CogTree: Cognition Tree Loss for Unbiased Scene Graph Generation

Jing Yu1∗†, Yuan Chai2∗, Yujing Wang3, Yue Hu1 and Qi Wu 4

1Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
2Intelligent Computing and Machine Learning Lab, School of ASEE, Beihang University, Beijing, China
3Key Laboratory of Machine Perception, MOE, School of EECS, Peking University, Beijing, China
4University of Adelaide, Australia

{yujing02, huyue}@iie.ac.cn, chaiyuan@buaa.edu.cn, yujwang@pku.edu.cn, qi.wu01@adelaide.edu.au

Abstract

Scene graphs are semantic abstraction of images that encourage visual understanding and reasoning. However, the performance of Scene Graph Generation (SGG) is unsatisfactory when faced with biased data in real-world scenarios. Conventional debiasing research mainly studies from the view of balancing data distribution or learning unbiased models and representations, ignoring the correlations among the biased classes. In this work, we analyze this problem from a novel cognition perspective: automatically building a hierarchical cognitive structure from the biased predictions and navigating that hierarchy to locate the relationships, making the tail relationships receive more attention in a coarse-to-fine mode. To this end, we propose a novel debiasing Cognition Tree (CogTree) loss for unbiased SGG. We first build a cognitive structure CogTree to organize the relationships based on the prediction of a biased SGG model. The CogTree distinguishes remarkably different relationships at first and then focuses on a small portion of easily confused ones. Then, we propose a debiasing loss specially for this cognitive structure, which supports coarse-to-fine distinction for the correct relationships. The loss is model-agnostic and consistently boosting the performance of several state-of-the-art models. The code is available at: https://github.com/CYVincent/Scene-Graph-Transformer-CogTree.

1 Introduction

Making abstraction from an image into high-level semantics is one of the most remarkable capabilities of humans. Scene Graph Generation (SGG) [Xu et al., 2017] — a task of extracting objects and their relationships in an image to form a graphical representation — aims to achieve the abstraction capability and bridge the gap between vision and language. SGG has greatly benefited the down-stream tasks of question answering [Norcliffe-Brown et al., 2018; Zhu et al., 2020] and visual understanding [Shi et al., 2019; Jiang et al., 2020]. Some works [Zhu et al., 2020; Jiang et al., 2020] feed the scene graphs into graph neural networks for relation-aware object representation. Some others [Hudson and Manning, 2019] perform sequential reasoning by traversing the relational graphs. Compared with independent objects, the rich relationships benefit explainable reasoning.

Although much effort has been made with high accuracy in object detection, the detected relationships are far from satisfaction due to the long-tailed data distribution. Only a small portion of the relationships have abundant samples (head) while most ones contain just a few (tail). This heavily biased training data causes biased relationship prediction. Tail relationships will be mostly predicted into head classes, which are not discriminative enough for high-level reasoning, such as falsely predicting on instead of looking at, and coarsely predicting on instead of walking on. Besides, tail relationships like standing on, sitting on and lying on can be even harder to distinguish from each other due to their visual similarity and scarce training data.

To tackle this problem, most research focuses on learning
unbiased models by re-weighting losses [Zareian et al., 2020] or disentangling unbiased representations [Tang et al., 2020]. However, humans can effectively infer the correct relationships even when some relationships appear more frequently than others. The essential difference between human and AI systems that has been ignored lies neither in the learning strategy nor feature representation, but in the way that the concepts are organized. To illustrate this discrepancy, we show an example: truck parked on street in Figure 1. Existing models treat all the relationships independently for flat classification (Figure 1(a)). Because of the data bias, the head relationships, e.g. on and near, obtain high predicted probability. In contrast, the cognition theory [Sarafyazd and Jazayeri, 2019] supports that humans process information hierarchically in the prefrontal cortex (PFC). We intuitively start from making a rough distinction between remarkably different relationships. As shown in Figure 1(b), relationships belonging to the concept “on”, e.g. on and parked on, will firstly be distinguished from the ones about “near”, e.g. near and behind; then we move to distinguish the slight discrepancy among easily confused ones in one concept, e.g. parked on, standing on and walking on.

