PPDL: Predicate Probability Distribution based Loss for Unbiased Scene Graph Generation

Wei Li[1,4], Haiwei Zhang[1,2,4*], Qijie Bai[1,4], Guoqing Zhao[3], Ning Jiang[3], Xiaojie Yuan[1,2,4]
[1]College of Computer Science, [2]College of Cyber Science, Nankai University, Tianjin, China
[3]Mashang Consumer Finance Co., Ltd
[4]Tianjin Key Laboratory of Network and Data Security Technology, Tianjin, China
liwei@dbis.nankai.edu.cn, {zhhaiwei, yuanxj}@nankai.edu.cn
qijie.bai@mail.nankai.edu.cn, {guoqing.zhao02, ning.jiang02}@msxf.com

Abstract

Scene Graph Generation (SGG) has attracted more and more attention from visual researchers in recent years, since Scene Graph (SG) is valuable in many downstream tasks due to its rich structural-semantic details. However, the application value of SG on downstream tasks is severely limited by the predicate classification bias, which is caused by long-tailed data and presented as semantic bias of predicted relation predicates. Existing methods mainly reduce the prediction bias by better aggregating contexts and integrating external priori knowledge, but rarely take the semantic similarities between predicates into account. In this paper, we propose a Predicate Probability Distribution based Loss (PPDL) to train the biased SGG models and obtain unbiased Scene Graphs ultimately. Firstly, we propose a predicate probability distribution as the semantic representation of a particular predicate class. Afterwards, we re-balance the biased training loss according to the similarity between the predicted probability distribution and the estimated one, and eventually eliminate the long-tailed bias on predicate classification. Notably, the PPDL training method is model-agnostic, and extensive experiments and qualitative analyses on the Visual Genome dataset reveal significant performance improvements of our method on tail classes compared to the state-of-the-art methods.

1. Introduction

Scene graph generation (SGG) [12] is concerned with producing comprehensive, structured representations about images. A scene graph is a directed graph composed of objects entity pairs with their relations in an image, in which the objects and relations are represented as nodes and edges respectively. Due to the rich structural-semantic information, scene graph has been widely used and achieved great improvements among these downstream tasks such as image generation [10, 11], visual question answering [6, 25], image captioning [2, 7, 15, 33, 37, 43], semantic image retrieval [12,23,24] and thus has been drawing more and more attention.

Although great progress has been made in capturing the object-to-object relationship and visual reasoning, the existing SGG methods still cannot meet the requirements of

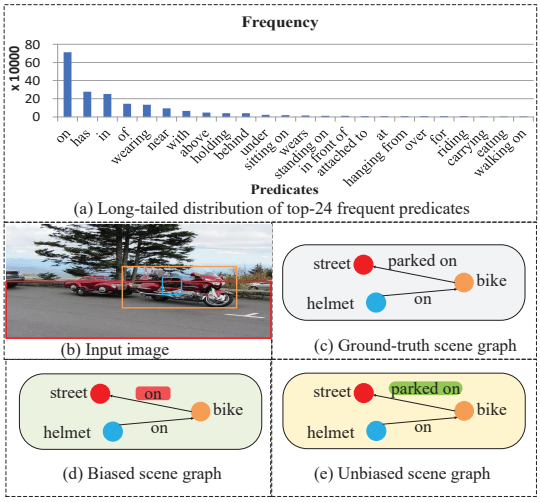

Figure 1. An illustration of long-tailed bias and unbiased scene graph generation (SGG). (a) The long-tailed distribution of different predicate categories in Visual Genome [14]. (b) The input image with bounding boxes. (c) The ground-truth scene graph. (d) The biased SG from VCTree [29] model. (e) The unbiasd SG from the same model with our proposed unbiased training method.

* Corresponding Author.

This work is supported by the Chinese Scientific and Technical Innovation Project 2030 (2018AAA0102100), National Natural Science Foundation of China (U1936206, U1903128).

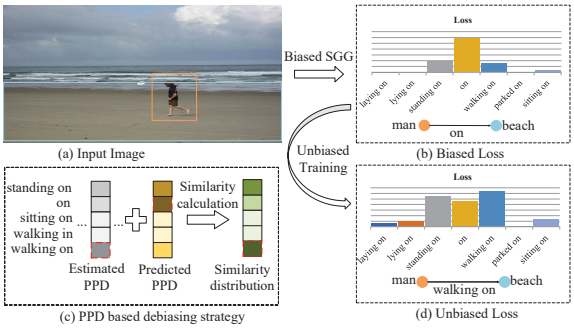

Figure 2. A toy example of **P**redicate **P**robability **D**istribution (**PPD**) based loss reweighting method. (a) The input image with bounding boxes. (b) The biased loss and SG from the biased model. (c) An illustration of our proposed PPD based debiasing strategy, which calculates the similarities between predicted PPDs and estimated ground-truth PPDs for subsequent loss reweighting. In this part, we use different shades of color to indicate the magnitude of the different values, and the highest value of each distribution is identified by a red dotted box. (d) The unbiased loss and SG from the same biased model with our unbiased training method.

downstream tasks for actual application scenarios. As illustrated in Fig.1 (a), the number of the head classes (e.g., "*on*", "*has*", "*of*", "*in*") far exceeds the tail classes (e.g., "*in front of*", "*walking on*"), which shows the long-tailed distribution of predicates in Visual Genome dataset. On the one hand, driven by long-tail data, most existing biased SGG methods are trained to "prefer" high-frequency predicates. Therefore, the tail classes tend to be neglected and misclassified in predicate classification. For instance, in Fig.1 (d), the relation between "*bike*", and "*street*" are predicted as high-frequency predicate "*on*" by the biased SGG model, VCTree [29]. On the other hand, the head classes tend to be less semantic while the tail classes contain much richer semantic information, hence even sometimes both head and tail classes can be regarded as correct in the same object pair, the semantic less result of high-frequency predicates can significantly degrade the performance of SGG in downstream tasks that require richer semantics, e.g., storytelling [30]. For these reasons, the fact that the long-tailed data distribution severely degrades the performance of SGG has attracted more attention.

To address the long-tailed distribution among different predicate classes, the Counterfactual Causal Inference [28] has been developed, which distinguishes the good and the bad bias in training, and then keeps the good one. Chen et al. [3] proposed to embed the statistical prior knowledge of predicates and object pairs in datasets into message passing. Furthermore, instead of relying on prior knowledge or better inference methods, Yu et al. [38] and Suhail et al. [26] improved the existing biased training method with an unbiased training loss. However, the approaches described above rarely care about the semantic similarity of different predicates.

