Resistance Training Using Prior Bias: Toward Unbiased Scene Graph Generation

Chao Chen [1,2,4,5,6*], **Yibing Zhan** [2], **Baosheng Yu** [3], **Liu Liu** [3], **Yong Luo** [1†], **Bo Du** [1†]

[1] School of Computer Science, Wuhan University
[2] JD Explore Academy
[3] The University of Sydney
[4] Hubei Key Laboratory of Multimedia and Network Communication Engineering
[5] Institute of Artificial Intelligence, Wuhan University
[6] National Engineering Research Center for Multimedia Software, Wuhan University
{chenchao, luoyong, dubo}@whu.edu.cn,
zhanyibing@jd.com, baosheng.yu@sydney.edu.au, liu.liu1@sydney.edu.au,

Abstract

Scene **G**raph **G**eneration (SGG) aims to build a structured representation of a scene using objects and pairwise relationships, which benefits downstream tasks. However, current SGG methods usually suffer from sub-optimal scene graph generation because of the long-tailed distribution of training data. To address this problem, we propose Resistance Training using Prior Bias (RTPB) for the scene graph generation. Specifically, RTPB uses a distributed-based prior bias to improve models' detecting ability on less frequent relationships during training, thus improving the model generalizability on tail categories. In addition, to further explore the contextual information of objects and relationships, we design a contextual encoding backbone network, termed as Dual Transformer (DTrans). We perform extensive experiments on a very popular benchmark, VG150, to demonstrate the effectiveness of our method for the unbiased scene graph generation. In specific, our RTPB achieves an improvement of over 10% under the mean recall when applied to current SGG methods. Furthermore, DTrans with RTPB outperforms nearly all state-of-the-art methods with a large margin. Code is available at https://github.com/ChCh1999/RTPB

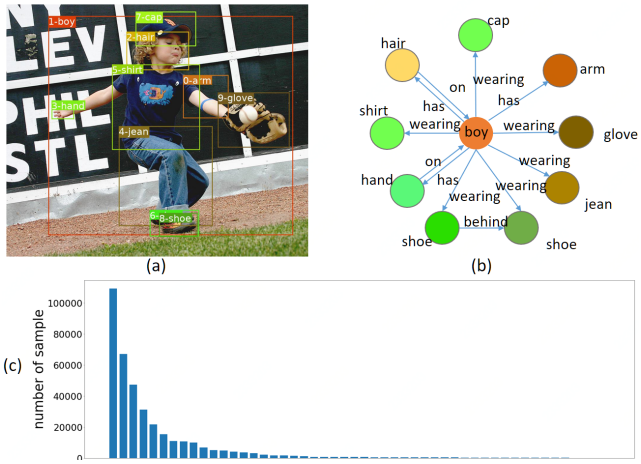


Figure 1: (a) an input image with detected objects. (b) a generated scene graph, which is a graphical representation of the input image with objects and pairwise relationships. (c) the long-tail distribution of object relationships in the Visual Genome (VG) dataset (Krishna et al. 2016).

Introduction

Scene graph generation aims to understand the semantic content of an image via a scene graph, where nodes indicate visual objects and edges indicate pairwise object relationships. An intuitive example of scene graph generation is shown in Figure 1. Scene graph generation is beneficial to bridge the gap between the low-level visual perceiving data and the high-level semantic description. Therefore, a reliable scene graph can provide powerful support for downstream tasks, such as image captioning (Zhong et al. 2020), image retrieval (Johnson et al. 2015), and visual question answering (Tang et al. 2019).

Scene graph generation usually suffers from the long-tail problem of relationships in training data (Tang et al. 2020; Chen et al. 2019b; Zhan et al. 2019). For example, as shown in Figure 1(c), the widely used scene graph generation dataset, Visual Genome (Krishna et al. 2016), is dominated by a few relationships with coarse-grained descriptions (head categories), whereas there are no sufficient annotations available for other less frequent relationships (tail categories). This severe imbalanced distribution makes it difficult for an unbiased scene graph generation, considering there are only a few training samples in real-world scenarios for recognizing less frequent relationships.

To address the above-mentioned issues, many class rebalancing strategies have been introduced for unbiased scene graph generation (Tang et al. 2020; Li et al. 2021a). However, existing methods still struggle to achieve satisfactory performance on tail categories, and more sophisticated solutions are desirable. Inspired that humans increase muscle strength by making their muscles work against a force, we propose resistance training using prior bias (RTPB) to

*This work was done when Chao Chen was a research intern at JD Explore Academy.
†Corrsponding Author.

improve the detection of less-frequently relationships and address the long-tail problem in SGG. Specifically, RTPB assigns a class-specific resistance to the model during training, where the resistance on each relationship is determined by a prior bias, named resistance bias. The resistance bias is first initialized by the prior statistic about the relationships in the training set. The motivation of a resistance bias is to enforce the model to strengthen the ability on less frequent relationships, since they are always corresponding to heavier resistance. In this way, RTPB enables the model to resist the influence of the imbalanced training dataset. Furthermore, to better explore global features for recognizing the relationships, we design a contextual encoding backbone network, named dual Transformer (DTrans for short). Specifically, the DTrans use two stacks of the Transformers to encode the global information for objects and relationships sequentially. To evaluate the effectiveness of the proposed RTPB, we introduce it into recent state-of-the-art methods and our DTrans. Specifically, RTPB can bring an improvement of over 10% in terms of the mean recall criterion compared with other methods. By integrating with the DTrans baseline, RTPB achieves a new state-of-the-art performance for the unbiased scene graph generation.

The main contributions of this paper are summarized as follows: 1) We propose a novel resistance training strategy using prior bias (RTPB) to improve the model generalizability on the rare relationships in the training dataset; 2) We define a general form of resistance bias and devise different types of specific bias by using different ways to model object relationship; 3) We introduce a contextual encoding structure based on the transformer to form a simple yet effective baseline for scene graph generation. Extensive experiments on a very popular benchmark demonstrate significant improvements of our RTPB and DTrans compared with many competitive counterparts for unbiased SGG.

