62076101, 62002090), the Program for Guangdong Introducing Innovative and Entrepreneurial Teams (No: 2017ZT07X183), and the ARC FL-170100117.

References

[1] M. Afonso, D. Bioucas, and M. Figueiredo. Fast image recovery using variable splitting and constrained optimization. *TIP*, 2010. 2

[2] I. Armeni, Z. He, J. Gwak, A. Zamir, M. Fischer, J. Malik, and S. Savarese. 3d scene graph: A structure for unified semantics, 3d space, and camera. In *ICCV*, 2019. 1

[3] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM journal on imaging sciences*, 2009. 2

[4] D. Chen, X. Liang, Y. Wang, and W Gao. Soft transfer learning via gradient diagnosis for visual relationship detection. In *WACV*, 2019. 2

[5] S. Chen, Y. Eldar, and L. Zhao. Graph unrolling networks: Interpretable neural networks for graph signal denoising. *TSP*, 2021. 2

[6] T. Chen, W. Yu, R. Chen, and L. Lin. Knowledge-embedded routing network for scene graph generation. In *CVPR*, 2019. 1, 2, 4, 6

[7] M. Chiou, H. Ding, H. Yan, C. Wang, R. Zimmermann, and J. Feng. Recovering the unbiased scene graphs from the biased ones. In *ACM MM*, 2021. 2

[8] V. Damodaran, S. Chakravarthy, A. Kumar, A. Umapathy, T. Mitamura, Y. Nakashima, N. Garcia, and C. Chu. Understanding the role of scene graphs in visual question answering. *arXiv preprint arXiv:2101.05479*, 2021. 1

[9] W. Dong, P. Wang, W. Yin, G. Shi, F. Wu, and X. Lu. Denoising prior driven deep neural network for image restoration. *TPAMI*, 2018. 2

[10] D.Xu, Y. Zhu, C. Choy, and Li Fei-Fei. Scene graph generation by iterative message passing. In *CVPR*, 2017. 2, 6

[11] K. Gregor and Y. LeCun. Learning fast approximations of sparse coding. In *ICML*, 2010. 2

[12] J. Gu, S. Joty, J. Cai, H. Zhao, X. Yang, and G. Wang. Unpaired image captioning via scene graph alignments. In *ICCV*, 2019. 1

[13] Y. Guo, L. Gao, X. Wang, Y. Hu, X. Xu, X. Lu, H. Shen, and J. Song. From general to specific: Informative scene graph generation via balance adjustment. In *ICCV*, 2021. 6

[14] R. Herzig, M. Raboh, G. Chechik, J. Berant, and A. Globerson. Mapping images to scene graphs with permutation-invariant structured prediction. In *NeurIPS*, 2018. 2, 6

[15] T. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. In *ICLR*, 2017. 4

[16] R. Krishna, Y. Zhu, O. Groth, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *IJCV*, 2017. 2

[17] A. Kuznetsova, H. Rom, et al. The open images dataset v4. *IJCV*, 2020. 2

[18] J. Li, C. Fang, and Z. Lin. Lifted proximal operator machines. In *AAAI*, volume 33, 2019. 4

[19] R. Li, S. Zhang, B. Wan, and X. He. Bipartite graph network with adaptive message passing for unbiased scene graph generation. In *CVPR*, 2021. 1, 2, 3, 6, 7

[20] Y. Li, M. Tofighi, J. Geng, V. Monga, and Y. Eldar. Efficient and interpretable deep blind image deblurring via algorithm unrolling. *TCI*, 2020. 2

[21] T. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017. 7

[22] X. Lin, C. Ding, J. Zeng, and D. Tao. Gps-net: Graph property sensing network for scene graph generation. In *CVPR*, 2020. 1, 2, 3, 5, 6, 7

[23] Q. Liu, F. Davoine, J. Yang, Y. Cui, Z. Jin, and F. Han. A fast and accurate matrix completion method based on qr decomposition and $\ell_{2,1}$-norm minimization. *TNNLS*, 2018. 5

[24] R. Liu, Z. Jiang, X. Fan, and Z. Luo. Knowledge-driven deep unrolling for robust image layer separation. *TNNLS*, 2019. 3

[25] C. Lu, R. Krishna, M. Bernstein, and F. Li. Visual relationship detection with language priors. In *ECCV*, 2016. 2, 7

[26] Y. Lu, H. Rai, J. Chang, B. Knyazev, G. Yu, S. Shekhar, G. Taylor, and M. Volkovs. Context-aware scene graph generation with seq2seq transformers. In *ICCV*, 2021. 2, 6, 7

[27] M. Mardani, Q. Sun, S. Vasawanala, V. Papyan, H. Monajemi, J. Pauly, and D. Donoho. Neural proximal gradient descent for compressive imaging. *arXiv preprint arXiv:1806.03963*, 2018. 2

[28] S. Markowitz, E. Snyder, Y. Eldar, and M. Do. Multimodal unrolled robust pca for background foreground separation. *arXiv preprint arXiv:2108.06031*, 2021. 2

[29] V. Monga, Y. Li, and Y. Eldar. Algorithm unrolling: Interpretable, efficient deep learning for signal and image processing. *SPM*, 2021. 2, 4

[30] A. Ortega, P. Frossard, J. Kovačević, J. Moura, and P. Vandergheynst. Graph signal processing: Overview, challenges, and applications. *Proceedings of the IEEE*, 2018. 4