Inspired by the *Focal Loss* [18] for dense object detection, we propose a novel loss function, **Predicate Probability Distribution based Loss (PPDL)**, to weaken the model's suppression for the tail classes. We first build a **Predicate Probability Distribution Matrix (PPDM)** to represent the estimated probability distribution of each predicate class. As shown in Fig.2, the predicate "*walking on*" is most likely to be misclassified as "*standing on*" and "*on*", because these predicates have the higher similarity to "*walking on*" in the probability distribution space. Thus, we can determine whether there is a bias in predicate classification of each training example by examining the similarity between the predicted predicate probability distribution and the corresponding estimated one. Thereafter, we can reweight the loss of predicted predicate if existing prediction bias. As illustrated in Fig.2 (b), the less semantic triplet $\langle$*man, on, beach*$\rangle$ is generated by biased model (e.g., VCTree model). However, after training with PPDL, we can down-weight the loss of head classes and focus on training on hard but meaningful tail classes, eventually obtaining much more meaningful predicates (e.g., "*walking on*" as shown in Fig.2 (d)).

In order to better estimate the probability distribution of each predicate class, we propose a dynamic updating method for PPDM during training time. Instead of relying on simple co-occurrence statistics, the PPDM can be adaptively updated by summing the probability distributions of the unbiased predicted relationships in each mini-batch and gradually approaches the real average probability distribution of the training data.

In summary, our main contributions are three-fold:

- We analyze the ignorance of semantic relevance among some predicates in existing SGG models and integrate the predicate probability distribution into unbiased training loss, PPDL, which is proposed to mitigate the impact of long-tailed data on SGG. We highlight that PPDL is a model-agnostic training strategy and thus applicable for a variety of existing SGG models.
- Furthermore, we propose an adaptively updating method for PPDM to estimate realistic probability distribution of each predicate during biased model training, which will be described in more detail below.
- Extensive experiments and qualitative analyses on the widely used SGG benchmark dataset of Visual Genome demonstrate the effectiveness of our proposed unbiased training loss, PPDL. Impressively, the proposed PPDL significantly improves most of the predicates, and the performance of tail classes is enhanced apparently.

2. Related Work

2.1. Scene Graph Generation (SGG)

The goal of SGG is to detect entity pairs with their relations in an image, in form of ⟨*Subject, Predicate, Object*⟩. Generally, SGG models consist of three main modules: proposal generation localizing the bounding box of objects, object classification labeling the detected objects, and relationship prediction predicting the predicates between pairwise objects, thus most novel SGG methods are mainly innovative in these three modules. Instead of relying on an extra powerful object detector [22] to obtain object proposals, Liu et al. [19] localized the objects and refined the bounding boxes by applying a fully convolutional network throughout the SGG model. For better utilizing the contexts for object classification and relationship prediction, RNNs and graph convolutional network were applied to propagate image contexts, e.g., IMP+ [32], MOTIFS [41], Graph R-CNN [36]. VCTree [29] captured local and global visual contexts by exploiting dynamic tree structures.

Furthermore, since Tang et al. [29] and Chen et al. [3] proposed the unbiased evaluation metric, *Mean Recall*, many researchers have focused on the long-tailed bias of the mainstream Visual Genome Dataset [14]. Gu et al. [8] and Chen et al. [3] integrated external knowledge into SGG models to address the bias of noisy annotations. Tang et al. [28] proposed to adapt counterfactual causal inference to eliminate the prediction bias caused by the long-tailed data. Yan et al. [34] proposed to perform reweighting with class relatedness-aware weights. CogTree [38] constructed a hierarchical cognitive tree of predicates with the bias of existing model and focused on a small portion of easily confused predicates. Suhail et al. [26] proposed an energy-based training method that allows the model to perform structure aware learning and to alleviate the long-tailed bias of predicate prediction. And there are some other interesting tricks for removing prediction bias or breaking the limitations of datasets in SGG. Yang et al. [35] proposed to model relation prediction probability as *Gaussian* distribution for generating diverse scene graphs. Chen et al. [4] introduced a semi-supervised method to train SGG models with a limited label setting. Zhang et al. [42] proposed graphical contrastive loss that specifically targets the entity instance confusion and proximal relationship ambiguity. Zareian et al. [40] proposed a weakly supervised learning framework for SGG that enables training without bounding box annotations.

The methods mentioned above rarely leverage the class relatedness between object pairs or achieve unbiased SGG through unbiased training. However, instead of utilizing the external knowledge or focusing on removing the bias of visual feature learning, we proposed to quantify the predicate correlation through the Euclidean Distance between the predicted predicate probability distribution and the corresponding estimated one, and then build an unbiased training framework based on the predicate relevance.

2.2. Class re-balancing

Real-world datasets often have long-tailed data distributions, as shown in Fig.1 (a), the number of predicates sampled from different categories varies greatly. Therefore, the biased models trained on these datasets tend to perform poorly on less presented classes. In response to this problem, the researchers proposed various class re-balancing methods, which can be divided into two main categories: one is the resampling method [21, 39], the other is the reweighting method [13, 20]. In general, resampling methods mainly means under-sampling [9] the frequent classes and over-sampling [1] the less presented classes. Nevertheless, when it comes to small datasets, under-sampling methods tend to ignore a large number of data examples, leading to not only data waste but also severe performance degradation. While over-sampling of low-frequency examples can cause overfitting problems on repeatedly sampled examples. The reweighting methods aim to give different weights to the predicted loss of different classes of examples, it is slightly more complex than the resampling method but possesses more maneuverability. The straightforward reweighting methods are to use the inverse of the proportions of different classes as weights of the predicted losses, but they have a significant detrimental effect on the overall performance, especially for the head classes. Lin et al. [17] proposed the concept of effective number of examples as having the greatest impact on the performance of the model and used the inverse of the proportion of effective number of examples as the weight value for loss re-balancing. Focal Loss [18] can also be seen as reweighting method, which reduces the loss weights of well-classified examples and focuses training on a small number of hard examples in order to improve the performance of hard examples and the average performance across the whole dataset.

Because the resampling method is prone to overfitting and performance degradation, we propose to re-balance the weight of different classes with the help of the correlation between the predicted probability distributions and the estimated ones, focusing on training hard examples and lessening the long-tailed bias.