Related Work

Scene Graph Generation

Scene graph generation belongs to visual relationship detection and uses a graph to represent visual objects and the relationships between them (Johnson et al. 2015). In early works, detection and prediction are performed only based on specific objects or object pairs (Lu et al. 2016), which can't make full use of the semantic information in the input image. To take the contextual information into consideration, later works utilize the whole image by designing various network architectures, such as biLSTM (Zellers et al. 2018), TreeLSTM (Tang et al. 2019) and GNN (Yang et al. 2018; Li et al. 2021b). Some other works try to utilize external information, such as linguistic knowledge and knowledge graph, to further improve scene graph generation (Lu et al. 2016; Chen et al. 2019b; Yu et al. 2017; Gu et al. 2019).

However, due to the annotator preference and image distribution, this task suffers from a severe data imbalance issue. Dozens of works thus try to address this issue, and the main focus is on reducing the influence of long-tail distribution and building a balanced model for relationship prediction. Tang et al. also try to use causal analysis (Tang et al. 2020) to reduce the influence of training data distribution on the final model. Some other works (Chen et al. 2019a; Zhan et al. 2020; Chiou et al. 2021) address this issue in a positive-unlabeled learning manner, and typical imbalance learning methods, such as re-sampling and cost-sensitive learning, are also introduced for scene graph generation (Li et al. 2021a; Yan et al. 2020). Unlike these approaches, we adopt the resistance training strategy using prior bias (RTPB), which utilizes a resistance bias item for the relationship classifier during training to optimize the loss value and classification margin of each type of relationship.

Imbalanced Learning

Imbalanced learning methods can be roughly divided into two categories: resampling and cost-sensitive learning.

Resampling Resampling methods change the class ratio to make the dataset a balanced one. These methods can be divided into two categories: oversampling and undersampling. Oversampling increases the count of the less frequent classes (Chawla et al. 2002), and hence may lead to overfitting for minority classes. Undersampling reduces the sample of major categories, and thus is not feasible when data imbalance is severe since it discards a portion of valuable data (Liu, Wu, and Zhou 2008).

Cost-sensitive Learning Cost-sensitive learning assigns different weights to samples of different categories (Elkan 2001). Re-weighting is a widely used cost-sensitive strategy by using prior knowledge to balance the weights across categories. Early works use the reciprocal of their frequency or a smoothed version of the inverse square root of class frequency to adjust the training loss of different classes. However, this tends to increase the difficulty of model optimization under extreme data imbalanced settings and large-scale scenarios. In (Cui et al. 2019), class balanced (CB) weight is defined as the inverse effective number of samples. Some other approaches dynamically adjust the weight for each sample based on the model's performance (Lin et al. 2017; Li, Liu, and Wang 2019; Cao et al. 2019).

Different from these approaches, which directly deal with the loss function, we add a prior bias on the classification logits for each class based on the distribution of training dataset. In this way, we provide a new approach to import additional category-specific information and address the imbalance problem through adjusting the classification boundaries. A similar idea is proposed in (Menon et al. 2020) for long-tailed recognition, and we differ from them significantly in the tasks and formulations.

Methods

Problem Setting and Overview

Problem Setting Given an image $\mathcal{I}$, our goal is to predicate a graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, where $\mathcal{V}$ is the set of objects in the image and $\mathcal{E}$ is pairwise relationships of objects in $\mathcal{V}$. Each object $v \in \mathcal{V}$ consists of the bounding box coordinates $\boldsymbol{b}_v$ for location and the class label c_v for type. An edge $e \in \mathcal{E}$ include the pair of a subject and a object, *i.e.*, (v_s, v_o), and the label of the relationship r between v_s and v_0. Typically,

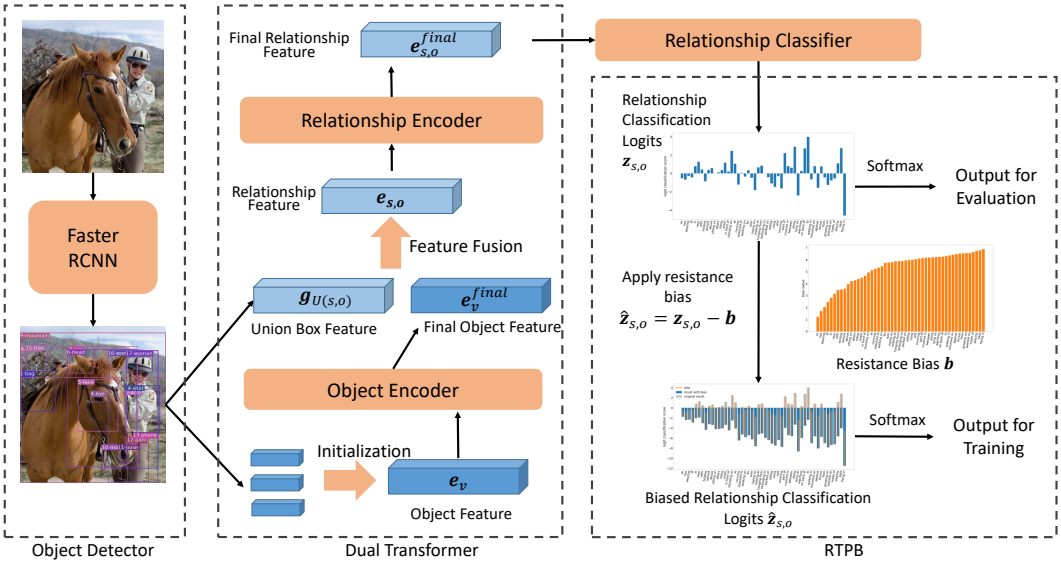


Figure 2: The overall structure of our method. We adopt Faster RCNN as the object detector. Dual Transformer is used to encode object/relationship features. RTPB addresses the long-tail problem by adjusting the classification logits of each relationship.

for object $v \in \mathcal{V}$, the label c_v belongs to $\mathcal{C}_e = \{1, 2, ..., L_e\}$, where L_e is the number of object labels. The label of relationship c_r are in $\mathcal{C}_r = \{1, 2, ..., L_r\}$, where L_r is the number of relationship labels.