Inspired by the hierarchical reasoning mechanism in PFC, we propose a novel loss function, Cognition Tree (CogTree) loss, for unbiased scene graph generation. We first propose to build a hierarchy of the relationships, imitating the knowledge structure built from the independent relationships in human mind. The CogTree is derived from the prediction of a biased SGG model that satisfies the aforementioned thinking principles: distinguishing remarkably different relationships at first and then focusing on a small portion of easily confused ones. Then we design a CogTree-based loss to train the SGG network from scratch. This loss enables the network to surpass the noises from inter-concept relationships and then intra-concept relationships progressively. It frees the SGG models from the burden of distinguishing detailed discrepancy among all the relationships at one time, resulting in more accurate prediction via this coarse-to-fine strategy.

The main contributions are summarized as follows: (1) We propose a debiasing loss based on the inductive hierarchical structure, which supports coarse-to-fine distinction for the correct relationships while progressively eliminating the interference of irrelevant ones. (2) We automatically build the cognitive structure of the relationships from the biased SGG predictions, which reveals the inherent hierarchy that the relationships are organized after biased training. (3) The proposed loss is model-agnostic and can be applied to several representative SGG models. The results of extensive experiments indicate that the CogTree loss remarkably alleviates the bias in SGG and our loss achieves a new state-of-the-art performance compared with existing debiasing methods.

2 Related Work

Scene Graph Generation. SGG [Xu et al., 2017] products graphical abstraction of an image and encourages visual relational reasoning and understanding in various down-stream tasks [Shi et al., 2019; Zhu et al., 2020]. Early works focus on object detection and relationship detection via independent networks [Lu et al., 2016; Zhang et al., 2017], ignoring the rich contextual information. To incorporate the global visual context, recent works leverage message passing mechanism [Xu et al., 2017; Chen et al., 2019a], recurrent architectural strategies [Zellers et al., 2018; Tang et al., 2019] and contrastive learning [Zhang et al., 2019] for more discriminative object and relationship representations. Although the accuracy is high in object detection, the relationship detection is far from satisfaction due to the heavily biased data. [Chen et al., 2019b; Tang et al., 2019] consider the biased SGG problem and propose mean Recall as the unbiased metric without corresponding debiasing solutions. The recent work [Liang et al., 2019] prunes the predominant relationships and keeps the tail but informative ones in the dataset. [Tang et al., 2020] proposes the first solution for unbiased SGG by counterfactual surgeries on causal graphs. We rethink SGG task from the cognition view and novelly solve the debiasing problem based on the coarse-to-fine structure of the relationships.

Biased Classification. Classification on highly-biased training data has been extensively studied in previous work, which can be divided into three categories: (1) balancing data distribution by data augmentation or re-sampling [Burnaev et al., 2015; Li and Vasconcelos, 2019]; (2) debiasing learning strategies by re-weighting losses or training curriculums [Lin et al., 2017b; Cui et al., 2019]; (3) separating biased representations from the unbiased for prediction [Cadene et al., 2019; Tang et al., 2020]. Our CogTree loss belongs to the second category but differs from existing methods in that we are the first to leverage the hierarchical structure inherent in the relationships for re-weighting, which enables more discriminative representation learning by a coarse-to-fine mode.

3 Methodology

We first introduce the automatic CogTree building process. Then a novel debiasing loss based on the tree structure is proposed to train the SGG models. Since the CogTree loss is model-agnostic and applicable to various SGG models, we case-study two representitive models, i.e. the widely compared MOTIFS [Zellers et al., 2018] and the state-of-the-art VCTree [Tang et al., 2019], and a newly proposed model, which first applies the transformer [Vaswani et al., 2017] architecture to the SGG scenario and achieves superior performance. We name it as SG-Transformer and take it as a strong baseline. The framework is illustrated in Figure 2.