3. Methodology

The PPDL unbiased training framework proposed in this paper consists of two main modules: one is the biased SGG, and the other is an unbiased training loss function. Thus in this section, we first illustrate an overview of general approaches of biased SGG, followed by a description of our proposed predicate probability distribution based unbiased training loss for SGG.

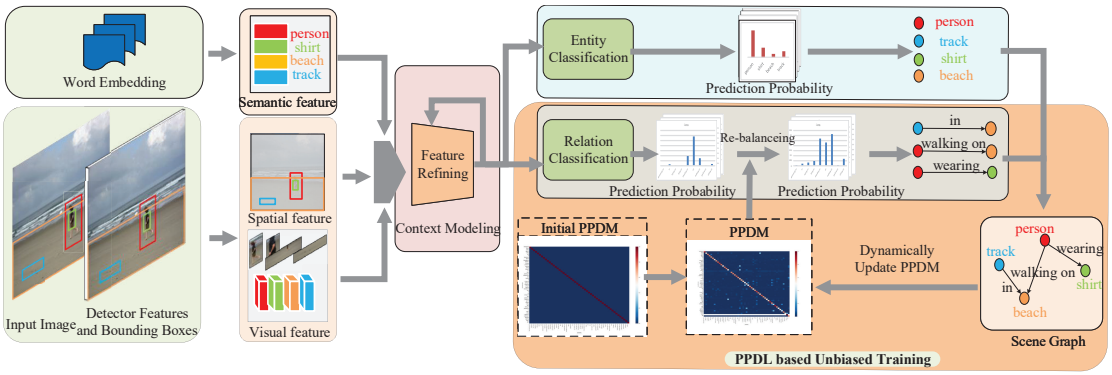

Figure 3. An illustration of PPDL unbias training framework. We extract features and proposals with an object detector and feed them into a feature interaction module. Image features are propagated iteratively to capture local and global contexts, and then further decoded into biased probabilities by the object and relationship classification modules. We initialize PPDM as the identity matrix and then iteratively update it during the training process. Meanwhile, we re-balance the Cross-Entropy loss based on the similarity between the predicted probability distribution of each predicate and the corresponding estimated one, finally obtain unbiased scene graphs. Best viewed in color.

3.1. Scene Graph Generation

As shown in Fig.3, SGG methods typically consist of two main modules: object entity detection and relationship classification. Generally, given an input image I, Object entity detection aims to obtain the visual features $\{x_i\}^n$, bounding boxes $B = \{b_i\}^n$ of each object, and preliminary labels $L = \{l_i\}^n$ of the detected object entities, where n represents the number of detected entities in the input image. As the formulation: "*Input*$\{x_i, l_i, b_i\} \rightarrow$ *Output*$\{f_i\}$", a set of object features $F = \{f_i\}^n$ can be obtained and used for object detection and subsequent relationship classification. Relationship classification is designed to obtain a set of relationships $R = \{r_{i,j} | i, j \in \{1, 2, ..., n\}\}^k$ among detected pairwise object entities. When judging the relationship between object i and object j, the relationship feature consists of three important parts: the object features f_i and f_j, the label embeddings of entity pair l_i and l_j, and the visual feature $x_{i,j}$ of the overlap area of two entities. Therefore, the relationship feature $f_{r_{i,j}}$ can be obtained as formulation: "*Input*$\{f_i, f_j, l_i, l_j, x_{i,j}\} \rightarrow$ *Output*$\{f_{r_{i,j}}\}$".

For better capturing the contexts, most existing SGG methods use the RNNs and graph convolutions to propagate image context and iteratively update these features. The updated object feature f_i^{t+1} and relation feature $f_{r_{i,j}}^{t+1}$ can be defined as $f_i^{t+1} = G\left(f_{r_{i,j}}^t, f_{r_{j,i}}^t, f_i^t\right)$ and $f_{r_{i,j}}^{t+1} = H\left(f_{r_{i,j}}^t, f_i^t, f_j^t\right)$, where $G(\cdot)$ and $H(\cdot)$ are layers for feature interaction of objects and relations respectively. Afterwards, these features are fed into the corresponding classification head to predict the object and relation labels. The output module can be divided into two decoders, which can be represented as $P_{o_i} = \mathcal{D}_{object}\left(f_i^T\right)$ and $P_{r_{i,j}} = \mathcal{D}_{predicate}\left(f_{r_{i,j}}^T\right)$, where T means the last iteration.

So far, the relationship triplets of pairwise objects can be obtained and further organized into scene graphs. However, the performance of different predicates is still long-tailed due to the unbalanced data and biased training strategy. In this work, we propose a predicate similarity based unbiased loss to eliminate the long-tailed bias during model training.

3.2. Predicate Probability Distribution based Loss

Most existing SGG methods always perform well in the head predicate classes but poorly in tail ones. However, the high-frequency head classes, such as "*on, near, of*", are not rich in semantics and are less helpful for downstream tasks. As shown in Fig.1, it is easy to find that prediction bias often occurs between two predicates with similar semantics, e.g., "*parked on*" and "*on*". For a particular object pair, the predicates with similar semantics will be closer in classification probability. For instance, in Fig.2 (b), the prediction probabilities of "*standing on*" and "*walking on*" are just less than that of the biased predicted predicate "*on*", but far exceed that of other predicates in the predicted probability distribution. This indicates that the biased models are weak in distinguishing between "*walking on*", "*standing on*" and "*on*" in this example. Thus, it's natural for us to find a way to measure the semantic similarity between predicates and then use it to correct the classification bias between similar predicates, which will be explained in detail below.

3.2.1 Unbiased Training Loss Function

The cross-entropy loss is commonly used in SGG models, which can be described as $\mathcal{L}_{CE} = -\sum_{i=1}^{m} y_i log(p_i)$, where p_i and y_i are the predicted predicate probability and the ground-truth predicate which represented as one-hot vector, and m is the number of predicate categories. Due to long-tailed data, the models trained with cross-entropy loss

"prefer" the head classes. Thus, for adjusting the optimization direction and focusing on training tail predicate classes, some researchers reweighted the loss of different predicates according to their frequencies. However, as aforementioned, the existing reweighting methods severely weaken the performance of the high-frequency relations.

