Devil's on the Edges: Selective Quad Attention for Scene Graph Generation

Deunsol Jung   Sanghyun Kim   Won Hwa Kim   Minsu Cho

Pohang University of Science and Technology (POSTECH), South Korea

http://cvlab.postech.ac.kr/research/SQUAT

Abstract

Scene graph generation aims to construct a semantic graph structure from an image such that its nodes and edges respectively represent objects and their relationships. One of the major challenges for the task lies in the presence of distracting objects and relationships in images; contextual reasoning is strongly distracted by irrelevant objects or backgrounds and, more importantly, a vast number of irrelevant candidate relations. To tackle the issue, we propose the Selective Quad Attention Network (SQUAT) that learns to select relevant object pairs and disambiguate them via diverse contextual interactions. SQUAT consists of two main components: edge selection and quad attention. The edge selection module selects relevant object pairs, i.e., edges in the scene graph, which helps contextual reasoning, and the quad attention module then updates the edge features using both edge-to-node and edge-to-edge cross-attentions to capture contextual information between objects and object pairs. Experiments demonstrate the strong performance and robustness of SQUAT, achieving the state of the art on the Visual Genome and Open Images v6 benchmarks.

1. Introduction

The task of scene graph generation (SGG) is to construct a visually-grounded graph from an image such that its nodes and edges respectively represent objects and their relationships in the image [27, 43, 46]. The scene graph provides a semantic structure of images beyond individual objects and thus is useful for a wide range of vision problems such as visual question answering [36, 37], image captioning [53], image retrieval [11], and conditional image generation [10], where a holistic understanding of the relationships among objects is required for high-level reasoning. With recent advances in deep neural networks for visual recognition, SGG has been actively investigated in the computer vision community. A vast majority of existing methods tackle SGG by first detecting candidate objects and then performing contextual reasoning between the objects via message pass-
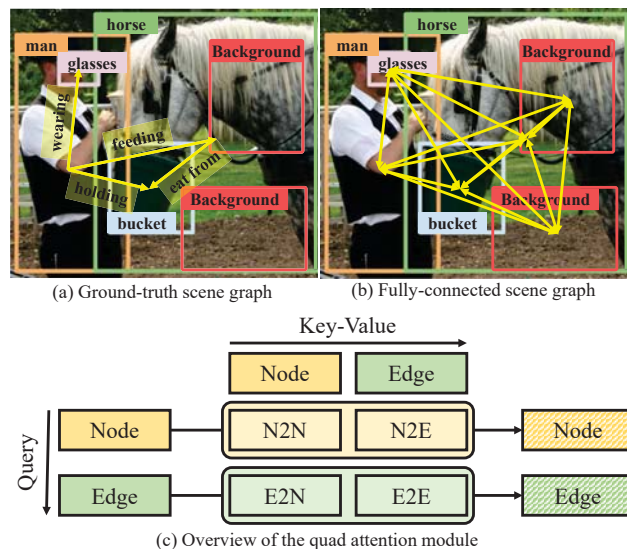

Figure 1.   (a) The ground-truth scene graph contains only 4 ground-truth objects and 4 relations between the objects. (b) Only 13% of edges in a fully-connected graph have the actual relationships according to the ground-truths. (c) The overview of the quad attention. The node features are updated by node-to-node (N2N) and node-to-edge (N2E) attentions, and the edge features are updated by edge-to-node (E2N) and edge-to-edge (E2E) attentions.

ing [19, 21, 43] or sequential modeling [28, 36, 50]. Despite these efforts, the task of SGG remains extremely challenging, and even the state-of-the-art methods do not produce reliable results for practical usage.

While there exist a multitude of challenges for SGG, the intrinsic difficulty may lie in the presence of distracting objects and relationships in images; there is a vast number of potential but irrelevant relations, *i.e.*, edges, which quadratically increase with the number of candidate objects, *i.e.*, nodes, in the scene graph. The contextual reasoning for SGG in the wild is thus largely distracted by irrelevant objects and their relationship pairs. Let us take a simple example as in Fig. 1, where 4 objects and 4 relations in its ground-truth scene graph exist in the given image. If our

object detector obtains 6 candidate boxes, 2 of which are from the background (red), then the contextual reasoning, *e.g.*, message passing or self-attention, needs to consider 30 potential relations, 87% of which are not directly related according to the ground-truth and most of them may thus act as distracting outliers. In practice, the situation is far worse; in the Visual Genome dataset, the standard benchmark for SGG, an image contains 38 objects and 22 relationships on average [43], which means that only around 1% of object pairs have direct and meaningful relations even when object detection is perfect. As will be discussed in our experiments, we find that existing contextual reasoning schemes obtain only a marginal gain at best and often degrade the performance. The crux of the matter for SGG may lie in developing a robust model for contextual reasoning against irrelevant objects and relations.

To tackle the issue, we propose the *Selective Quad Attention Network (SQUAT)* that learns to select relevant object pairs and disambiguate them via diverse contextual interactions with objects and object pairs. The proposed method consists of two main components: edge selection and quad attention. The edge selection module removes irrelevant object pairs, which may distract contextual reasoning, by predicting the relevance score for each pair. The quad attention module then updates the edge features using edge-to-node and edge-to-edge cross-attentions as well as the node features using node-to-node and node-to-edge cross-attentions; it thus captures contextual information between all objects and object pairs, as shown in Figure 1 (c). Compared to previous methods [19, 21], which perform either node-to-node or node-to-edge interactions, our quad attention provides more effective contextual reasoning by capturing diverse interactions in the scene graph. For example, in the case of Fig. 1 (a), ['man', 'feeding', 'horse'] relates to ['man', 'holding', 'bracket'] and ['horse', 'eat from', 'bracket'], and vice versa; node-to-node or node-to-edge interactions are limited in capturing such relations between the edges.

Our contributions can be summarized as follows:

- We introduce the edge selection module for SGG that learns to select relevant edges for contextual reasoning.

- We propose the quad attention module for SGG that performs effective contextual reasoning by updating node and edge features via diverse interactions.