In the commonly used SGG pipeline, the probability of scene graph $Pr(\mathcal{G}|\mathcal{I})$ is formulated as:

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

where $Pr(\mathcal{B}|\mathcal{I})$ indicates the proposal generation. The $\mathcal{B}$ is the set of bounding box b_v, which is conducted by an object detector. $Pr(\mathcal{O}|\mathcal{I}, \mathcal{B})$ denotes the object classification. The $\mathcal{O}$ is the set of object class c_v, and $Pr(\mathcal{G}|\mathcal{I}, \mathcal{B}, \mathcal{O})$ means the final relationship classification.

Methods Overview Figure 2 shows the overall structure of our method. Following previous works (Tang et al. 2020; Zellers et al. 2018), we utilize Faster RCNN (Ren et al. 2015) to generate object proposals and corresponding features from the input image. For each object proposal, the object detector conducts the bounding box coordinates $\boldsymbol{b}_v$, visual feature $\boldsymbol{g}_v$, and object classification score $\boldsymbol{z}_v \in \mathbb{R}^{|\mathcal{C}_v|}$. Besides, to improve the relationship prediction, the backbone also generates the feature of the union boxes of each pair of objects. Then, we introduce our *Dual Transformer (DTrans)*, which uses two stacks of Transformers to encode the context-aware representations for objects and relationships, respectively. Moreover, to address the long-tail problem for SGG, we adopt the ***r**esistance **t**raining using **p**rior **b**ias (RTPB)* for unbiased SGG. In the training phase, the RTPB assigns resistance to the model by the prior bias, named resistance bias.

Dual Transformer

We propose the Dual Transformer (DTrans) for better contextual information encoding. As shown in Figure 2, based on the outputs of the object detector, i.e., the location of objects, the feature of objects, and the feature of union boxes, the DTrans uses self-attention to encode the context-aware representation of objects and relationships.

Self-attention Our model uses two stacks of transformers to obtain the contextual information for objects and the corresponding pairwise relationships. Each transformer encodes the input features with self-attention mechanisms (Vaswani et al. 2017), which mainly consists of the attention and the feed-forward network. The attention matrix is calculated as

$$Attention(Q, K, V) = \sigma(\frac{QK^T}{\sqrt{d_k}})V, \quad (2)$$

where query (Q), keys (K), and value (V) are obtained from the input feature through three different linear transformation layers, $1/\sqrt{d_k}$ is the scaling factor for the dot product of Q and K (Vaswani et al. 2017), and σ is the softmax function. Besides, we utilize the multi-head attention, which is used to divide the Q, K, and V into n_h parts and calculates the attentions of each part. Next, a feed-forward network (FFN) is used to mix the attentions of each part up.

Object Encoder As shown in Figure 2, based on all the results about object proposals from the object detector, we first compose them to initialize a feature vector of the corresponding object as follows:

$$\mathbf{e}_v = W_o[pos(\boldsymbol{b}_v), \boldsymbol{g}_v, ebd(c_v)], \quad (3)$$

where $[\cdot, \cdot]$ denotes the concatenation operation, the pos is a learnable position encoding method for object location, $ebd(c_v)$ is the GloVe vector embedding for object label c_v, and W_o is a linear transformation layer used to initialize the object feature from the above information. Then, we feed the

$\mathbf{e}_v$ into a stack of n_o transformers to obtain the final context-aware features $\mathbf{e}_v^{final}$ for objects. The $\mathbf{e}_v^{final}$ is first used to produce the final classification result of objects as follows:

$$\mathbf{p}_v = \sigma(W_{clf}^o \mathbf{e}_v^{final}), \tag{4}$$

where W_{clf}^o is the object classifier. And then, the $\mathbf{e}_v^{final}$ is also used to generate the features of the pairwise relationships of objects in the later part of our model.

Relationship Encoder Before encoding the features of relationships, we first carry out a feature fusion operation to generate the basic representation of relationships. For each directed object pair (s, o), we concatenate the visual feature $\mathbf{g}_{U(s,o)}$ of the union box of s and o, and the context-aware features of the subject s and the object o. Then, we use a linear transformation to obtain the representation of the relationship between the subject s and the object o:

$$\mathbf{e}_{s,o} = W_r[\mathbf{g}_{U(s,o)}, \mathbf{e}_s^{final}, \mathbf{e}_o^{final}], \tag{5}$$

where W_r is a linear transformation layer used to compress the relationship features. $\mathbf{e}_s^{final}$ and $\mathbf{e}_o^{final}$ are the final context-aware features of subject s and object o.

Then, we use another stack of n_r Transformers to encode the features of relationships and produce the final feature $\mathbf{e}_{s,o}^{final}$ for the relationship between subject s and object o. Finally, the relationship classifier uses the feature $\mathbf{e}_{s,o}^{final}$ to recognize the pairwise relationships as follows

$$\mathbf{p}_{s,o} = \sigma(\mathbf{z}_{s,o}) = \sigma(W_{cls}^r \mathbf{e}_{s,o}^{final}), \tag{6}$$

where $\mathbf{z}_{s,o} = W_{cls}\mathbf{e}_{s,o}^{final}$ is vector of the logits for relationship classification, andW_{cls}^r is the relationship classifier.

Resistance Training using Prior Bias

Inspired by the idea of resistance training that muscle turns to be stronger if they are training with heavier resistance (Kraemer and Ratamess 2004), we propose the resistance training using prior bias (RTPB) for unbiased SGG. Our RTPB uses the prior bias as the resistance for the model only during training to adjust the model's strength for different relationships. We name the prior bias as resistance bias, which is applied to the model in the training phase as:

$$\hat{\mathbf{z}}_{s,o} = \mathbf{z}_{s,o} - \mathbf{b}, \tag{7}$$

where $\mathbf{z}_{s,o} = [z_1, z_2, ..., z_{L_r}]$ is the vector of logits for relationship classification in Eq. (6), and $\mathbf{b} = [b_1, b_2, ..., b_{L_r}]$ is the vector of resistance bias for each relationship. By assigning relatively heavier resistance to the tail relationships during training, we enable the model without resistance to better handle the tail relationships and obtain a balanced performance between the tail and head relationships.