In order to find a suitable way to measure the similarity between relational predicates and adjust the reweighting weight for the loss function, we propose to represent each predicate class as a prediction probability distribution with a dimension of predicate categories. We select the predicates which are correctly predicted by biased models and average them then, the PPD_r can be expressed as follows:

$$PPD_r = \frac{1}{N} \sum_{(x,y=r)\in D} P(x|p=y) \tag{1}$$

where x and N are the visual feature of an image and the number of correctly predicted relations in the dataset, p and y mean the biased predicted predicate and the ground-truth one, respectively, and $P(x|p=y)$ means the predicted probability distribution of relation which is correctly predicted by biased model. Furthermore, we can calculate the Euclidean Distance (ED) between two PPDs, and use its reciprocal to represent the similarity between two predicates. This process can be expressed as follows:

$$s(p_i, r) = \frac{1}{ED(g(x^i), PPD_r) + 1} \tag{2}$$

where $s(p_i, r)$ is the similarity between relation i and predicate r, and x^i, $g(\cdot)$ are the feature of relation i and the prediction probability generation function. The $ED(\cdot)$ means Euclidean Distance of two probability distributions.

Combining the similarity between two PPDs and the frequency based reweighting methods, we propose a Binarization Function $\theta(p_i, \mathcal{R}_{gt})$ which is presented as Equ.3 to determine whether to reweight the loss,

$$\theta(p_i, \mathcal{R}_{gt}) = \begin{cases} 1\,, & \textit{if } p_i \neq r^{'}, r^{'} = \arg\max\limits_{r, r\in\mathcal{R}_{gt}}(s(p_i, r)); \\ 0\,, & \textit{otherwise.} \end{cases} \tag{3}$$

where $\mathcal{R}_{gt}$ means the estimated PPDs of all predicate classes in ground-truth data, and $\arg\max\limits_{r, r\in\mathcal{R}_{gt}}(s(p_i, r))$ is a mathematical function for finding the predicate class $r^{'}$ with the largest similarity with the predicted predicate probability distribution. If the predicted predicate p_i is different from the most similar predicate $r^{'}$, there may be prediction bias when predicting the predicate of relation i.

By simply reweighting the loss according to the frequencies of different predicate categories, the traditional reweighting methods ignore the correctly predicted high-frequency examples and focus too much on the low-frequency ones, leading to the overfitting for tail classes and

Algorithm 1 PPDM update algorithm and PPD based loss re-balancing strategy during training time.

Require: Training dataset D; initial $PPDM_0$; the weight vector $\mathcal{W} \in R^K$; balancing parameter β; momentum α; Cross-Entropy $CE(\cdot)$.
Ensure: Unbiased model $\mathcal{G}(\cdot)$; estimated $PPDM_T$.

```
Let t = 0;
Initialize PPDM_0 as a unit matrix;
for t ∈ {1, 2, 3, ..., N_Epoch} : do
    Set the shuffled dataset D as D^t;
    for each mimi batch B = {(x^i, p_i)} ∈ D^t : do
        Set g(x^i) as the predicted biased probabilities of relation i;
        Set similarity matrix S = {s(p_i, r)} ∈ R^{|B|×K};
        Set L = {l_i} as unbiased loss for predicates classified;
        for each predicate probability g(x^i) : do
            for each predicate class r ∈ R_gt, R_gt ∈ R^K : do
                s(p_i, r) = 1 / (ED(g(x^i), PPDM_{t-1,r}) + 1);
            end for
            B' ← {(x^i, p_i) ∈ B | p_i = arg max_{r, r∈R_gt}(s(p_i, r))};
            l_i = CE(g(x^i)) + β × θ(p_i, R_gt) × (W · CE(g(x^i)));
        end for
        for each predicate class r ∈ {1, 2, 3, ..., K} : do
            PPDM'_r ← 1/|B'| Σ_{(x,p=r)∈B'} g(x);
        end for
        PPDM_t ← α × PPDM' + (1 − α) × PPDM_{t-1};
    end for
end for
return model G(·) and PPDM_T;
```

performance damage to head classes. With the $\theta(p_i, \mathcal{R}_{gt})$, we can adaptively give a heavier penalty to the predicted predicates that do not match the corresponding ground-truth predicate probability distribution, and set the loss weight to zero for correctly predicted predicates. Thus, the predicate probability distribution based loss we proposed only focuses on the biased predicted relations and causes less damage to the performance of high-frequency predicates, and it is designed as follows:

$$\mathcal{L}_{PPDL} = \theta(p_i, \mathcal{R}_{gt}) \times (\mathcal{W} \cdot \mathcal{L}_{CE}) \tag{4}$$

where the weight vector $\mathcal{W}$ is simply represented by the inverse of the fraction of each predicate classes that appear in the dataset, the $\mathcal{L}_{CE}$ means Cross-Entropy loss, and the $\cdot$ means vector dot product operation. Furthermore, we introduce an equilibrium parameter β to balance traditional cross-entropy loss and the reweighted one. The final loss function can be represented as follows:

$$\mathcal{L} = \mathcal{L}_{CE} + \beta \times \mathcal{L}_{PPDL} \tag{5}$$

3.2.2 Dynamic Updating Strategy For PPD

The long-tailed data is caused not only by the tendency of annotators to label simple predicates, but also by the incomplete relationship annotation. Therefore, the PPD calculated by Equ.1 ignores many relationships that are correctly predicted but not annotated in the dataset, and does

not completely eliminate the long-tailed data impact. Hence we propose to select predicted predicate which is closet to the corresponding estimated predicate representation instead of which matches the ground-truth. Generally, as shown in Equ.2, we evaluate the similarity of two predicates by calculate the inverse of the Euclidean Distance of their prediction probability distributions. Furthermore, adopting the dynamic updating idea in [5], we propose a dynamic updating strategy for PPDM estimating. As shown in Algorithm 1, we select the prediction probability that matches the corresponding estimated predicate probability distribution and calculate the average predicate probability distribution of each mini batch as follows:

$$\mathcal{B}' \leftarrow \{(x_i, p_i) \in \mathcal{B} | p_i = \arg\max_{r, r \in \mathcal{R}_{gt}}(s(p_i, r))\} \quad (6)$$

where $\mathcal{B}'$ is the collection of results that meet the criteria mentioned above. Then, as shown in Equ.7, the estimated $PPDM$ can be dynamically updated by the per-batch average $PPDM'$ and a momentum α, making the latest estimate play a more important role in the updating process.

$$PPDM_t \leftarrow \alpha \times PPDM' + (1 - \alpha) \times PPDM_{t-1} \quad (7)$$

4. Experiments

4.1. Experiment Setting

Dataset. We evaluate our approach on commonly used large-scale Visual Genome (VG) benchmark [14], which consists of 108077 images across 75k object categories and 40k predicate categories. Since most of relationship categories contain too few examples to support training, we train the models with the most frequent 150 object classes and 50 predicate classes following the setting from previous works [28, 29, 32, 38, 41]. 5000 images are selected for validation, and training set and test set account for 70% and 30% of dataset, respectively.