- The proposed SGG model, SQUAT, outperforms the state-of-the-art methods on both Visual Genome and Open Images v6 benchmarks. In particular, SQUAT achieves remarkable improvement on the SGDet settings, which is the most realistic and challenging.

2. Related work

Scene graph generation The vast majority of SGG methods [19, 21, 50] predict scene graphs in two stages: object detection and contextual reasoning. While the first stage is typically done by a pre-trained detection module [1, 31], contextual reasoning is performed by different types of message passing [2, 5, 12, 19–21, 23, 30, 41–43, 48, 49], which uses a graph neural network with node-to-edge and edge-to-node attentions, or sequential modeling [8, 28, 36, 50], which updates the node features with node-to-node attention and constructs edge features with edge-to-node attention. Unlike the previous methods, we propose quad attention, which comprises node-to-node, node-to-edge, edge-to-node, and edge-to-edge interactions, to capture all types of context exchange between candidate objects and their pairs for relational reasoning. In contextual reasoning, most of the methods consider all the candidate object pairs, *i.e.*, a fully-connected graph whose nodes are candidate objects. While Graph R-CNN [46] proposes a relation proposal network that prunes the edges from a fully-connected graph, it focuses on reducing the cost of message passing and does not analyze the effect of edge selection on the performance of scene graph generation. In contrast, we introduce an effective edge selection method and provide an in-depth analysis of it. On the other hand, since dataset imbalance/bias has recently emerged as a critical bottleneck for learning SGG[1], several methods [3, 16, 34, 40, 47] propose to adopt the techniques from long-tailed recognition, *e.g.*, data resampling [6, 19] and loss reweighting [9, 15, 45].

Transformers for vision tasks and graph structures Transformers [38] have been adapted to the various computer vision tasks, *e.g.*, object classification [7, 26], object detection [1, 26, 32, 54] and segmentation [26, 52], and also extended for graph structures [13, 22, 29, 33]. Despite their success, vision transformer networks typically suffer from high complexity and memory consumption. Several variants of transformer networks [4, 14, 32, 39, 54] have been proposed to tackle the issue and showed that a proper sparsification technique, *e.g.*, Sparse DETR [32], can not only reduce the cost of computation and memory but also improve the task performance. Our transformer network is designed to perform contextual reasoning for scene graph generation by capturing the inherent relationships between objects and relevant object pairs, and unlike existing sparsification methods, which focus on token pruning [32] or local attention [14, 54], our edge selection module prunes not only the query edges to update but also the key-value edges used for updating.

---

[1]For example, in the Visual Genome dataset, the most frequent entity class is 35 times larger than the least frequent one, and the most frequent predicate class is 8,000 times larger than the least frequent one.
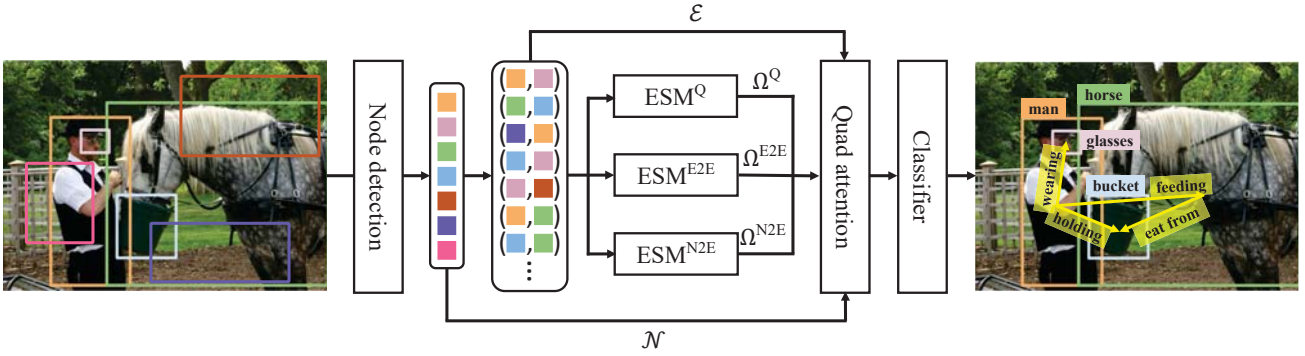
Figure 2. The overall architecture of Selective Quad Attention Networks (SQUAT). SQUAT consists of three components: the node detection module, the edge selection module, and the quad attention module. First, the node detection module extracts nodes $\mathcal{N}$ by detecting object candidate boxes and extracting their features. Also, all possible pairs of the nodes are constructed as initial edges $\mathcal{E}$. Second, the edge selection module select valid edges $(\Omega^{\text{Q}}, \Omega^{\text{E2E}}, \Omega^{\text{N2E}})$ with high relatedness scores. Third, the quad attention module updates the node and edge features via four types of attention. Finally, the output features are passed into a classifier to predict the scene graph. See Sec. 4 for the details.

3. Problem Definition

Given an image I, the goal of SGG is to generate a visually grounded graph $G = (\mathcal{O}, \mathcal{R})$ that represents objects $\mathcal{O}$ and their semantic relationships $\mathcal{R}$ for object classes $\mathcal{C}$ and predicate classes $\mathcal{P}$. An object $o_i \in \mathcal{O}$ is described by a pair of a bounding box $b_i \in [0,1]^4$ and its class label $c_i \in \mathcal{C}$: $o_i = (b_i, c_i)$. A relationship $r_k \in \mathcal{R}$ is represented by a triplet of a subject $o_i \in \mathcal{O}$, an object $o_j \in \mathcal{O}$, and a predicate label $p_{ij} \in \mathcal{P}$: $r_k = (o_i, o_j, p_{ij})$, which represents relationship p_{ij} between subject o_i and object o_j.