In the following, we first introduce four resistance biases from the prior statistic about the training set. Then, we analyze how RTPB works through the loss and the objective.

Instances for Resistance Bias We define the basic form of resistance bias b_i for relationship $i \in \mathcal{C}_r$ as follows:

$$b_i = -log\left(\frac{w_i^a}{\sum_{j\in\mathcal{C}_r} w_j^a} + \epsilon\right), \tag{8}$$

where w_i is the weight of relationship i, and a, ϵ are the hyper-parameters used to adjust the distribution of b_i. For resistance bias, the category weight w_i is negatively correlated to the resistance bias b_i. Therefore, to assign the head categories of relationship small resistance, distribution of the weight w_i should be correlated to the long-tail distribution of the training set. We introduce four resistance biases:

Count Resistance Bias (CB) Intuitively, for the general classification task, we can set the bias weight w_i of resistance bias with the proportion of sample for each relationship in the training set. We denote this type of resistance bias as **c**ount resistance **b**ias (CB).

Valid Resistance Bias (VB) Besides the number of samples, the long-tail problem also occurs between the general descriptions and the detailed ones. Therefore, we formulate valid pair as follows: if there is relationship $i \in \mathcal{C}_r$ for object pair (s, o) in the training set, the (s, o) is a valid pair of relationship i. The distribution of valid pair count for the popular SGG data set, Visual Genome (Krishna et al. 2016), is shown in the Figure.3. We set the weight w_i as the proportion of valid pair for relationship i. In this way, the RTPB can distinguish between the general descriptions of relationships and the rare ones. We call this resistance bias **V**alid Resistance **B**ias (VB).

Pair Resistance Bias (PB) However, for the SGG task, the classification of relationship is related to not only the relationship, but the pair of entities. Thus we try to use the relationship distribution of each object pair. For different object pairs, the bias are different. In application, we select the bias based on the classification result of objects. For object pair (s, o), the resistance bias for relationship i is follows:

$$b_{s,o,i} = -log\left(\frac{w_{s,o,i}^a}{\sum_{j\in\mathcal{C}_r} w_{s,o,j}^a} + \epsilon\right), \tag{9}$$

where $w_{s,o,i}$ s the ratio of the relationship i in all the relationships between subject s and object o in the training set. And we call this resistance bias as **P**air Resistance **B**ias (PB)

Estimated Resistance Bias (EB) Because many pairs of s, o have only a little number of valid relationships and few annotations, the number of relationship sample for particular object pair can hardly show the general distribution of relationship in many cases. Thus we propose subject-predicate and predicate-object count n^{sppo} to estimate the relationship distribution. For subject s, object o and relationship i, it's calculated as

$$n_{s,o,i}^{sppo} = \sqrt{\sum_{o'\in\mathcal{C}_e} n_{s,o',i} \times \sum_{s'\in\mathcal{C}_e} n_{s',o,i}}, \tag{10}$$

where $n_{s,o,i}$ is the count of relationship i between subject s and object o. Then we produce a new resistance bias by replacing the weight $w_{s,o,i}$ in Eq. (9) with the proportion $n_{s,o,i}^{sppo}/\sum_{j\in\mathcal{C}_r} n_{s,o,j}^{sppo}$. We name this type of resistance as **E**stimated Resistance **B**ias (EB).

Loss with RTPB In this section, we analyze the effect of RTPB through classification loss. In the classifier, we conduct the final relationship classification probability by a softmax function, and we use the cross-entropy (CE) to evaluate

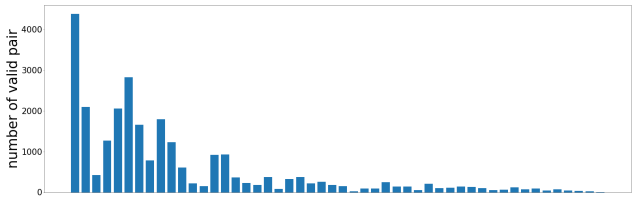


Figure 3: The distribution of the valid pair count.

the classification optimization objective. The classification probability and CE loss are shown in follows,

$$p_i = Pr(i|s,o,\mathcal{I}) = \sigma(\mathbf{z}_{s,o})_i = \frac{e^{z_i}}{\sum_{j\in\mathcal{C}_r} e^{z_j}}, \tag{11}$$

$$\mathcal{L}_i = -log(p_i) = -log\frac{e^{z_i}}{\sum_{j\in\mathcal{C}_r} e^{z_j}}, \tag{12}$$

where p_i is the probability that the relationship between object pair (s,o) in image $\mathcal{I}$ is i, and $\mathbf{z}_{s,o}$is the vector of logits for relationship classification mentioned in Eq. (6). If we apply our RTPB, the probability p_i in Eq. (11) turns to be:

$$\hat{p}_i = \sigma(\mathbf{z}_{s,o} - \mathbf{b})_i = p_i\frac{e^{-b_i}}{\sum_{j\in\mathcal{C}_r} e^{-b_j}p_j},$$

where $\mathbf{b} = [b_1, b_2, ..., b_{L_r}]$ is vector of resistance bias. And the softmax cross-entropy loss turns to be:

$$\begin{aligned}\hat{\mathcal{L}}_i &= -log(\hat{p}_i) = -log\frac{e^{z_i-b_i}}{\sum_{j\in\mathcal{C}_r} e^{z_j-b_j}} \\ &= -log\frac{e^{z_i}}{\sum_{j\in\mathcal{C}_r} e^{z_j}} + b_i + log\sum_{j\in\mathcal{C}_r}\frac{e^{-b_j}e^{z_j}}{\sum_{k\in\mathcal{C}_r} e^{z_k}} \\ &= \mathcal{L}_i + \theta_i,\end{aligned} \tag{13}$$