Evaluation Settings. Following Xu et al. [32], we train and evaluate various SGG models in three setups: (1) *Predicate Classification (**PredCls**)* predicting the predicate of pairwise objects given bounding boxes and object labels, (2) *Scene Graph Classification (**SGCls**)* predicting the predicates and the object labels given bounding boxes, (3) *Scene Graph Generation (**SGGen**)* predicting predicate between each pair of the detected objects with only input image. Since traditional metrics recall@K (R@K) cannot reflect the impact from long-tailed data, we utilize the mean Recall@K (mR@K) as the main metrics following [3, 28], which evaluates the R@K of each predicate class separately and averages them then. Furthermore, Unconstrained and constrained mR@K are used to indicate the semantic richness of multi and single output relationships, respectively.

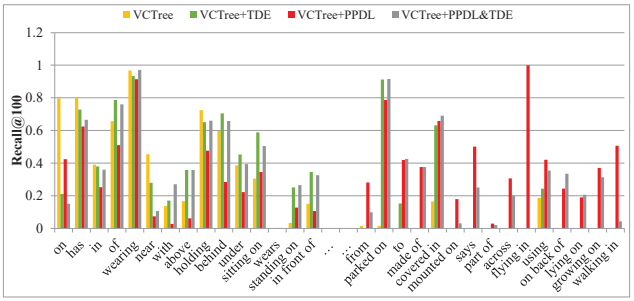

Figure 4. Performance comparison among several methods on VG150 dataset. The constrained R@100 for the head 15 and the tail 15 predicate classes on the PredCls task is presented.

Implementation Details. Following previous works [3, 28, 41], we adopt a frozen Faster-RCNN [22] as the object detector, which is equipped with the ResNeXt-101-FPN backbone [17, 31] and pre-trained by Tang et al. [28]. For SGG training, we adopt the Scene-Graph-Generation benchmark [27] proposed by Tang et al. [28], which trained SGG models using SGD as an optimizer. Batch size and initial learning rate are set to 12 and 0.01 for three evaluation setups. Following Tang et al. [28], the learning rate would be decayed by 10 two times after the validation performance plateaus. The momentum α and β were set to 0.1 and 0.03 separately. All experiments are implemented with PyTorch and two NVIDIA 2080 GPUs are used for training.

4.2. Comparisons with State-of-the-Arts

Settings. In order to verify the performance improvement of our proposed PPDL unbiased training method, we experimented our proposed method with several existing biased models (e.g., IMP+ [32], MOTIFS [41], VCTree [29]) and compared them with other unbiased SGG methods that incorporate debiasing strategies (e.g., TDE [28], CogTree [38], EBML [26], PCPL [34]) in SGG.

Quantitative Results. Since the long-tailed data distribution, as shown in Table 1, 2 and Fig.4, the mean recall of biased models is pulled down seriously due to the poor performance of tail predicate classes, and our model achieves much higher mR@K metrics than all biased baseline models. In Table 1, PPDL performs much better than other debiasing strategies (e.g., TDE, CogTree, EBML) by achieving much higher mR@K metrics but paying a smaller R@K metrics reduction. The mR@100 of VCTree+PPDL is 18%, 60.0%, and 20% higher than that of VCTree+TDE in three evaluation tasks. Particularly, since our method is model-agnostic, it is possible to combine the advantages of different models and achieve much better performance. As we can see in Table 1 and Fig.4, combining TDE and PPDL can achieve much better performance than applying them separately. Although the PCPL method outperforms our

Method	Predicate Classification		Scene Graph Classification		Scene Graph Generation	
	mR@50/100	R@50/100	mR@50/100	R@50/100	mR@50/100	R@50/100
IMP† [12]	9.8 / 10.5	59.3 / 61.3	5.8 / 6	34.6 / 35.4	3.8 / 4.8	20.7 / 24.5
MOTIFS† [41]	14.0 / 15.3	65.2 / 67.1	7.7 / 8.2	35.8 / 36.5	5.7 / 6.6	27.2 / 30.3
VCTree† [29]	17.9 / 19.4	66.4 / 68.1	10.1 / 10.8	38.1 / 38.8	6.9 / 8.0	27.9 / 31.3
PCPL† [34]	35.2 / 37.8	50.8 / 52.6	18.6 / 19.6	27.6 / 28.4	9.5 / 11.7	14.6 / 18.6
IMP+EBML† [26]	11.8 ↑2.0 / 12.8 ↑2.3	- / -	6.8 ↑1.0 / 7.2 ↑1.2	- / -	4.2 ↑0.4 / 5.4 ↑0.6	- / -
IMP‡+PPDL	**24.8** ↑**15.0** / **25.3** ↑**14.8**	39.5 / 39.7	**14.2** ↑**8.4** / **15.9** ↑**9.9**	25.8 / 26.7	**9.8** ↑**6.0** / **10.4** ↑**5.6**	18.5 / 19.4
MOTIFS+TDE† [28]	25.5 ↑11.5 / 29.1 ↑13.8	46.2 / 51.4	13.1 ↑5.4 / 14.9 ↑6.7	27.7 / 29.9	8.2 ↑2.5 / 9.8 ↑3.2	16.9 / 20.3
MOTIFS+CogTree† [38]	26.4 ↑12.4 / 29.0 ↑13.7	35.6 / 36.8	14.9 ↑7.2 / 16.1 ↑7.9	21.6 / 22.2	10.4 ↑4.7 / 11.8 ↑5.2	20.0 / 22.1
MOTIFS+EBML† [26]	18.0 ↑4.0 / 19.5 ↑4.2	- / -	10.2 ↑2.5 / 11.0 ↑2.8	- / -	7.7 ↑2.0 / 9.1 ↑2.5	- / -
MOTIFS‡+PPDL	**32.2** ↑**18.2** / **33.3** ↑**18.0**	47.2 / 47.6	**17.5** ↑**9.8** / **18.2** ↑**10.0**	28.4 / 29.3	**11.4** ↑**5.7** / **13.5** ↑**6.9**	21.2 / 23.9
VCTree+TDE† [28]	25.4 ↑7.5 / 28.7 ↑9.3	47.2 / 51.6	12.2 ↑2.1 / 14.0 ↑3.2	25.4 / 27.9	9.3 ↑2.4 / 11.1 ↑3.1	19.4 / 23.2
VCTree+CogTree† [38]	27.6 ↑9.7 / 29.7 ↑10.3	44.0 / 45.4	18.8 ↑8.7 / 19.9 ↑9.1	30.9 / 31.7	10.4 ↑3.5 / 12.1 ↑4.1	18.2 / 20.4
VCTree+EBML† [26]	18.2 ↑0.3 / 19.7 ↑0.3	- / -	12.5 ↑2.4 / 13.5 ↑2.7	- / -	7.7 ↑0.8 / 9.1 ↑1.1	- / -
VCTree+TDE&EBML† [26]	26.7 ↑8.8 / 30.0 ↑10.6	- / -	18.2 ↑8.1 / 20.5 ↑9.7	- / -	9.7 ↑2.8 / 11.6 ↑3.6	- / -
VCTree‡+PPDL	**33.3** ↑**15.4** / 33.8 ↑14.4	47.6 / 48.0	**21.8** ↑**11.7** / **22.4** ↑**11.6**	32.1 / 33.0	11.3 ↑4.4 / 13.3 ↑5.3	20.1 / 22.9
VCTree+TDE‡&PPDL	33.0 ↑15.1 / **36.2** ↑**16.8**	41.6 / 43.6	20.2 ↑10.1 / 22.0 ↑11.2	24.8 / 26.2	**12.2** ↑**5.3** / **14.4** ↑**6.4**	13.6 / 16.5