4. Selective Quad Attention Networks

To generate semantically meaningful scene graphs as described in Section 3, we propose the Selective Quad Attention Network (SQUAT) that consists of three main components as shown in Fig. 2: the node detection module (Sec. 4.1), the edge selection module (Sec. 4.2), and the quad attention module (Sec. 4.3). First, the node detection module establishes nodes for a scene graph by detecting object candidate boxes and extracting their features. All possible pairs of the nodes are constructed as potential edges. Second, among all the potential edges, the edge selection module selects valid edges with high relatedness scores. Third, the quad attention module updates the features of nodes and valid edges via four types of attention: node-to-node (N2N), node-to-edge (N2E), edge-to-node (E2N), and edge-to-edge (E2E). For the quad attention module, we use three edge selection modules: query edge selection module for entire quad attention (ESM^Q) and key-value edge selection modules for N2E attention (ESM$^{\text{N2E}}$) and E2E attention (ESM$^{\text{E2E}}$). The nodes and edges may require different sets of edges to update their features, and some pruned edges may help to update nodes or selected edges.

For example, an edge between a person and a background, *e.g.*, an ocean, is invalid but can help to predict the relationships between a person and other objects. Only the valid edges extracted by ESM$^{\text{N2E}}$ and ESM$^{\text{E2E}}$ are used to update the features of the nodes and valid edges from ESM^Q. Finally, the output features are then passed into a classifier to predict relationship classes. The remainder of this section presents the details of each component and training procedure (Sec. 4.4). In this section, the calligraphic font, *i.e.*, $\mathcal{N}$ and $\mathcal{E}$, denotes a set of features while the italic, *i.e.*, N and E, denotes a matrix of stacked features of the set.

4.1. Node detection for object candidates

Given an image I, we use a pre-trained object detector, *i.e.*, Faster R-CNN [31] in our experiments, to extract object bounding boxes and their class labels. Let $b_i \in [0,1]^4$ be the i-th object box coordinate and $v_i \in \mathbb{R}^{d_v}$ its visual feature where d_v is the dimension of the visual feature. We construct a node feature f_i by transforming b_i and v_i via

$$f_i = W_o[W_v v_i; W_g b_i], \tag{1}$$

where W_o, W_v, and W_g are linear transformation matrices and $[\cdot; \cdot]$ is a concatenation operation. The edge feature f_{ij} is formed by concatenating two node features f_i and f_j and performing a linear transformation as

$$f_{ij} = W_p[f_i; f_j], \tag{2}$$

where W_p is the linear transformation matrix. As in Fig. 2, the set of entire node features $\mathcal{N} = \{f_i | 1 \leq i \leq n\}$ and the set of all possible edge features $\mathcal{E} = \{f_{ij} | 1 \leq i, j \leq n, i \neq j\}$ are passed into the edge selection and quad attention modules, whose details are described below. We denote the stacks of the features in $\mathcal{N}$ and $\mathcal{E}$ as N and E for the sake of simplicity.
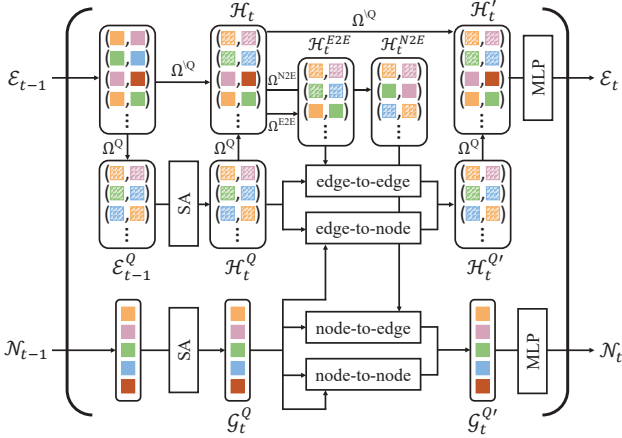
Figure 3. Detailed architecture of the quad attention. The node features are updated by node-to-node and node-to-edge attentions, and the valid edge features, selected by $\mathrm{ESM}^{\mathrm{Q}}$, are updated by edge-to-node and edge-to-edge attentions. The key-value of node-to-edge and edge-to-edge attentions are selected by $\mathrm{ESM}^{\mathrm{N2E}}$ and $\mathrm{ESM}^{\mathrm{E2E}}$. See Sec. 4.3 for the details. Best viewed in color.

4.2. Edge selection for relevant object pairs

While node features N and edge features E can be updated via attentive message passing, a large number of irrelevant edges in E interferes with the attention process. We thus propose to prune invalid edges (*i.e.*, non-existing/false relationships) before proceeding to the quad attention module, which will be described in the next subsection.

In order to remove such distracting edges, we introduce an edge selection module (ESM) that takes an edge feature f_{ij} between nodes i and j and predicts its relatedness score s_{ij} using a simple multi-layer perceptron. We choose the pairs with top-$\rho\%$ highest relatedness scores as valid edges to use in the following quad attention module.

As mentioned earlier, we use three edge selection modules: $\mathrm{ESM}^{\mathrm{Q}}$, $\mathrm{ESM}^{\mathrm{N2E}}$, and $\mathrm{ESM}^{\mathrm{E2E}}$. Each edge selection module ESM^a takes the initial edge features $\mathcal{E}$ as inputs and outputs the valid edge index set Ω for each module, resulting in Ω^{Q}, Ω^{E2E}, and Ω^{N2E}.

4.3. Quad attention for relationship prediction

To capture contextual interactions between the nodes and the edges, we propose a quad attention scheme inspired by the transformer decoder [38]. The main component of the quad attention is multi-head attention:

$$\mathrm{MHA}(Q, K, V) = [\mathrm{HA}_1; \cdots; \mathrm{HA}_h] W^O \quad (3)$$

$$\mathrm{HA}_i = \mathrm{softmax}\left(\frac{(QW_i^{\mathrm{Q}})(KW_i^{\mathrm{K}})^T}{\sqrt{d_k}}\right) V W_i^{\mathrm{V}}, \quad (4)$$

where Q, K, and V are query, key, and value matrices. $W_i^{\mathrm{Q}}, W_i^{\mathrm{K}}$, and W_i^{V} are learnable transformation parame-

ters for Q, K, and V, respectively, d_k is the dimension of the query vector, and W^{O} is a learnable transformation parameter for the output. Each attention head HA_i captures the information from different representation subspaces in parallel, and the multi-head attention aggregates them.