where $\theta_i = b_i + log\sum_{j\in\mathcal{C}_r} e^{-b_j}p_j$. For the loss function, RTPB can be regard as a dynamic re-weighting method based on the resistance bias b_i and the relationship prediction p_i. And the weight for the relationship i between (s,o) is θ_i in Eq. (13). Because $\frac{d\theta_i}{db_i} = 1 - \frac{e^{-b_i}Pr(i|s,o,I)}{\sum_{j\in\mathcal{C}_r} e^{-b_j}Pr(j|s,o,I)} > 0$, the RTPB can up-weight the relationships that correspond to larger resistance bias while down-weighting the rest. In this way, RTPB reduces the loss contribution from head relationships and highlight the tail relationships to keep a balance between head and tail. Take binary classification as an example, and we can visualize the loss value for the pair with a specific resistance bias value; as shown in the Appendix.

Moreover, the loss value tends to be larger when the predicted distribution of p_i in Eq. (11) is close to the distribution of $e^{-b_i} = \frac{w_i^a}{\sum_{j\in\mathcal{C}_r} w_j^a} + \epsilon$, which is correlated to the distribution of the predefined weight ω_i of each relationship in Eq. (8). Because the distribution of ω_i correlates to the long-tail distribution of the training set, the resistance bias provides continuous supervision against the long-tailed distribution to avoid biased prediction.

Objective with RTPB From another point of view, we can simply regard the $\hat{\mathbf{z}}_{s,o} = \mathbf{z}_{s,o} - \mathbf{b}$ as a joint to be optimized in the training phase and the result should be close to the baseline without RTPB. We then get the final prediction $\mathbf{z}_{s,o} = \hat{\mathbf{z}}_{s,o} + \mathbf{b}$. In this manner, compared with the model without RTPB, we tend to judge the ambiguous predictions as belonging to another one with a larger resistance bias. Thus the classification boundary of two classes is tilted to the one with larger resistance. The tilt depends on the relative value relationship of resistance bias between the two relationship categories. For example, if we judge the label of a relationship as i, the $\hat{z}_i$ should outperform all of the rest $\{\hat{z}_j, j \neq i\}$ by a necessary classification margin $b_j - b_i$.

Experiments

In this section, we evaluate our method on the most popular scene graph generation dataset, Visual Genome (VG). To demonstrate the effectiveness of our method, we compare it with several recent methods, including MOTIFS, VCTree, and BGNN. We also perform comprehensive ablation studies on different components and provide a discussion on different cost-sensitive methods for imbalance learning.

Dataset and Evaluation Metrics

Dataset We perform extensive experiments on Visual Genome (VG) (Krishna et al. 2016) dataset. Similar to previous work (Xu et al. 2017; Zellers et al. 2018; Tang et al. 2020), we use the widely adapted subset, VG150, which consists of the most frequent 150 object categories and 50 predicate categories. The original split only has training set (70%) and test set (30%). We follow (Tang et al. 2020) to sample a 5k validation set for parameter tuning.

Evaluation Metrics For a fair comparison, we follow previous work (Zellers et al. 2018; Tang et al. 2020; Li et al. 2021a) and evaluate the proposed method on three sub-tasks of scene graph generation as follows.

1. Predicate Classification (**PredCls**). On this task, the input includes not only the image, but the bounding boxes of objects and the labels of objects;
2. Scene Graph Classification (**SGCls**). It only gives the bounding boxes;
3. Scene Graph Detection (**SGDet**). On this task, no additional information other than the raw image will be given.

For each sub-task, previous works use Recall@k (or R@k) to report the performance of scene graph generation (Lu et al. 2016; Xu et al. 2017), which indicates the recall rate of ground-truth relationship triplets (subject-predicate-object) among the top k predictions of each test image. Considering that the above-mentioned metric (R@k) caters to the bias caused by the long-tailed distribution of object relationships, the model can only achieve a high recall rate by predicting the frequent relationships. Therefore, it can not demonstrate the effectiveness of different models on less frequent relationships. To this end, the mean recall rate of each relationships, (mR@k), has been widely used in recent works (Zellers et al. 2018; Tang et al. 2019, 2020; Li et al. 2021a).

Models	Predcls			SGcls			SGdet		
	mR@20	mR@50	mR@100	mR@20	mR@50	mR@100	mR@20	mR@50	mR@100
KERN‡(Chen et al. 2019b)	-	17.7	19.2	-	9.4	10.0	-	6.4	7.3
PCPL‡(Yan et al. 2020)	-	35.2	37.8	-	18.6	19.6	-	9.5	11.7
GPS-Net‡(Lin et al. 2020)	17.4	21.3	22.8	10.0	11.8	12.6	6.9	8.7	9.8
BGNN(Li et al. 2021a)	-	30.4	32.9	-	14.3	16.5	-	10.7	12.6
MOTIFS†(Zellers et al. 2018)	13.6	17.2	18.6	7.8	9.7	10.3	5.6	7.7	9.2
MOTIFS+TDE(Tang et al. 2020)	18.5	24.9	28.3	11.1	13.9	15.2	6.6	8.5	9.9
MOTIFS+DLFE(Chiou et al. 2021)	22.1	26.9	28.8	12.8	15.2	15.9	8.6	11.7	13.8
MOTIFS + **RTPB(CB)**	28.8	35.3	37.7	16.3	19.4	20.6	9.7	13.1	15.5
VCTree†(Tang et al. 2019)	13.4	16.8	18.1	8.5	10.8	11.5	5.4	7.4	8.6
VCTree+TDE(Tang et al. 2020)	17.2	23.3	26.6	8.9	11.8	13.4	6.3	8.6	10.3
VCTree + DLFE(Chiou et al. 2021)	20.8	25.3	27.1	15.8	18.9	20.0	8.6	11.8	13.8
VCTree+**RTPB(CB)**	27.3	33.4	35.6	**20.6**	**24.5**	**25.8**	9.6	12.8	15.1
DTrans	15.1	19.3	21.0	9.9	12.1	13.0	6.6	9.0	10.8
DTrans+**RTPB(CB)**	**30.3**	**36.2**	**38.1**	19.1	21.8	22.8	**12.7**	**16.5**	**19.0**
DTrans+**RTPB(VB)**	25.5	31.1	33.3	17.3	20.1	21.3	11.9	15.7	18.4
DTrans+**RTPB(PB)**	17.4	21.6	23.1	11.9	14.0	14.7	7.7	10.1	11.9
DTrans+**RTPB(EB)**	22.7	26.7	28.4	15.2	17.4	18.2	11.0	14.1	16.1