Table 1. Comparison of **constrained R@K** and **constrained mR@K** for PredCls, SGCls, and SGGen tasks. † means that the performance is reported by the respective paper. ‡ indicates that model is re-implemented under our implementation setting. ↑ indicates the performance improvement compared to the corresponding base models. † and ‡ in Table 2 and 3 have the same meaning.

Method	PredCls	SGCls	SGGen
	mR@50/100	mR@50/100	mR@50/100
IMP† [12]	20.3/28.9	12.1/16.9	5.4/8.0
MOTIFS† [41]	27.5/37.9	15.4/20.6	9.3/12.9
VCTree‡	34.8/47.1	22.5/30.0	12.4/16.8
PCPL† [34]	**50.6/62.6**	26.8/32.8	10.4/14.4
IMP‡+PPDL	33.9/38.4	19.5/23.8	11.3/12.6
MOTIFS‡+PPDL	41.8/46.6	22.5/25.9	15.5/18.8
VCTree‡+PPDL	43.3/47.0	27.9/31.3	15.1/18.3
VCTree+TDE‡&PPDL	45.8/58.2	**29.3/36.8**	**16.7/20.6**

Table 2. Comparison of the **unconstrained mR@K** on three evaluation tasks. The second largest value is underlined.

Type	Method	Predicate Classification R@50/100	Predicate Classification mR@50/100
1	Baseline‡	**65.8/67.4**	17.1/18.4
2	PPDL*	48.7/49.3	31.6/32.3
3	PPDL($w/o\ \theta(\cdot)$)	47.1/46.5	32.9/**33.8**
4	PPDL($w/\ \theta(\cdot)$)	47.6/48.0	**33.3/33.8**

Table 3. Ablation study of our method. The **constrained R@K** and **constrained mR@K** of the VCTree model on the PredCls tasks are presented. $*$ indicates model training using PPDL without dynamic updating strategy. The baseline model was trained with Cross-Entropy Loss.

method on the PredCls task, our method achieves higher mR@K on the SGCls and SGGen tasks. Unlike PCPL method that measures predicate similarity by the distance between predicate representations, our method utilizes dynamically estimated predicate probability representations to judge prediction bias. Therefore, we speculate that PPDL relies less on ground truth labels. And this property en-

β	Predicate Classification mR@20	mR@50	mR@100
0.01	25.6	31.2	31.9
0.02	28.7	31.9	32.9
0.03	29.2	**33.0**	**33.6**
0.04	29.6	32.5	33.3
0.05	**29.9**	32.7	33.2

Table 4. Ablation study for β on VCTree [29] model. The **constrained mR@20/50/100** on the PredCls task are presented.

hances the advantage of PPDL over PCPL method in SGCls and SGGen tasks. Furthermore, as shown in Table 1, the R@50/100 of biased methods (e.g., IMP [16], MOTIFS [41], VCTree [29]) are higher than that of debiasing methods (e.g., CogTree [38], TDE [28], EBML [26], PCPL [34]) and our proposed PPDL. However, compared to the state-of-the-arts debiasing strategies, e.g., CogTree and TDE, PPDL achieves better or comparable performance on the R@K metric in three sub-tasks. For instance, in the SGGen sub-task for the MOTIFS model, our R@100 is 18% and 8% higher than TDE and CogTree, respectively. As shown in Fig.4, PPDL significantly improves the R@100 of the tail classes and performs better on most of all predicate categories. This proves that our PPDL significantly improves the performance of the tail classes and has less and acceptable damage to the performance of the head classes.

Qualitative Analysis. To better present the effectiveness of PPDL on the semantic enhancement of predicates, we provide qualitative examples in Fig.5. As we can observe in the results of bottom two rows, the baseline model (VCTree) classfies relations as coarse-grained predicates (e.g., "*on*", "*near*"), while our method successfully classifies rela-