Fig. 3 shows the architecture of our quad attention layer. Following the transformer decoder layer, the t-th quad attention layer takes output edge features E_{t-1} and node features N_{t-1} from the $(t-1)$-th layer as its input and update them with a self-attention first. Instead of updating all possible edge features E_{t-1}, we only update the valid edge features E_{t-1}^{Q}, whose indices are in Ω^{Q} extracted from $\mathrm{ESM}^{\mathrm{Q}}$:

$$G_t = \mathrm{LN}(N_{t-1} + \mathrm{MHA}(N_{t-1}, N_{t-1}, N_{t-1})), \quad (5)$$

$$H_t^{\mathrm{Q}} = \mathrm{LN}(E_{t-1}^{\mathrm{Q}} + \mathrm{MHA}(E_{t-1}^{\mathrm{Q}}, E_{t-1}^{\mathrm{Q}}, E_{t-1}^{\mathrm{Q}})), \quad (6)$$

where LN is layer normalization, G_t and H_t are the output of the self-attention layer for node features and valid edge features, respectively. For key-value edge features of N2E and E2E attentions, we extract the key-value set from the updated entire edge set $\mathcal{H}_t = \mathcal{H}_t^{\mathrm{Q}} \cup \mathcal{E}^{\backslash \mathrm{Q}}$, where $\mathcal{H}_t^{\mathrm{Q}}$ is the set of updated valid edges for query and $\mathcal{E}^{\backslash \mathrm{Q}} = \mathcal{E} - \mathcal{E}^{\mathrm{Q}}$. Then, H_t^{Q} are refined by E2N and E2E attentions and G_t are refined by N2N and N2E attentions:

$$G_t' = \mathrm{LN}(G_t + \underbrace{\mathrm{MHA}(G_t, G_t, G_t)}_{\text{node-to-node attention}} \quad (7)$$
$$+ \underbrace{\mathrm{MHA}(G_t, H_t^{\mathrm{N2E}}, H_t^{\mathrm{N2E}})}_{\text{node-to-edge attention}}),$$

$$H_t^{\mathrm{Q}\prime} = \mathrm{LN}(H_t^{\mathrm{Q}} + \underbrace{\mathrm{MHA}(H_t^{\mathrm{Q}}, G_t, G_t)}_{\text{edge-to-node attention}} \quad (8)$$
$$+ \underbrace{\mathrm{MHA}(H_t^{\mathrm{Q}}, H_t^{\mathrm{E2E}}, H_t^{\mathrm{E2E}})}_{\text{edge-to-edge attention}}),$$

where H_t^{N2E} and H_t^{E2E} are selected by the indices Ω^{N2E} and Ω^{E2E} from the stack of $\mathcal{H}_t$, *i.e.*, H_t. Each attention explicitly represents a particular type of relationship between edges and nodes and helps to construct contextual information for the scene graph generation. Lastly, G_t' and H_t' are further updated by multi-layer perceptron (MLP) followed by the residual connection and a layer normalization:

$$N_t = \mathrm{LN}(G_t' + \mathrm{MLP}(G_t')) \quad (9)$$
$$E_t = \mathrm{LN}(H_t' + \mathrm{MLP}(H_t')), \quad (10)$$

where H_t' is the stack of $\mathcal{H}_t' = \mathcal{H}^{\mathrm{Q}\prime} \cup \mathcal{E}^{\backslash \mathrm{Q}}$, and the quad attention layer outputs N_t and E_t.

The inputs N_0 and E_0 of the first quad attention layer are the entire node features N and all possible edge features E,

which are defined in Sec. 4.1. Every quad attention layer uses the same valid edge sets to update the node features and valid edge features by four types of attention. Given the output edge features E_T of the last T-th quad attention layer, each edge feature $e_{ij} \in \mathcal{E}_T$ is passed into a feedforward MLP to produce a probability distribution y_{ij} over the predicate classes $\mathcal{P}$.

4.4. Training objective

To train SQUAT, we use a combination of two loss functions: a cross-entropy loss for the predicate classification and a binary cross-entropy loss for the edge selection module. The first predicate classification loss is defined as:

$$\mathcal{L}_{\text{PCE}} = \frac{1}{|\mathcal{E}|} \sum_{i,j=1, i \neq j}^{|\mathcal{N}|} \mathcal{L}_{\text{CE}}(y_{ij}, \hat{y}_{ij}), \qquad (11)$$

where $\mathcal{L}_{\text{CE}}$ is the cross-entropy loss and $\hat{y}_{ij}$ is a one-hot vector of ground-truth relationship labels $\hat{p}_{ij}$ between object i and object j. To train the edge selection module, we use auxiliary binary cross-entropy defined as:

$$\mathcal{L}_{\text{ESM}}^a = \frac{1}{|\mathcal{E}|} \sum_{i,j=1, i \neq j}^{|\mathcal{N}|} \mathcal{L}_{\text{BCE}}(s_{ij}^a, \hat{s}_{ij}), \qquad (12)$$

where $\mathcal{L}_{\text{BCE}}$ is the binary cross-entropy loss, $\hat{s}_{ij}$ is the binary indicator of whether object i and object j have a relationship or not, and $a \in \mathcal{A} = \{Q, E2E, N2E\}$. The entire loss is defined as:

$$\mathcal{L} = \mathcal{L}_{\text{PCE}} + \lambda \frac{1}{|\mathcal{A}|} \sum_{a \in \mathcal{A}} \mathcal{L}_{\text{ESM}}^a, \qquad (13)$$

where $\lambda > 0$ is a hyper-parameter. In training, $\mathcal{L}_{\text{CE}}$ does not affect the parameters of ESM directly due to the hard selection of ESM, and the gradient passes on to train the edge feature extraction; ESM is mainly trained by $\mathcal{L}_{\text{ESM}}$.

5. Experiments

In this section, we perform a diverse set of experiments to evaluate the proposed model. We use two datasets: 1) Visual Genome (VG) [18] and 2) OpenImages v6 [17] datasets to train and evaluate model performances. We intend to show that our model can be generalized over heterogeneous cases by demonstrating competitive results on the two independent datasets.