Table 1: The performance on VG (Krishna et al. 2016) under graph constraints setting. † indicates the results reproduced using the code of (Tang et al. 2020). ‡ models are with VGG16 backbone (Simonyan and Zisserman 2015), while others are with ResNeXt-101-FPN backbone (Lin et al. 2017). (CB), (VB), (PB), and (EB) indicate the count resistance bias, the valid resistance bias, the pair resistance bias, and the estimated resistance bias, respectively.

Therefore, we report the performance using the mean Recall, and the results of Recall are available in the appendix.

Implementation Details

We implement the proposed method using PyTorch (Paszke et al. 2019). For a fair comparison, we use the object detection model similar to (Tang et al. 2020), i.e., a pretrained Faster RCNN with ResNeXt-101-FPN as the backbone network. The object detector is fine-tuned on the VG dataset. Then, we froze the object detector and train the rest parts of the SGG model. For the DTrans, the number of object encoder layers is $n_o = 4$ and the number of relationship encoder layers is $n_r = 2$. For the proposed resistance bias, we use $a = 1$ and $\epsilon = 0.001$ if not otherwise stated. For the background relationship, the bias is a constant value $log\frac{1}{\|\mathcal{C}_r\|}$, where the $\mathcal{C}_r$ is the set of relationship labels. We perform our experiments using a single NVIDIA V100 GPU. We train the DTrans model for 18000 iterations with batch size 16. Specifically, it takes around 24 hours for the SGDet task, and less than 12 hours for the PredCls/SGCls task. For the SGDet task, we use all pairs of objects for training instead of those pair of objects with overlap. For other experimental settings, we always keep the same with the previous work (Tang et al. 2020).

Comparison with Recent Methods

To demonstrate the effectiveness of the proposed method, we apply RTPB on several representative methods for scene graph generation, including MOTIFS, VCTree, and our DTrans baseline model. As shown in Table 1, we find that RTPB can achieve consistent improvements in the mean recall metric for all of MOTIFS, VCTree, and our DTrans. Models equipped with our RTPB achieve new state-of-the-art and out-preform previous methods by a clear margin.

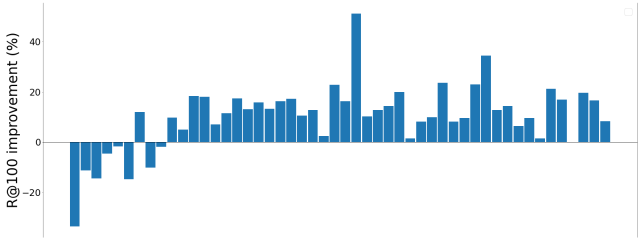


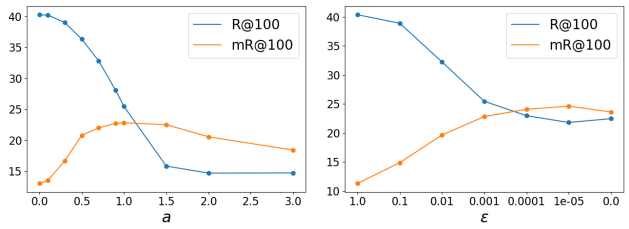
Figure 4: The improvements of Recall@100 on the SGDet task using RTPB(CB). The number of relationships decreases from left to right. It shows that RTPB achieves higher recall rate on most relationships with a sacrifice on only a small number of frequent relationships.

Take SGDet as an example, as shown in Figure 4, the recall rate on most relationships has been significantly improved (DTrans w/o or w/ RTPB). Specifically, RTPB may also lead to slight performance degradation on the frequent relationships without over-fitting on the head categories. Therefore, we see a clear improvement when using the mean recall rate as the evaluation metric, which demonstrates RTPB can build a better-balanced model for unbiased SGG.

Among four instance of resistance bias, CB performs the best on mR because it simply calculates the relationship distribution based on the whole dataset. VB/PB/EB are proposed based on more critical conditions. Although we believe that adding more critical conditions would lead to more practical prior bias, currently, limited data are not sufficient to obtain proper empirical bias under the given conditions.

Ablation Studies

In this subsection, we evaluate the influence of important hyper-parameters (a and ϵ) and an extended version of resis-



(a) (b)

Figure 5: Influence of (a) different a and (b) different ϵ.

tance bias, i.e., soft resistance bias. Besides, we compare our RTPB with several conventional imbalance learning methods on the SGG task.

Hyper-parameters for Resistance Bias Two important hyper-parameters in RTPB, a and ϵ are used to control the distribution of the resistance bias. Specifically, $a > 0$ is used to adjust the divergence of the bias, while a small $\epsilon \in [0, 1]$ is used to control the maximum relative difference. We perform experiments on the SGCls task to evaluate the influence of a and ϵ in the RTPB. As shown in Figure 5(a), when increasing a, the overall recall rate decreases and the mean recall rate increases; when $a > 1.0$, the mean recall rate also decreases. Therefore, we use $a = 1.0$ in our experiments. As shown in Figure 5(b), we find that $\epsilon = 1e{-}3$ achieves a good trade-off between the overall recall rate and the mean recall rate. Furthermore, we find out that when either $a = 0$ or $\epsilon \leq 1$, the performance is very close to the baseline method because the resistance biases for different relationships are equal to each other.