3.1 Automatic CogTree Building

Once the SGG model has been trained on the long-tailed data, the model tends to be biased on predicting “head” relationships. When different samples are predicted with the same relationship, they mostly share similar properties on either visual appearance (e.g. walking on and standing on) or high-level semantics (e.g. has and with). We regard these relationships sharing common properties as in one concept. We automatically induce concepts from the biased predictions and re-organize the relationships based on the concepts by a tree structure, denoted as Cognition Tree (CogTree). As shown on the top of Figure 2, this automatic process contains the following three steps:
**Step 1: Bias-Based Concept Induction.** For all the samples in the ground-truth class $d_i$, we predict their labels via a biased model and calculate the distribution of predicted label frequency, denoted as $p_i$. The most frequently predicted class $d_i$ is regarded as the concept relationship of the ground-truth class $d_i$. As shown in Figure 2(Step 1), on is the concept relationship of itself, standing on and walking on. The above operation induces all the relationships into $C$ concepts with corresponding concept relationships $\{c_i\}_i$.

**Step 2: Concept-Centered Subtree Building.** We reorganize all the relationships in each concept by a Concept-Centered Subtree. For the $i^{th}$ subtree, the root is $c_i$ while the leaves are the fine-grained relationships induced in $c_i$. Note that, if a subtree only contains a root $c_i$, e.g., parked on in Figure 2 (Step 2(A)), then $c_i$ is considered not representative enough to summarize common properties. We further induce it to the most approximate concept relationship, which has the second highest frequency in $p_i$, e.g., induce parked on to on in Figure 2 (Step 2(B)). This process outputs $T$ subtrees.

**Step 3: Cognition-Based Subtree Aggregation.** We aggregate the $T$ subtrees into one hierarchical CogTree. As illustrated in Figure 3, CogTree contains four functional layers and each layer induces relationships into coarser groups than the layer that comes after it. Specifically, the **Root Layer** $(y_0)$ contains a virtual node. The **Concept Layer** $(y_1)$ distinguishes which concept the input belongs to. It contains $T$ virtual nodes, each representing a subtree induced in Step 2. The following **Coarse-fine Layer** $(y_2)$ tells whether the input can be described by a coarse-grained or fine-grained relationship in a certain concept. To this end, $y_2$ splits each node in $y_1$ into two nodes: one leaf node indicating the concept relationship while the other virtual node representing the cluster of fine-grained relationships in that concept. Each virtual node in $y_2$ links to its fine-grained relationships of the corresponding concept in the **Fine-grained Layer** $(y_3)$. $y_3$ focuses on distinguishing the slight discrepancy among easily confused relationships, e.g., standing on and walking on.

### 3.2 Debiasing CogTree Loss

We propose a CogTree loss for training the SGG model for unbiased predictions. This loss encourages the model to classify relationships from coarse to fine according to the CogTree structure (see the bottom right of Figure 2). To better make the loss functions (3) and (4) understood, several notations are provided in advance as below:

- **Ground-Truth CogTree Path (GCP).** It denotes the ground-truth classification path in CogTree for a training sample. Given the ground-truth label of a sample, we track the path from the root to the leaf with that label in CogTree and denote this path as the ground-truth path $L_{path} = \{l_0, l_1, \ldots, l_M\}$, where the $k^{th}$ value $l_m$ is the ground-truth node at layer $y_m$.

- **Predicted CogTree Probability (PCP).** It means the predicted probability of each node in CogTree for a given sample by a biased SGG model. We denote the predicted probability set of all the classes as $P_{pred}$ and denote the probability of class $k$ as $p_k$. The PCP of node $i$ is defined as below:

$$z_i = \begin{cases} p_k \frac{1}{|C(i)|} \sum_{j \in C(i)} z_j & \text{if } i \text{ is a leaf and } class(i) = k \\ \frac{1}{|C(i)|} \sum_{j \in C(i)} w_j & \text{if } i \text{ is not a leaf} \end{cases}$$

(1)

where $C(i)$ means the children of node $i$. $class(i)$ denotes the corresponding class of node $i$.