5.1. Datasets and evaluation metrics

5.1.1 Visual Genome [18]

The Visual Genome dataset is composed of 108k images with an average of 38 objects and 22 relationships per image. However, most of the predicate classes have less than 10 samples. Therefore, we adopt the widely-used VG split [21, 50] to select the most frequent 150 object classes and 50 predicate classes. Following the [50], we first split the dataset into a training set (70%) and a test set (30%). Then, we sample 5k validation images from the training set to tune the hyperparameters. We evaluate SQUAT on three subtasks: Predicate Classification (PredCls), Scene Graph Classification (SGCls), and Scene Graph Detection (SGDet). The PredCls predicts the relationships given the ground-truth bounding boxes and object labels, the SGCls aims to predict the object labels and the relationships given the ground-truth bounding boxes, and the SGDet targets predicting the object bounding boxes, object labels, and relationships without any ground-truth. As the evaluation metrics, we adopt the mean recall@K (mR@K), as previously used in scene graph generation literature [2, 35]. mR@K is the average of recall@K for each relation. Following [43], we apply the graph-constraint, in which each object pair can have only one relationship, for evaluation.

5.1.2 OpenImages v6 [17]

The OpenImages v6 dataset has 126,368 images for the training, 1,813 images for the validation, and 5,322 images for the test. Each image in the dataset has 4.1 objects and 2.8 relationships on average. The dataset has 301 object classes and 31 predicate classes. Compared with the Visual Genome dataset, the quality of annotation is far more robust and complete. For OpenImages v6, following [17, 51], we calculate Recall@50 (R@50), weighted mean AP of relationships (wmAP$_{\text{rel}}$), and weighted mean AP of phrases (wmAP$_{\text{phr}}$) as evaluation metrics. AP$_{\text{rel}}$ evaluates the two object bounding boxes, the subject box and the object box, and three labels, the triplets of the subject, the object, and the predicate. AP$_{\text{phr}}$ evaluates a union bounding box of subject and object and three labels as the same as AP$_{\text{rel}}$. To reduce the dataset bias in evaluation, we calculate wmAP$_{\text{rel}}$ and wmAP$_{\text{phr}}$ with weighted average of per-relationship AP$_{\text{phr}}$ and AP$_{\text{phr}}$, respectively. The weight of each relationship is calculated by their relative ratios in the validation set. The final score score$_{\text{wtd}}$ is obtained as $0.2 \times \text{R@50} + 0.4 \times \text{wmAP}_{\text{rel}} + 0.4 \times \text{wmAP}_{\text{phr}}$.

5.2. Implementation details

As in the previous work [19, 36], we adopt ResNeXt-101-FPN as a backbone network and Faster R-CNN as an object detector. The model parameters of the pre-trained object detector are frozen during the training time. We use a bi-level sampling [19] to handle the long-tailed distribution of the datasets. The hyperparameters of bi-level sampling are set the same as in [19]. We set the hyper-parameter $\lambda = 0.1$ for the loss function. The keeping ratio ρ is set to 70% for the SGDet setting on both the Visual Genome dataset and the OpenImages v6 dataset in the training. In

Methods	PredCls			SGCls			SGDet		
	mR@20	mR@50	mR@100	mR@20	mR@50	mR@100	mR@20	mR@50	mR@100
IMP+[‡] [43]	8.9	11.0	11.8	5.2	6.2	6.5	2.8	4.2	5.3
Motifs[‡] [50]	11.5	14.6	15.8	6.5	8.0	8.5	4.1	5.5	6.8
RelDN [51]	-	15.8	17.2	-	9.3	9.6	-	6.0	7.3
VCTree[‡] [36]	12.4	15.4	16.6	6.3	7.5	8.0	4.9	6.6	7.7
MSDN [21]	-	15.9	17.5	-	9.3	9.7	6.1	7.2	
GPS-Net [23]	-	15.2	16.6	-	8.5	9.1	-	6.7	8.6
RU-Net [25]	-	-	24.2	-	-	14.6	-	-	10.8
HL-Net [24]	-	-	22.8	-	-	13.5	-	-	9.2
VCTree-TDE [35]	18.4	25.4	28.7	8.9	12.2	14.0	6.9	9.3	11.1
Seq2Seq [28]	21.3	26.1	30.5	11.9	14.7	16.2	7.5	9.6	12.1
GPS-Net[†]	21.5	27.1	29.1	6.4	10.1	12.3	6.6	9.4	11.9
JMSGG [44]	-	24.9	28.0	-	13.1	14.7	-	9.8	11.8
BGNN[†] [19]	-	30.4	32.9	-	14.3	16.5	-	10.7	12.6
SQUAT [†] (Ours)	**25.6**	**30.9**	**33.4**	**14.4**	**17.5**	**18.8**	**10.6**	**14.1**	**16.5**

Table 1. The scene graph generation performance of three subtasks on Visual Genome (VG) dataset with graph constraints. † denotes that the bi-level sampling [19] is applied for the model. ‡ denotes that the results are reported from the [35].

Methods	R@50	wmAP		score_wtd
		rel	phr	
RelDN [51]	73.1	32.2	33.4	40.9
VCTree [36]	76.1	34.2	33.1	42.1
Motifs [50]	71.6	29.9	31.6	38.9
VCTree+TDE [35]	69.3	30.7	32.8	39.3
GPS-Net [23]	74.8	32.9	34.0	41.7
GPS-Net[†] [23]	74.7	32.8	33.9	41.6
BGNN[†] [19]	75.0	33.5	34.2	42.1
HL-Net [24]	76.5	35.1	34.7	43.2
RU-Net [25]	**76.9**	**35.4**	34.9	**43.5**
SQUAT [†]	75.8	34.9	**35.9**	**43.5**

Table 2. The scene graph generation performance on OpenImages v6 dataset with graph constraints. † denotes that the bi-level sampling [19] is applied for the model.

the early stages of training, the edge selection model is not reliable, causing instability during training. To tackle the issue, we pre-trained the edge selection module for a few thousand iterations using $\mathcal{L}_{ESM}$ while freezing all other parameters and then trained the entire SQUAT except for the object detector. Complete implementation details are specified in the supplementary material.