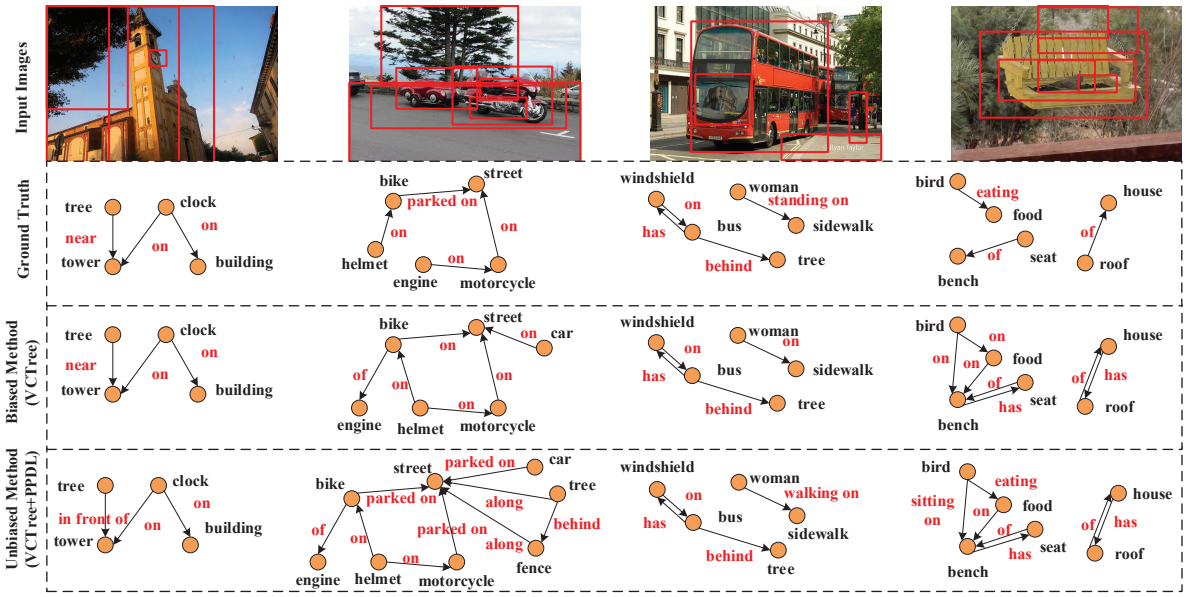

Figure 5. Qualitative results for PredCls sub-task. The top two rows show input images, detected bounding boxes indicated with red boxes, and ground-truth scene graphs. The bottom two rows show the scene graphs generated from the VCTree baseline model and VCTree+PPDL method respectively. Due to space limitations, part of the detected objects is removed from the results. Best viewed in color.

tions to more meaningful and fine-grained predicates (e.g., *bike* $\xrightarrow{parked\ on}$ *street*). Therefore, it is obvious to find that our method improves the baseline methods greatly and has a significant contribution on the predicate classification of tail classes. Although some of our unbiased results (e.g., *tree* $\xrightarrow{in\ front\ of}$ *tower*) seem to be different from the ground-truth SG, our predicted predicates are much more accurate and valuable from a semantic point of view.

4.3. Ablation Studies

Analysis of PPDL. As shown in Table 3, we conduct ablation studies on the newly proposed reweighting method and the PPD based classes re-balancing policy. Type 1 is a baseline method which was trained with the implementation settings described above, type 2 is an ablation study of dynamic updating strategy, type 3 and type 4 respectively trained the VCTree model with the PPDL loss without/with the Binarization Function $\theta(\cdot)$. Specifically, the $\theta(\cdot)$ was set to 1 throughout the experiment in type 3 for removing the effect of $\theta(\cdot)$. According to the comparison between type 1 and type 4, we can observe that PPDL is indeed helpful to the performance of the tail class, and the mR@100 increases by 83%. Furthermore, compared to type 3, the R@50/100 and mR@50 of type 4 increases by 1% / 3% and 1%, respectively. This means that PPDL, as an effective debiasing strategy adapted from reweighting methods, can largely preserve the performance of the head classes when up-weighting the loss of tail classes. Furthermore, after ablating the dynamic updating strategy, our method can obtain the R@50/100 metrics of 48.7/49.3 and the mR@50/100 metrics of 31.6/32.3 in the PredCls task, it shows that the performance improvement is mainly contributed by the unbiased loss, and the dynamic updating strategy further improves the mR@K metrics.

Analysis of β. We experiment with different β values from 0.01 to 0.05, in order to judge the effect of different loss balancing hyperparameter on the performance of the model. As shown in Table 4, we can observe that the mR@K (20/50/100) of PredCls task increases with the increase of β. While the β goes up to a certain extent, the performance of models begins to fall. Therefore, we set β to 0.03 in the model training, which can achieve better results.

5. Conclusion

In this paper, we explore to use estimated probability distribution to represent high-level semantic of predicate and reweight training loss according to the similarity between predicted predicate and the estimated one. With the help of the dynamic updating strategy for predicate prediction distributions, we can estimate a more realistic predicate prediction distribution for each predicate class and better measure the similarity between two predicates. We lessen the bias brought by the long-tailed data distribution and maintain the performance of head predicate classes, and this results in a more even performance for all predicate classes. Moreover, the method is model-agnostic and proven to improve the performance of various biased SGG models, demonstrating the effectiveness of our PPDL method.

References

[1] Jonathon Byrd and Zachary Lipton. What is the effect of importance weighting in deep learning? In International Conference on Machine Learning, pages 872–881. PMLR, 2019. 3

[2] Shizhe Chen, Qin Jin, Peng Wang, and Qi Wu. Say as you wish: Fine-grained control of image caption generation with abstract scene graphs. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 9962–9971, 2020. 1

[3] Tianshui Chen, Weihao Yu, Riquan Chen, and Liang Lin. Knowledge-embedded routing network for scene graph generation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 6163–6171, 2019. 2, 3, 6

[4] Vincent S Chen, Paroma Varma, Ranjay Krishna, Michael Bernstein, Christopher Re, and Li Fei-Fei. Scene graph prediction with limited labels. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 2580–2590, 2019. 3

[5] Meng-Jiun Chiou, Henghui Ding, Hanshu Yan, Changhu Wang, Roger Zimmermann, and Jiashi Feng. Recovering the unbiased scene graphs from the biased ones. In Proceedings of the 29th ACM International Conference on Multimedia, pages 1581–1590, 2021. 6

[6] Vinay Damodaran, Sharanya Chakravarthy, Akshay Kumar, Anjana Umapathy, Teruko Mitamura, Yuta Nakashima, Noa Garcia, and Chenhui Chu. Understanding the role of scene graphs in visual question answering. arXiv preprint arXiv:2101.05479, 2021. 1

[7] Jiuxiang Gu, Shafiq Joty, Jianfei Cai, Handong Zhao, Xu Yang, and Gang Wang. Unpaired image captioning via scene graph alignments. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 10323–10332, 2019. 1

[8] Jiuxiang Gu, Handong Zhao, Zhe Lin, Sheng Li, Jianfei Cai, and Mingyang Ling. Scene graph generation with external knowledge and image reconstruction. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 1969–1978, 2019. 3

[9] Guo Haixiang, Li Yijing, Jennifer Shang, Gu Mingyun, Huang Yuanyue, and Gong Bing. Learning from class-imbalanced data: Review of methods and applications. Expert Systems with Applications, 73:220–239, 2017. 3

[10] Roei Herzig, Amir Bar, Huijuan Xu, Gal Chechik, Trevor Darrell, and Amir Globerson. Learning canonical representations for scene graph to image generation. In European Conference on Computer Vision, pages 210–227. Springer, 2020. 1