- **Class-Balanced Weight (CBW).** It denotes the balance weight of each node in CogTree. Since the success of re-weighting strategy for debiasing, we adopt a well performed weighting factor [Cui et al., 2019] to compute CBW of each leaf and uniformly define CBW of each node as below:

$$w_i = \begin{cases} (1 - \beta)/(1 - \beta^{n_i}) & \text{if } i \text{ is a leaf} \\ \frac{1}{|C(i)|} \sum_{j \in C(i)} w_j & \text{if } i \text{ is not a leaf} \end{cases}$$

(2)

where $\beta$ is a hyper-parameter in the range of $[0, 1)$ and $n_i$ is the sample number of the corresponding class of leaf $i$.

**CogTree Loss.** It contains two parts for complementary
benefits: CogTree-based class-balanced (TCB) loss for debiasing by coarse-to-fine relationship classification and class-balanced (CB) loss for debiasing by re-weighting on softmax cross-entropy loss. The two loss terms are defined below.

Given a sample with the ground-truth path $l_{\text{path}}$, as illustrated in Figure 3, we compute the class-balanced softmax cross-entropy loss on each layer of CogTree, and average the results to obtain the TCB loss as below:

$$\mathcal{L}_{\text{TCB}} = \frac{1}{|l_{\text{path}}|} \sum_{i \in l_{\text{path}}} - w_i \log(\frac{\exp(z_i)}{\sum_{j \in B(i)} \exp(z_j)})$$

where $B(i)$ means the brothers of node $i$. TCB loss forces the network to surpass the noises from inter-concept relationships and learn concept-specific embeddings first, and then surpass the noises within one concept to refine relationship-specific embeddings, resulting in more fine-grained predictions.

Given a sample with the ground-truth class $k$, CB loss computes the class-balanced softmax cross-entropy based on the biased predicted probabilities $P_{\text{pred}}$:

$$\mathcal{L}_{\text{CB}} = - w_k \log(\frac{\exp(p_k)}{\sum_{j \in P_{\text{pred}}} \exp(p_j)})$$

where $w_k$ is the weighting factor [Cui et al., 2019] for class $k$. Empirically, we found it the best to combine TCB with CB. In conclusion, our CogTree loss is:

$$\mathcal{L} = \mathcal{L}_{\text{CB}} + \lambda \mathcal{L}_{\text{TCB}}$$

where $\lambda = 1$ is the balancing weight. The SGG model is trained from scratch by the CogTree loss and predicting results via flat multi-class classifier without modification.

### 3.3 Scene Graph Generation Models

We case-study on three SGG models, i.e. MOTIFS, VCTree and our SG-Transformer. As shown in the bottom left of Figure 2, their architectures mainly contains two processes: object classification (bottom) for detecting the object positions and labels and relationship classification (top) for predicting the relationships between objects. We first review MOTIFS, and then introduce VCTree and SG-Transformer in a comparable fashion. All models are trained by the conventional cross-entropy loss [Zellers et al., 2018].

**MOTIFS.** A pre-trained Faster R-CNN [Ren et al., 2015] is applied to extract top $K$ objects $O = \{o_i\}_K$ and describe object $o_i$ by the RoIAlign visual feature $v_i$, the tentative object label $l_i$, and the spatial feature $b_i$. The above three kinds of features are concatenated and fed into $Encoder_o$ to obtain the refined object embedding $m_i$. $m_i$ is then decoded by $Decoder_o$ to predict its final object label $l^*_o$. MOTIFS implemented $Encoder_o$ as a Bi-LSTM and adopts an LSTM as $Decoder_o$. The relationship encoder $Encoder_r$ first adopts another Bi-LSTM to capture the contextual information and then fuses the pairwise object features with the union region feature for decoding. The relationship decoder $Decoder_r$ is a fully connected layer followed by a softmax layer.

**VCTree.** Different from MOTIFS, VCTree utilizes two independent TreeLSTMs as $Encoder_o$ and $Encoder_r$, respectively. It additionally takes the bounding box pair feature as another input of $Decoder_r$.