5.3. Comparison with state-of-the-art models

As shown in Table 1, on the Visual Genome dataset, SQUAT outperforms the state-of-the-art models on every setting, PredCls, SGCls and SGDet. Especially, SQUAT outperforms the state-of-the-art models by a large margin of 3.9 in mR@100 on the SGDet setting, which is the most realistic and important setting in practice as there is no perfect object detector. There are more invalid pairs in the

SGDet setting than in other settings since the detected object bounding boxes from the pre-trained object detector includes many background boxes. This means that previous work for contextual modeling was most likely distracted by the invalid pairs; thus, SQUAT shows significant performance improvement on the SGDet setting. BGNN [19] also leverage a scoring function to scale the messages of the invalid edges, however, SQUAT shows better results with our edge selection module to discard invalid object pairs. This becomes doable with the quad attention mechanism with edge selection which helps to reduce noise and outliers from invalid pairs more effectively. Also, SQUAT shows the performance improvements by 2.3 and 0.5 on the SG-Cls and the PredCls settings with mR@100, respectively; the more complex and realistic the task, the more noticeable the performance improvement of SQUAT becomes. It shows that SQUAT, composed of edge selection and quad attention, is appropriate for contextual reasoning to generate scene graphs even in a complex scene.

Also, as shown in Table 2, SQUAT achieve competitive results or even outperform compared with the state-of-the-art models on the OpenImages v6 dataset with score_wtd. Since there are fewer objects and relationships in the images of the OpenImages v6 dataset than of the Visual Genome, the edge selection module seems less effective for the OpenImages v6 dataset. As there is a trade-off in recall and mean recall when bi-level sampling is utilized [2,28], the result of SQUAT in Table 2 is a compromised metric for R@50. But still, the R@50 of SQUAT is still competitive with that from RU-Net [25] and outperforms other recent baselines, and we achieve the best performance in wmAP_phr by a large margin. It shows SQUAT is effective in improving the scene graph generation performance and also in simple scenes.

Variants			SGDet		
Q	E2E	N2E	mR@20	mR@50	mR@100
BGNN [19]			7.49	10.31	12.46
			9.12	12.45	15.00
✓			9.92	13.22	15.66
	✓	✓	9.84	13.04	15.60
✓	✓	✓	10.57	14.12	16.47

Table 3. The ablation study on model variants on edge selection. We remove the edge selection module for query selection and key-value selection.

Variants			SGDet		
Q	E2E	N2E	mR@20	mR@50	mR@100
shared	shared	shared	9.61	12.70	14.85
distinct	shared	shared	9.63	12.54	14.64
distinct	distinct	distinct	10.57	14.12	16.47

Table 4. The ablation study on model variants on edge selection. We share the edge selection module for query selection and key-value selection. 'shared' denotes the edge selection modules share the parameters.

Method				SGDet		
N2N	N2E	E2N	E2E	mR@20	mR@50	mR@100
✓	✓			7.02	9.74	11.57
✓	✓		✓	9.76	12.98	15.30
✓	✓	✓		9.70	12.27	15.03
		✓	✓	9.90	13.05	15.28
	✓	✓	✓	9.77	12.93	15.42
✓		✓	✓	9.99	13.02	15.54
✓	✓	✓	✓	10.57	14.12	16.47

Table 5. The ablation study on model variants on quad attention. N2N, N2E, E2N, E2E denote the node-to-node, node-to-edge, edge-to-node, and edge-to-edge attentions, respectively.

Qualitative visualizations of SQUAT and more experiments on SQUAT with 1) additional measurement, *e.g.*, recall and non-graph constraint measurement, on Visual Genome, 2) performance with plug-and-play long-tailed recognition techniques and 3) additional qualitative results are given in Supplementary.

5.4. Ablation study

5.4.1 Model variants on edge selection

To verify the effectiveness of the edge selection module, we evaluate the model from which each component of edge selection is removed on the Visual Genome dataset. As shown in Table 3, we observe that the quad attention module without the edge selection module shows much lower performance at mR@100 (-8.9%) than the full model which has the edge selection module; thus, to select the valid edges is important for the scene graph generation. On the other hand, the quad attention module without the edge selection module shows 20.4% higher performance than the BGNN and achieves 15.00 on mR@100. It shows the effectiveness of the quad attention module itself without the edge selection module. We also observe that the query selection is more critical than the key-value selection for the scene graph generation; it shows that selecting what to update is important for the scene graph generation.

To evaluate the effectiveness of three distinct edge selection modules, we evaluate the models, some of which

edge selection modules are shared. In Table 4, ESM^a, of which $a \in \{Q, E2E, N2E\}$ are denoted 'shared' in the column, share the same parameters. We observe that the three fully-distinct edge selection modules boost the scene graph generation performances. It shows there exist differences between the edges needed to update both features and the edges that need to be updated.

5.4.2 Model variants on quad attention

To verify the effectiveness of the quad attention module, we evaluate the model from which each attention is removed on the Visual Genome dataset. As shown in Table 5, the full model with quad attention outperforms the other model variants. We also observe that the SQUAT without updating edges, *i.e.*, edge-to-node and edge-to-edge attentions, performs poorer than the SQUAT without updating nodes, *i.e.*, node-to-node and node-to-edge attentions. It shows that updating edge features with context information is important for context reasoning. Especially, SQUAT without edge-to-edge attention shows worse performance than without edge-to-node attention since edge-to-edge attention, which is neglected in the previous work, can capture high-level information.