[11] Justin Johnson, Agrim Gupta, and Li Fei-Fei. Image generation from scene graphs. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 1219–1228, 2018. 1

[12] Justin Johnson, Ranjay Krishna, Michael Stark, Li-Jia Li, David Shamma, Michael Bernstein, and Li Fei-Fei. Image retrieval using scene graphs. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 3668–3678, 2015. 1, 7

[13] Bingyi Kang, Zhuang Liu, Xin Wang, Fisher Yu, Jiashi Feng, and Trevor Darrell. Few-shot object detection via feature reweighting. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 8420–8429, 2019. 3

[14] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. International journal of computer vision, 123(1):32–73, 2017. 1, 3, 6

[15] Xiangyang Li and Shuqiang Jiang. Know more say less: Image captioning based on scene graphs. IEEE Transactions on Multimedia, 21(8):2117–2130, 2019. 1

[16] Yikang Li, Wanli Ouyang, Bolei Zhou, Kun Wang, and Xiaogang Wang. Scene graph generation from objects, phrases and region captions. In Proceedings of the IEEE International Conference on Computer Vision, pages 1261–1270, 2017. 7

[17] Tsung-Yi Lin, Piotr Dollár, Ross B. Girshick, Kaiming He, Bharath Hariharan, and Serge J. Belongie. Feature pyramid networks for object detection. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 936–944, 2017. 3, 6

[18] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In Proceedings of the IEEE international conference on computer vision, pages 2980–2988, 2017. 2, 3

[19] Hengyue Liu, Ning Yan, Masood Mortazavi, and Bir Bhanu. Fully convolutional scene graph generation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 11546–11556, 2021. 3

[20] Tongliang Liu and Dacheng Tao. Classification with noisy labels by importance reweighting. IEEE Transactions on pattern analysis and machine intelligence, 38(3):447–461, 2015. 3

[21] Robbie T Nakatsu. An evaluation of four resampling methods used in machine learning classification. IEEE Intelligent Systems, 2020. 3

[22] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. Advances in neural information processing systems, 28:91–99, 2015. 3, 6

[23] Brigit Schroeder and Subarna Tripathi. Structured query-based image retrieval using scene graphs. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, pages 178–179, 2020. 1

[24] Sebastian Schuster, Ranjay Krishna, Angel Chang, Li Fei-Fei, and Christopher D Manning. Generating semantically precise scene graphs from textual descriptions for improved image retrieval. In Proceedings of the fourth workshop on vision and language, pages 70–80, 2015. 1

[25] Jiaxin Shi, Hanwang Zhang, and Juanzi Li. Explainable and explicit visual reasoning over scene graphs. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 8376–8384, 2019. 1

[26] Mohammed Suhail, Abhay Mittal, Behjat Siddiquie, Chris Broaddus, Jayan Eledath, Gerard Medioni, and Leonid Sigal. Energy-based learning for scene graph generation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 13936–13945, 2021. 2, 3, 6, 7

[27] Kaihua Tang. A scene graph generation codebase in pytorch, 2020. https://github.com/KaihuaTang/Scene-Graph-Benchmark.pytorch. 6

[28] Kaihua Tang, Yulei Niu, Jianqiang Huang, Jiaxin Shi, and Hanwang Zhang. Unbiased scene graph generation from biased training. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 3716–3725, 2020. 2, 3, 6, 7

[29] Kaihua Tang, Hanwang Zhang, Baoyuan Wu, Wenhan Luo, and Wei Liu. Learning to compose dynamic tree structures for visual contexts. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 6619–6628, 2019. 1, 2, 3, 6, 7

[30] Ruize Wang, Zhongyu Wei, Piji Li, Qi Zhang, and Xuanjing Huang. Storytelling from an image stream using scene graphs. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 34, pages 9185–9192, 2020. 2

[31] Saining Xie, Ross B. Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 5987–5995, 2017. 6

[32] Danfei Xu, Yuke Zhu, Christopher B Choy, and Li Fei-Fei. Scene graph generation by iterative message passing. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 5410–5419, 2017. 3, 6

[33] Ning Xu, An-An Liu, Jing Liu, Weizhi Nie, and Yuting Su. Scene graph captioner: Image captioning based on structural visual representation. Journal of Visual Communication and Image Representation, 58:477–485, 2019. 1

[34] Shaotian Yan, Chen Shen, Zhongming Jin, Jianqiang Huang, Rongxin Jiang, Yaowu Chen, and Xian-Sheng Hua. Pcpl: Predicate-correlation perception learning for unbiased scene graph generation. In Proceedings of the 28th ACM International Conference on Multimedia, pages 265–273, 2020. 3, 6, 7

[35] Gengcong Yang, Jingyi Zhang, Yong Zhang, Baoyuan Wu, and Yujiu Yang. Probabilistic modeling of semantic ambiguity for scene graph generation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 12527–12536, 2021. 3

[36] Jianwei Yang, Jiasen Lu, Stefan Lee, Dhruv Batra, and Devi Parikh. Graph r-cnn for scene graph generation. In Proceedings of the European conference on computer vision (ECCV), pages 670–685, 2018. 3

[37] Xu Yang, Kaihua Tang, Hanwang Zhang, and Jianfei Cai. Auto-encoding scene graphs for image captioning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 10685–10694, 2019. 1

[38] Jing Yu, Yuan Chai, Yujing Wang, Yue Hu, and Qi Wu. Cogtree: Cognition tree loss for unbiased scene graph generation. In Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21, pages 1274–1280, 2021. 2, 3, 6, 7

[39] Lean Yu, Rongtian Zhou, Ling Tang, and Rongda Chen. A dbn-based resampling svm ensemble learning paradigm for credit classification with imbalanced data. Applied Soft Computing, 69:192–202, 2018. 3

[40] Alireza Zareian, Svebor Karaman, and Shih-Fu Chang. Weakly supervised visual semantic parsing. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 3736–3745, 2020. 3

[41] Rowan Zellers, Mark Yatskar, Sam Thomson, and Yejin Choi. Neural motifs: Scene graph parsing with global context. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 5831–5840, 2018. 3, 6, 7

[42] Ji Zhang, Kevin J. Shih, Ahmed Elgammal, Andrew Tao, and Bryan Catanzaro. Graphical contrastive losses for scene graph parsing, 2019. 3

[43] Yiwu Zhong, Liwei Wang, Jianshu Chen, Dong Yu, and Yin Li. Comprehensive image captioning via scene graph decomposition. In European Conference on Computer Vision, pages 211–229. Springer, 2020. 1