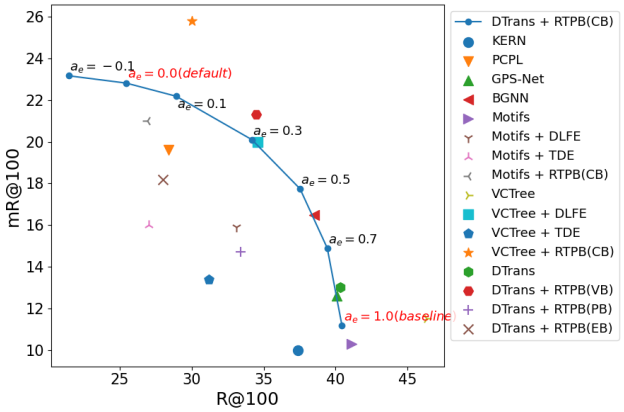


Figure 6: The result of inference with different a_e.

Soft Resistance Bias To better understand how RTPB addresses the trade-off between head and tail relationships, we introduce a soft version of resistance bias during inference phase as follows.

$$b_i^e = -log(\frac{w_i^{a_e}}{\sum_{j \in \mathcal{C}_r} w_j^{a_e}} + \epsilon), \tag{14}$$

Loss	mR@20	mR@50	mR@100
Baseline	9.9	12.2	13.0
$\mathcal{L}_{Reweight}$	9.9	14.3	16.8
$\mathcal{L}_{ClsBal}$	12.8	15.5	17.0
$\mathcal{L}_{focal}$	9.1	11.3	12.1
$\mathcal{L}_{LDAM}$	8.4	10.3	10.8
RTPB(CB)	**19.1**	**21.8**	**22.8**

Table 2: Comparison with other cost-sensitive methods.

where $a_e \leq a$, and b_i^e is the smooth version of resistance bias. By applying a smoothed resistance bias in the inference phase, we can evaluate different impacts of RTPB on the trained model. Following the analysis of the resistance bias for adjusting the classification boundary, a smooth resistance bias reduces the margin areas of head categories from $alog\frac{w_i}{w_j}$ to $(a - a_e)log\frac{w_i}{w_j}$. Specifically, 1) when $a_e = 0$, the result is same as the original one; and 2) when $a_e = a = 1$, the result is close to the baseline without RTPB, i.e., Motifs (Zellers et al. 2018) and DTrans. The experimental results of different a_e are shown in Figure 6. The model used for evaluation is trained with CB and $a = 1$. As the a_e decreases, the classification margin of head categories keeps becoming larger. Therefore, the model performs worse on the few head categories but performs better on the rest tail categories. For the evaluation result, this means a consistent improvement in mean recall metric as shown in Figure 6. Besides, the model trained with RTPB can achieve similar performance as many current SGG methods with different a_e. Thus RTPB can be regarded as a general baseline for different unbiased SGG methods.

Other Cost-sensitive Methods

As shown in Table 2, we compare our method with the following cost-sensitive methods on the SGCls task, using our DTrans as the backbone. 1) **Re-weighting Loss**: using the fraction of the count of each class as the weight of loss. 2) **Class Balanced Re-weighting Loss** (Cui et al. 2019): using the effective number of samples to re-balance the loss. 3) **Focal Loss** (Lin et al. 2017): using the focal weight to adjust the losses for well-learned samples and hard samples. 4) **Label Distribution-Aware Margin Loss** (Cao et al. 2019): using the prior margin for each class to balance the model.

We evaluate these methods on the SGCls task with our DTrans. As shown in Table 2, our method outperforms the rest of the methods by a clear margin in the mean recall.

Conclusion

In this paper, we propose a novel resistance training strategy using prior bias (RTPB) for unbiased SGG. Experimental results on the popular VG dataset demonstrate that our RTPB achieves a better head-tail trade-off for the SGG task than existing counterparts. We also devise a novel transformer-based contextual encoding structure to encode global information for visual objects and relationships. We obtain significant improvements over recent state-of-the-art approaches, and thus set a new baseline for unbiased SGG.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China under Grants 61822113, 41871243, 62002090, and the Science and Technology Major Project of Hubei Province (Next-Generation AI Technologies) under Grant 2019AEA170. Dr. Baosheng Yu is supported by ARC project FL-170100117. Dr. Liu Liu is supported by ARC project DP-180103424.

References

Cao, K.; Wei, C.; Gaidon, A.; Aréchiga, N.; and Ma, T. 2019. Learning Imbalanced Datasets with Label-Distribution-Aware Margin Loss. In Wallach, H. M.; Larochelle, H.; Beygelzimer, A.; d'Alché-Buc, F.; Fox, E. B.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, 1565–1576.

Chawla, N. V.; Bowyer, K. W.; Hall, L. O.; and Kegelmeyer, W. P. 2002. SMOTE: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16: 321–357.

Chen, D.; Liang, X.; Wang, Y.; and Gao, W. 2019a. Soft Transfer Learning via Gradient Diagnosis for Visual Relationship Detection. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 1118–1126. IEEE.

Chen, T.; Yu, W.; Chen, R.; and Lin, L. 2019b. Knowledge-Embedded Routing Network for Scene Graph Generation. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, 6163–6171. Computer Vision Foundation / IEEE.

Chiou, M.-J.; Ding, H.; Yan, H.; Wang, C.; Zimmermann, R.; and Feng, J. 2021. Recovering the Unbiased Scene Graphs from the Biased Ones. *arXiv preprint arXiv:2107.02112*.

Cui, Y.; Jia, M.; Lin, T.; Song, Y.; and Belongie, S. J. 2019. Class-Balanced Loss Based on Effective Number of Samples. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, 9268–9277. Computer Vision Foundation / IEEE.

Elkan, C. 2001. The Foundations of Cost-Sensitive Learning. In *Proceedings of the 17th International Joint Conference on Artificial Intelligence - Volume 2*, IJCAI'01, 973–978. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc. ISBN 1558608125.