5.5. The effect of the edge selection module

We applied the edge selection module to the scene graph generation model with message passing methods. Since it is not known which pairs of objects have a relationship, the message passing methods use the fully-connected graph in the inference time. However, we empirically observe that message passing on fully-connected graph is meaningless or even harmful for the scene graph generation. We use three baselines, IMP [43], BGNN [19], and SQUAT, for ablation study on the Visual Genome dataset. Table 6 shows that message passing through fully-connected graph is harmful for the scene graph generation. Even though BGNN uses a gating function to scale down the messages from invalid edges, it does not work well.
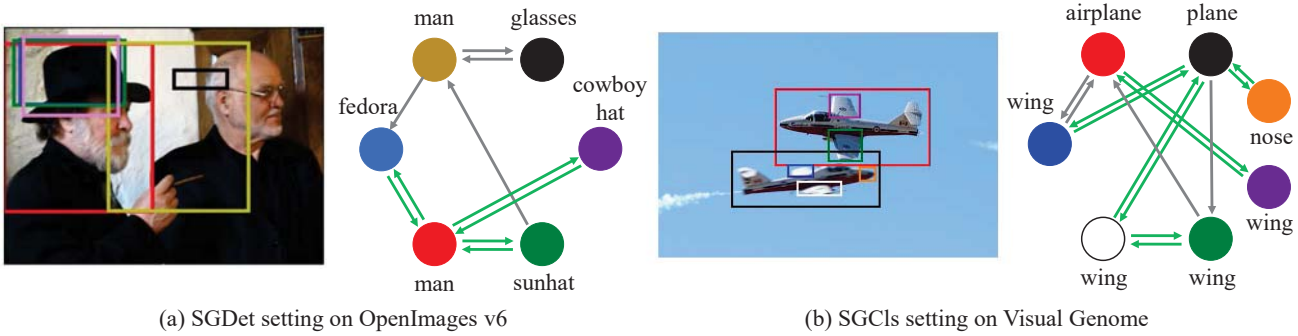
(a) SGDet setting on OpenImages v6

(b) SGCls setting on Visual Genome

Figure 4. Qualitative results for edge selection module ESM^Q for query selection. The selected edges after edge selection are drawn in the right graph. The green arrows denote the valid pairs, and the gray arrows denote the invalid pairs. The keeping ratio for the two settings is the same $\rho = 35\%$. All of the valid edges remain, and most of the invalid edges are removed.

| model | Graph | SGDet | | |
		mR@20	mR@50	mR@100
IMP [43]	No	4.09	5.56	6.53
	Full	2.87	4.24	5.42
BGNN [19]	No	8.99	11.84	13.56
	Full	7.49	10.31	12.46
	ES	9.00	11.86	14.20
	GT	14.15	16.41	17.09
SQUAT	No	8.68	11.52	13.99
	Full	9.12	12.45	15.00
	ES	10.57	14.12	16.47
	GT	17.95	19.21	19.51

Table 6. The ablation study on message passing for the scene graph generation. There are four settings depending on which graphs are used in the message passing: No, Full, ES, and GT. Every result is reproduced with the authors' code.

To further investigate the effect of edge selection, we applied message passing through ground-truth scene graphs to each model. In table 6, 'No' represents that message passing is not used, 'Full' and 'GT' indicate that message passing is used with the complete graph and the ground-truth scene graph, respectively; 'ES' means that the proposed edge selection module is used with message passing. As shown in Table 6, every model with the message passing through ground truth outperforms state-of-the-art models by a substantial margin, showing that removing the invalid edges is crucial for scene graph generation. The edge selection module clearly improves not only the performance of SQUAT but also that of BGNN, the previous state-of-the-art model. It indicates that the edge selection module effectively removes the invalid edges and can be used as a plug-and-play module for message-passing-based scene graph methods.

5.6. Qualitative results

Qualitative results for the edge selection module are shown in Fig. 4. As shown in Fig. 4 (a), the object detection module extracts 6 bounding boxes, then the fully-connected graph has 30 edges in total, where only 6 valid edges are in the ground-truth. After edge selection with keeping ratio $\rho = 35\%$, only 10 edges remain where 6 valid edges all remain. It significantly reduces noises from invalid edges. The other example in Fig. 4 (b) shows the same tendency.

6. Conclusion

We presented a novel scene graph generation model that predicts a scene graph within an image. The method is designed to selectively utilize valid edges using our proposed quad attention module, and update the model from the valid edges only. The edge selection module effectively filters out invalid edges to sparsify a noisy scene graph, and thus it removes uncertainties brought by invalid edges. The quad attention module, which is composed of four components — node-to-node, node-to-edge, edge-to-node, and edge-to-edge attentions — captures the high-level information for accurately predicting relationships among different objects. We have shown the effectiveness of the SQUAT, and each component under various settings was properly validated in the experiments to demonstrate stability.

References

[1] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020. 2

[2] 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, 5, 6

[3] 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. 2

[4] Krzysztof Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Davis, Afroz Mohiuddin, Lukasz Kaiser, et al. Rethinking attention with performers. *International Conference on Learning Representations*, 2021. 2

[5] Bo Dai, Yuqi Zhang, and Dahua Lin. Detecting visual relationships with deep relational networks. In *Proceedings of the IEEE conference on computer vision and Pattern recognition*, pages 3076–3086, 2017. 2

[6] Alakh Desai, Tz-Ying Wu, Subarna Tripathi, and Nuno Vasconcelos. Learning of visual relations: The devil is in the tails. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 15404–15413, October 2021. 2

[7] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 2

[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. 2

[9] Yuyu Guo, Lianli Gao, Xuanhan Wang, Yuxuan Hu, Xing Xu, Xu Lu, Heng Tao Shen, and Jingkuan Song. From general to specific: Informative scene graph generation via balance adjustment. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 16383–16392, October 2021. 2

[10] 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

[11] 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

[12] Siddhesh Khandelwal, Mohammed Suhail, and Leonid Sigal. Segmentation-grounded scene graph generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15879–15889, 2021. 2

[13] Jinwoo Kim, Saeyoon Oh, and Seunghoon Hong. Transformers generalize deepsets and can be extended to graphs & hypergraphs. *Advances in Neural Information Processing Systems*, 34:28016–28028, 2021. 2

[14] Nikita Kitaev, Łukasz Kaiser, and Anselm Levskaya. Reformer: The efficient transformer. *International Conference on Learning Representations*, 2020. 2

[15] Boris Knyazev, Harm de Vries, Cătălina Cangea, Graham W Taylor, Aaron Courville, and Eugene Belilovsky. Graph density-aware losses for novel compositions in scene graph generation. *arXiv preprint arXiv:2005.08230*, 2020. 2

[16] Boris Knyazev, Harm de Vries, Cătălina Cangea, Graham W. Taylor, Aaron Courville, and Eugene Belilovsky. Generative compositional augmentations for scene graph prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 15827–15837, October 2021. 2