**SG-Transformer.** It adopts the transformer architecture as $Encoder_o$ and $Encoder_r$ by adjusting inputs for the SGG scenario. As shown in Figure 4, $Encoder_o$ contains $N$ object-to-object (O2O) transformer blocks with the input of object embeddings $\{v_i\}_K$. $Encoder_r$ consists of $M$ relation-to-object (R2O) transformer blocks with the input of relationship embeddings $\{r_{ij}\}_{K \times K}$ and the object embeddings $\{m_i\}_K$. $r_{ij}$ is a combination of three kinds of features from objects $o_i$ and $o_j$: spatial features, union region feature, word embeddings [Pennington et al., 2014] of the predicted object labels. The output from $Encoder_r$ is concatenated with $m_i$ and $m_j$ for decoding. $Decoder_o$ and $Decoder_r$ contain a fully connected layer followed by a softmax layer.

### 4 Experiments

**Dataset:** We evaluate our models on the widely compared Visual Genome (VG) split [Xu et al., 2017], with the 150 most frequent object classes and 50 most frequent relationship classes in VG [Krishna et al., 2017]. The VG split only contains training set and test set and we follow [Zellers et al., 2018] to sample a 5K validation set from the training set.

**Tasks and Evaluation:** The SGG task contains three sub-tasks [Zellers et al., 2018]: (1) Predicate Classification (PredCls) takes the ground-truth object labels and bounding boxes for relationship prediction; (2) Scene Graph Classification (SGCls) takes ground-truth bounding boxes for object and relationship prediction; (3) Scene Graph Detection (SGDet) predict SGGs from scratch. The conventional metric Recall@K (R@K) is reported but not considered as the main evaluation due to its bias. We adopt the recently proposed balanced metric mean Recall@K (mR@K) [Chen et al., 2019b; Tang et al., 2019]. mR@K calculates R@K for each class independently, and then average the results.
We evaluate the CogTree loss on three models: MOTIFS, VCTree and SG-Transformer, and compare the performance.

We investigate each loss terms, the weighting method, two tree building strategies and two methods for computing PCP. Our re-implemented SGG models are denoted by the superscript.

Implementation: The object detector is the pre-trained Faster R-CNN [Ren et al., 2015] with ResNeXt-101-FPN [Lin et al., 2017a]. $\lambda$ is set to 1 and $\beta$ is set to 0.999. SG-Transformer contains 3 O2O blocks and 2 R2O blocks with 12 attention heads. Models are trained by SGD optimizer with 5 epochs. The mini-batch size is 12 and the learning rate is $1.2 \times 10^{-3}$. Experiments are implemented with PyTorch and conducted with NVIDIA Tesla V100 GPUs.

4.1 State-of-the-Art Comparison

We evaluate the CogTree loss on three models: MOTIFS, VCTree and SG-Transformer, and compare the performance with the state-of-the-art debiasing approach TDE [Tang et al., 2020]. We also compare with the biased models, including IMP+ [Xu et al., 2017], FREQ [Zellers et al., 2018], KERN [Chen et al., 2019b]. The results are shown in Table 1. In the view of mR@K, CogTree remarkably improves all the baselines on all the metrics. The strongest baseline VCTree also achieves 3.6%~13.6% boost. CogTree consistently outperforms conventional debiasing methods: focal loss, reward and resample, and the SOTA debiasing method TDE. Wherein, mR@20 achieves superior improvement compared with mR@50, which indicates that more accurate predictions have higher rank by CogTree. It is worth highlighting that CogTree outperforms TDE on both mR@K and most metrics of R@K on the strongest baseline VCTree.

Analysis of Labeling Bias. We observe a drop of R@K with the increase of mR@K from baselines to CogTree. Further analysis in Figure 5 reveals that annotators prefer simple and vague labels due to bounded rationality [Simon, 1990]; (2) CogTree predicts more fine-grained relationships. Our analysis is consistent with the recent study [Tang et al., 2020].