[31] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *arXiv preprint arXiv:1506.01497*, 2015. 3, 8

[32] J. Shi, H. Zhang, and J. Li. Explainable and explicit visual reasoning over scene graphs. In *CVPR*, 2019. 1

[33] I. Shuman, S. Narang, P. Frossard, A. Ortega, and P. Vandergheynst. The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. *SPM*, 2013. 2, 3, 4

[34] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 7

[35] J. Song, P. Babu, and D. Palomar. Sparse generalized eigenvalue problem via smooth optimization. *TSP*, 2015. 3

[36] P. Sprechmann, A. Bronstein, and G. Sapiro. Learning efficient sparse and low rank models. *TPAMI*, 2015. 2, 5

[37] Y. Sun, P. Babu, and D. Palomar. Majorization-minimization algorithms in signal processing, communications, and machine learning. *TSP*, 2016. 4

[38] K. Tang, Y. Niu, J. Huang, J. Shi, and H. Zhang. Unbiased scene graph generation from biased training. In *CVPR*, 2020. 2, 6, 7

[39] K. Tang, H. Zhang, B. Wu, W. Luo, and W. Liu. Learning to compose dynamic tree structures for visual contexts. In *CVPR*, 2019. 1, 2, 5, 6, 7

[40] J. Wald, H. Dhamo, N. Navab, and F. Tombari. Learning 3d semantic scene graphs from 3d indoor reconstructions. In *CVPR*, 2020. 1

[41] W. Wang, Y. Cao, J. Zhang, and D. Tao. Fp-detr: Detection transformer advanced by fully pre-training. In *International Conference on Learning Representations*, 2021. 8

[42] W. Wang, R. Wang, S. Shan, and X. Chen. Sketching image gist: Human-mimetic hierarchical scene graph generation. In *ECCV*, 2020. 7

[43] M. West. Outlier models and prior distributions in bayesian linear regression. *Journal of the Royal Statistical Society: Series B (Methodological)*, 1984. 4

[44] S. Xie, R. Girshick, P. Dollar, Z. Tu, and K. He. Aggregated residual transformations for deep neural networks. In *CVPR*, 2017. 7

[45] S. Yan, C. Shen, Z. Jin, J. Huang, R. Jiang, Y. Chen, and X. Hua. Pcpl: Predicate-correlation perception learning for unbiased scene graph generation. In *ACM MM*, 2020. 2

[46] G. Yang, J. Zhang, Y. Zhang, B. Wu, and Y. Yang. Probabilistic modeling of semantic ambiguity for scene graph generation. In *CVPR*, 2021. 1, 4, 5, 6, 7

[47] J. Yang, J. Lu, S. Lee, D. Batra, and D. Parikh. Graph r-cnn for scene graph generation. In *ECCV*, 2018. 1, 2, 3, 4, 7

[48] Yan Yang, Jian Sun, Huibin Li, and Zongben Xu. Deep admm-net for compressive sensing mri. In *NeurIPS*, 2016. 2

[49] Y. Yang, J. Sun, H. Li, and Z. Xu. Admm-csnet: A deep learning approach for image compressive sensing. *TPAMI*, 2018. 2

[50] G. Yin, L. Sheng, B. Liu, N. Yu, X. Wang, J. Shao, and C. Loy. Zoom-net: Mining deep feature interactions for visual relationship recognition. In *ECCV*, 2018. 7

[51] R. Yu, A. Li, V. Morariu, and L. Davis. Visual relationship detection with internal and external linguistic knowledge distillation. In *ICCV*, 2017. 7

[52] R. Zellers, Mark Yatskar, Sam Thomson, and Yejin Choi. Neural motifs: Scene graph parsing with global context. In *CVPR*, 2018. 2, 3, 5, 6, 7

[53] Y. Zhan, J. Yu, T. Yu, and D. Tao. On exploring undetermined relationships for visual relationship detection. In *CVPR*, 2019. 7

[54] H. Zhang, Z. Kyaw, S. Chang, and T. Chua. Visual translation embedding network for visual relation detection. In *CVPR*, 2017. 7

[55] H. Zhang, X. Liu, H. Fan, Y. Li, and Y. Ye. Fast and accurate low-rank tensor completion methods based on qr decomposition and $\ell_{2,1}$ norm minimization. *arXiv preprint arXiv:2108.03002*, 2021. 5

[56] J. Zhang and B. Ghanem. Ista-net: Interpretable optimization-inspired deep network for image compressive sensing. In *CVPR*, 2018. 2

[57] J. Zhang, K. Shih, A. Elgammal, A. Tao, and B. Catanzaro. Graphical contrastive losses for scene graph parsing. In *CVPR*, 2019. 2, 6, 7

[58] J. Zhang and D. Tao. Empowering things with intelligence: a survey of the progress, challenges, and opportunities in artificial intelligence of things. *IEEE Internet of Things Journal*, 8(10):7789–7817, 2020. 1

[59] K. Zhang, L. Gool, and R. Timofte. Deep unfolding network for image super-resolution. In *CVPR*, 2020. 2

[60] Y. Zhong, J. Shi, J. Yang, C. Xu, and Y. Li. Learning to generate scene graph from natural language supervision. In *ICCV*, 2021. 1, 2, 6