[17] Ivan Krasin, Tom Duerig, Neil Alldrin, Vittorio Ferrari, Sami Abu-El-Haija, Alina Kuznetsova, Hassan Rom, Jasper Uijlings, Stefan Popov, Shahab Kamali, Matteo Malloci, Jordi Pont-Tuset, Andreas Veit, Serge Belongie, Victor Gomes, Abhinav Gupta, Chen Sun, Gal Chechik, David Cai, Zheyun Feng, Dhyanesh Narayanan, and Kevin Murphy. Openimages: A public dataset for large-scale multi-label and multi-class image classification. *Dataset available from https://storage.googleapis.com/openimages/web/index.html*, 2017. 5

[18] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, Michael Bernstein, and Li Fei-Fei. Visual genome: Connecting language and vision using crowdsourced dense image annotations. 2016. 5

[19] Rongjie Li, Songyang Zhang, Bo Wan, and Xuming He. 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*, pages 11109–11119, 2021. 1, 2, 5, 6, 7, 8

[20] Yikang Li, Wanli Ouyang, Bolei Zhou, Jianping Shi, Chao Zhang, and Xiaogang Wang. Factorizable net: an efficient subgraph-based framework for scene graph generation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 335–351, 2018. 2

[21] 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. 1, 2, 5, 6

[22] Kevin Lin, Lijuan Wang, and Zicheng Liu. Mesh graphormer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 12939–12948, 2021. 2

[23] Xin Lin, Changxing Ding, Jinquan Zeng, and Dacheng Tao. Gps-net: Graph property sensing network for scene graph generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3746–3753, 2020. 2, 6

[24] Xin Lin, Changxing Ding, Yibing Zhan, Zijian Li, and Dacheng Tao. Hl-net: Heterophily learning network for scene graph generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19476–19485, 2022. 6

[25] Xin Lin, Changxing Ding, Jing Zhang, Yibing Zhan, and Dacheng Tao. Ru-net: Regularized unrolling network for scene graph generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19457–19466, 2022. 6

[26] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021. 2

[27] Cewu Lu, Ranjay Krishna, Michael Bernstein, and Li Fei-Fei. Visual relationship detection with language priors. In *European Conference on Computer Vision*, 2016. 1

[28] Yichao Lu, Himanshu Rai, Jason Chang, Boris Knyazev, Guangwei Yu, Shashank Shekhar, Graham W Taylor, and Maksims Volkovs. Context-aware scene graph generation with seq2seq transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15931–15941, 2021. 1, 2, 6

[29] Erxue Min, Runfa Chen, Yatao Bian, Tingyang Xu, Kangfei Zhao, Wenbing Huang, Peilin Zhao, Junzhou Huang, Sophia Ananiadou, and Yu Rong. Transformer for graphs: An overview from architecture perspective. *arXiv preprint arXiv:2202.08455*, 2022. 2

[30] Mengshi Qi, Weijian Li, Zhengyuan Yang, Yunhong Wang, and Jiebo Luo. Attentive relational networks for mapping images to scene graphs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3957–3966, 2019. 2

[31] 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, 2015. 2, 3

[32] Byungseok Roh, JaeWoong Shin, Wuhyun Shin, and Saehoon Kim. Sparse detr: Efficient end-to-end object detection with learnable sparsity. *arXiv preprint arXiv:2111.14330*, 2021. 2

[33] Yu Rong, Yatao Bian, Tingyang Xu, Weiyang Xie, Ying Wei, Wenbing Huang, and Junzhou Huang. Self-supervised graph transformer on large-scale molecular data. *Advances in Neural Information Processing Systems*, 33:12559–12571, 2020. 2

[34] 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

[35] 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. 5, 6

[36] 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, 5, 6

[37] Damien Teney, Lingqiao Liu, and Anton van Den Hengel. Graph-structured representations for visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2017. 1

[38] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 2, 4

[39] Sinong Wang, Belinda Z Li, Madian Khabsa, Han Fang, and Hao Ma. Linformer: Self-attention with linear complexity. *arXiv preprint arXiv:2006.04768*, 2020. 2

[40] Tzu-Jui Julius Wang, Selen Pehlivan, and Jorma Laaksonen. Tackling the unannotated: Scene graph generation with bias-reduced models. *arXiv preprint arXiv:2008.07832*, 2020. 2

[41] Wenbin Wang, Ruiping Wang, Shiguang Shan, and Xilin Chen. Exploring context and visual pattern of relationship for scene graph generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8188–8197, 2019. 2

[42] Sanghyun Woo, Dahun Kim, Donghyeon Cho, and In So Kweon. Linknet: Relational embedding for scene graph. *Advances in Neural Information Processing Systems*, 31, 2018. 2

[43] 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. 1, 2, 5, 6, 7, 8

[44] Minghao Xu, Meng Qu, Bingbing Ni, and Jian Tang. Joint modeling of visual objects and relations for scene graph generation. *Advances in Neural Information Processing Systems*, 34, 2021. 6

[45] 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. 2

[46] 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. 1, 2

[47] Yuan Yao, Ao Zhang, Xu Han, Mengdi Li, Cornelius Weber, Zhiyuan Liu, Stefan Wermter, and Maosong Sun. Visual distant supervision for scene graph generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 15816–15826, October 2021. 2

[48] Guojun Yin, Lu Sheng, Bin Liu, Nenghai Yu, Xiaogang Wang, Jing Shao, and Chen Change Loy. Zoom-net: Mining deep feature interactions for visual relationship recognition. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 322–338, 2018. 2

[49] Alireza Zareian, Svebor Karaman, and Shih-Fu Chang. Bridging knowledge graphs to generate scene graphs. In

European Conference on Computer Vision, pages 606–623. Springer, 2020. 2

[50] 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. 1, 2, 5, 6

[51] Ji Zhang, Kevin J Shih, Ahmed Elgammal, Andrew Tao, and Bryan Catanzaro. Graphical contrastive losses for scene graph generation. 2019. 5, 6

[52] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip HS Torr, et al. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6881–6890, 2021. 2

[53] 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

[54] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. *International Conference on Learning Representations*, 2021. 2