Figure 5: The pie chart shows the relationships that are correctly predicted by the baseline but regarded as incorrect by CogTree. The predictions by CogTree are shown under the pie chart. The increase of mR@K bringing a drop of R@K is commonly existed (see Table 1). Combining with the qualitative analysis, we believe that the contradiction of R@K and mR@K is caused by two reasons: (1) annotators prefer simple and vague labels due to bounded rationality [Simon, 1990]; (2) CogTree predicts more fine-grained relationships. Our analysis is consistent with the recent study [Tang et al., 2020].

Table 1: State-of-the-art comparison on R@K and mR@K. Our re-implemented SGG models are denoted by the superscript *.

Table 2: R@K of the tail 45 classes on predicate classification.

Table 3: Ablation study of key components in CogTree loss.
Table 4: Performance with different balancing weights.

| λ   | Predicate Classification | Scene Graph Classification | Scene Graph Detection |
|-----|--------------------------|----------------------------|-----------------------|
| 0.4 | 21.17 / 27.20 / 29.59   | 12.09 / 15.03 / 16.12     | 7.22 / 9.72 / 11.30  |
| 0.7 | 21.28 / 26.39 / 28.69   | 12.22 / 14.91 / 15.88     | 8.62 / 11.30 / 12.70 |
| 1   | 22.89 / 28.38 / 30.97   | 12.96 / 15.68 / 16.72     | 7.92 / 11.05 / 12.70 |
| 1.3 | 22.06 / 27.11 / 29.21   | 12.35 / 15.04 / 15.87     | 7.84 / 10.48 / 12.19 |
| 1.6 | 21.34 / 26.80 / 29.18   | 12.20 / 14.70 / 15.82     | 6.78 / 9.04 / 10.22  |

4.4 Parameter Analysis

In Table 4, we assess the effect of λ in Eq. 5. λ = 1 achieves the best performance on most metrics. The performance drops slightly in the range of [0.7, 1.3]. We also vary the number of O2O and R2O blocks from 2 to 4 in SG-Transformer and obtain the highest mR@K with 3 O2O blocks and 2 R2O blocks. We use the above settings in our full model.

5 Conclusion

In this paper, we propose a CogTree loss to generate unbiased scene graphs with highly biased data. We novelty leverage the biased prediction from SGG models to organize the independent relationships by a tree structure, which contains multiple layers corresponding to the relationships from coarse to fine. We propose a CogTree loss specially for the above tree structure that supports hierarchical distinction for the correct relationships while progressively eliminating the irrelevant ones. The loss is model-agnostic and consistently boosting the performance of various SGG models with remarkable improvement. How to incorporate commonsense knowledge to optimize the CogTree structure will be our future work.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (Grant No. 62006222).
References

[Burnaev et al., 2015] Evgeny Burnaev, Pavel Erofeev, and Artem Papanov. Influence of resampling on accuracy of imbalanced classification. In ICMV, pages 987521–987521, 2015.

[Cadene et al., 2019] Remi Cadene, Corentin Dancette, Hedi Ben younes, Matthieu Cord, and Devi Parikh. Rubi: reducing unimodal biases for visual question answering. In NeurIPS, pages 841–852, 2019.

[Chen et al., 2019a] Long Chen, Hanwang Zhang, Jun Xiao, Xiangnan He, Shiliang Pu, and Shih-Fu Chang. Counterfactual critic multi-agent training for scene graph generation. In ICCV, pages 4613–4623, 2019.

[Chen et al., 2019b] Tianshi Chen, Weihao Yu, Riquan Chen, and Liang Lin. Knowledge-embedded routing network for scene graph generation. In CVPR, pages 6163–6171, 2019.

[Cui et al., 2019] Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. Class-balanced loss based on effective number of samples. In CVPR, pages 9268–9277, 2019.

[Hudson and Manning, 2019] Drew Hudson and Christopher D Manning. Learning by abstraction: the neural state machine. In NeurIPS, pages 5903–5916, 2019.

[Jiang et al., 2020] Xiaoze Jiang, Jing Yu, Zengchang Qiu, Yingying Zhuang, Xingxing Zhang, Yue Hu, and Qi Wu. Dualvd: an adaptive dual encoding model for deep visual understanding in visual dialogue. In AAAI, pages 11125–11132, 2020.