Gu, J.; Zhao, H.; Lin, Z.; Li, S.; Cai, J.; and Ling, M. 2019. Scene Graph Generation With External Knowledge and Image Reconstruction. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, 1969–1978. Computer Vision Foundation / IEEE.

Johnson, J.; Krishna, R.; Stark, M.; Li, L.; Shamma, D. A.; Bernstein, M. S.; and Li, F. 2015. Image retrieval using scene graphs. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, 3668–3678. IEEE Computer Society.

Kraemer, W. J.; and Ratamess, N. A. 2004. Fundamentals of resistance training: progression and exercise prescription. *Medicine & science in sports & exercise*, 36(4): 674–688.

Krishna, R.; Zhu, Y.; Groth, O.; Johnson, J.; Hata, K.; Kravitz, J.; Chen, S.; Kalantidis, Y.; Li, L.; Shamma, D. A.; Bernstein, M. S.; and Li, F. 2016. Visual Genome: Connecting Language and Vision Using Crowdsourced Dense Image Annotations. *CoRR*, abs/1602.07332.

Li, B.; Liu, Y.; and Wang, X. 2019. Gradient Harmonized Single-Stage Detector. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01): 8577–8584.

Li, R.; Zhang, S.; Wan, B.; and He, X. 2021a. Bipartite Graph Network with Adaptive Message Passing for Unbiased Scene Graph Generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 11109–11119.

Li, Y.; Yu, J.; Zhan, Y.; and Chen, Z. 2021b. Relationship graph learning network for visual relationship detection. In *Proceedings of the 2nd ACM International Conference on Multimedia in Asia*, 1–7.

Lin, T.; Goyal, P.; Girshick, R. B.; He, K.; and Dollár, P. 2017. Focal Loss for Dense Object Detection. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, 2999–3007. IEEE Computer Society.

Lin, X.; Ding, C.; Zeng, J.; and Tao, D. 2020. GPS-Net: Graph Property Sensing Network for Scene Graph Generation. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, 3743–3752. IEEE.

Liu, X.-Y.; Wu, J.; and Zhou, Z.-H. 2008. Exploratory undersampling for class-imbalance learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 39(2): 539–550.

Lu, C.; Krishna, R.; Bernstein, M.; and Fei-Fei, L. 2016. Visual Relationship Detection with Language Priors. In Leibe, B.; Matas, J.; Sebe, N.; and Welling, M., eds., *Computer Vision – ECCV 2016*, 852–869. Cham: Springer International Publishing. ISBN 978-3-319-46448-0.

Menon, A. K.; Jayasumana, S.; Rawat, A. S.; Jain, H.; Veit, A.; and Kumar, S. 2020. Long-tail learning via logit adjustment. *CoRR*, abs/2007.07314.

Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; Desmaison, A.; Köpf, A.; Yang, E.; DeVito, Z.; Raison, M.; Tejani, A.; Chilamkurthy, S.; Steiner, B.; Fang, L.; Bai, J.; and Chintala, S. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In Wallach, H. M.; Larochelle, H.; Beygelzimer, A.; d'Alché-Buc, F.; Fox, E. B.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, 8024–8035.

Ren, S.; He, K.; Girshick, R. B.; and Sun, J. 2015. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In Cortes, C.; Lawrence, N. D.; Lee,

D. D.; Sugiyama, M.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, 91–99.

Simonyan, K.; and Zisserman, A. 2015. Very Deep Convolutional Networks for Large-Scale Image Recognition. In Bengio, Y.; and LeCun, Y., eds., *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Tang, K.; Niu, Y.; Huang, J.; Shi, J.; and Zhang, H. 2020. Unbiased Scene Graph Generation From Biased Training. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, 3713–3722. IEEE.

Tang, K.; Zhang, H.; Wu, B.; Luo, W.; and Liu, W. 2019. Learning to Compose Dynamic Tree Structures for Visual Contexts. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, 6619–6628. Computer Vision Foundation / IEEE.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention is All you Need. In Guyon, I.; von Luxburg, U.; Bengio, S.; Wallach, H. M.; Fergus, R.; Vishwanathan, S. V. N.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, 5998–6008.

Xu, D.; Zhu, Y.; Choy, C. B.; and Fei-Fei, L. 2017. Scene Graph Generation by Iterative Message Passing. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, 3097–3106. IEEE Computer Society.

Yan, S.; Shen, C.; Jin, Z.; Huang, J.; Jiang, R.; Chen, Y.; and Hua, X. 2020. PCPL: Predicate-Correlation Perception Learning for Unbiased Scene Graph Generation. *CoRR*, abs/2009.00893.

Yang, J.; Lu, J.; Lee, S.; Batra, D.; and Parikh, D. 2018. Graph R-CNN for Scene Graph Generation. In *Proceedings of the European Conference on Computer Vision (ECCV)*.

Yu, R.; Li, A.; Morariu, V. I.; and Davis, L. S. 2017. Visual Relationship Detection with Internal and External Linguistic Knowledge Distillation. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, 1068–1076. IEEE Computer Society.

Zellers, R.; Yatskar, M.; Thomson, S.; and Choi, Y. 2018. Neural Motifs: Scene Graph Parsing With Global Context. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, 5831–5840. IEEE Computer Society.

Zhan, Y.; Yu, J.; Yu, T.; and Tao, D. 2019. On exploring undetermined relationships for visual relationship detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5128–5137.

Zhan, Y.; Yu, J.; Yu, T.; and Tao, D. 2020. Multi-task compositional network for visual relationship detection. *International Journal of Computer Vision*, 128(8): 2146–2165.

Zhong, Y.; Wang, L.; Chen, J.; Yu, D.; and Li, Y. 2020. Comprehensive Image Captioning via Scene Graph Decomposition. In Vedaldi, A.; Bischof, H.; Brox, T.; and Frahm, J.-M., eds., *Computer Vision – ECCV 2020*, 211–229. Cham: Springer International Publishing. ISBN 978-3-030-58568-6.