[Krishna et al., 2017] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, Michael S. Bernstein, and Li Fei-Fei. Visual genome: connecting language and vision using crowd-sourced dense image annotations. IJCV, 123:32–73, 2017.

[Li and Vasconcelos, 2019] Yi Li and Nuno Vasconcelos. Repair: removing representation bias from dataset resampling. In CVPR, pages 9572–9581, 2019.

[Liang et al., 2019] Yuanzhi Liang, Yalong Bai, Wei Zhang, Xueming Qian, Li Zhu, and Tao Mei. Vrr-vg: refocusing visually-relevant relationships. In ICCV, pages 10403–10412, 2019.

[Lin et al., 2017a] Tsung-Yi Lin, Piotr Dollar, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In CVPR, pages 2117–2125, July 2017.

[Lin et al., 2017b] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollar. Focal loss for dense object detection. In ICCV, pages 2980–2988, 2017.

[Lu et al., 2016] Cewu Lu, Ranjay Krishna, Michael Bernstein, and Li Fei-Fei. Visual relationship detection with language priors. In ECCV, pages 852–869, 2016.

[Norcliffe-Brown et al., 2018] Will Norcliffe-Brown, Stathis Vafeias, and Sarah Parisot. Learning conditioned graph structures for interpretable visual question answering. In NeurIPS, pages 8334–8343, 2018.

[Pennington et al., 2014] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In EMNLP, pages 1532–1543, 2014.

[Ren et al., 2015] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: towards real-time object detection with region proposal networks. In NeurIPS, pages 91–99, 2015.

[Sarafyazd and Jazayeri, 2019] Morteza Sarafyazd and Mehrdad Jazayeri. Hierarchical reasoning by neural circuits in the frontal cortex. Science, 364(6441):eaav8911, 2019.

[Shi et al., 2019] Jiaxin Shi, Hanwang Zhang, and Juani Li. Explainable and explicit visual reasoning over scene graphs. In CVPR, pages 8376–8384, 2019.

[Simon, 1990] Herbert A Simon. Bounded rationality. In Utility and probability, pages 15–18. Springer, 1990.

[Tang et al., 2019] Kiahua Tang, Hanwang Zhang, Baoyuan Wu, Wenhan Luo, and Wei Liu. Learning to compose dynamic tree structures for visual contexts. In CVPR, pages 6619–6628, 2019.

[Tang et al., 2020] Kiahua Tang, Yulei Niu, Jianqiang Huang, Jiaxin Shi, and Hanwang Zhang. Unbiased scene graph generation from biased training. In CVPR, pages 3716–3725, 2020.

[Vaswani et al., 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Liion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In NeurIPS, pages 5998–6008, 2017.

[Xu et al., 2017] Danfei Xu, Yuke Zhu, Christopher B Choy, and Li Fei-Fei. Scene graph generation by iterative message passing. In CVPR, pages 5410–5419, 2017.

[Zareian et al., 2020] Alireza Zareian, Svebor Karaman, and Shih-Fu Chang. Bridging knowledge graphs to generate scene graphs. In ECCV, pages 606–623, 2020.

[Zellers et al., 2018] Rowan Zellers, Mark Yatskar, Sam Thomson, and Yejin Choi. Neural motifs: scene graph parsing with global context. In CVPR, pages 5831–5840, 2018.

[Zhang et al., 2017] Ji Zhang, Mohamed Elhoseiny, Scott Cohen, Walter Chang, and Ahmed Elgammal. Relationship proposal networks. In CVPR, pages 5678–5686, 2017.

[Zhang et al., 2019] Ji Zhang, Kevin J. Shih, Ahmed Elgammal, Andrew Tao, and Bryan Catanzaro. Graphical contrastive losses for scene graph parsing. In CVPR, pages 11535–11543, 2019.

[Zhu et al., 2020] Zhao Zhu, Jing Yu, Yujing Wang, Yajing Sun, Yue Hu, and Qi Wu. Mucko: multi-layer cross-modal knowledge reasoning for fact-based visual question answering. In IJCAI, pages 1097–1103, 2020.