BGT-Net: Bidirectional GRU Transformer Network for Scene Graph Generation

Naina Dhingra   Florian Ritter   Andreas Kunz

Innovation Center Virtual Reality, ETH Zurich

{ndhingra, kunz}@iwf.mavt.ethz.ch; ritterf@ethz.ch

Abstract

Scene graphs are nodes and edges consisting of objects and object-object relationships, respectively. Scene graph generation (SGG) aims to identify the objects and their relationships. We propose a bidirectional GRU (BiGRU) transformer network (BGT-Net) for the scene graph generation for images. This model implements novel object-object communication to enhance the object information using a BiGRU layer. Thus, the information of all objects in the image is available for the other objects, which can be leveraged later in the object prediction step. This object information is used in a transformer encoder to predict the object class as well as to create object-specific edge information via the use of another transformer encoder. To handle the dataset bias induced by the long-tailed relationship distribution, softening with a log-softmax function and adding a bias adaptation term to regulate the bias for every relation prediction individually showed to be an effective approach. We conducted an elaborate study on experiments and ablations using open-source datasets, i.e., Visual Genome, Open-Images, and Visual Relationship Detection datasets, demonstrating the effectiveness of the proposed model over state of the art.

1. Introduction

Visual understanding of scenes are broadly covered by object detection [30, 23] and localization [29, 6] for single or multiple objects. Evolved from detection of various objects, image segmentation [7, 1] is another research topic which helps to understand the attributes of the scene. While these techniques supply some useful information of the image, but a scene also largely depends on the interactions or relations between objects. This idea led to scene graphs which describe the scene by incorporating the objects and their pairwise relations. This relation is represented by a directed edge pointing from the subject to the object. Evaluating a scene by detecting the objects and the relationships between them allows building a graph consisting of nodes representing the objects and the edges representing the relations. It consists of number of triplets represented in <subject-relationship-object> form. They help in aid-
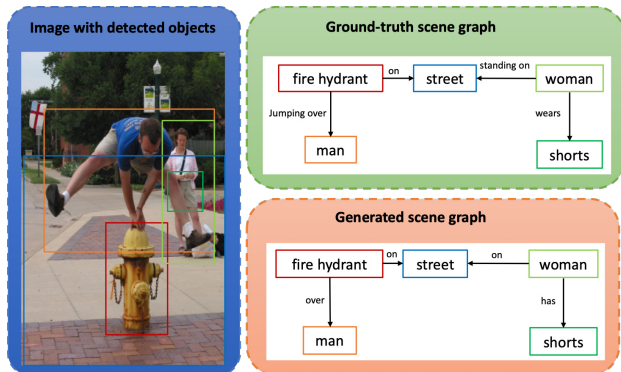

Figure 1: Two different scene graphs of the same given image on the left. Top: The ground-truth scene graph as annotated in the Visual Genome dataset. Below: A generated scene graph. As in many cases, the generated scene graph does capture the visual scene correctly, since the scene graph generation model only predicts frequently appearing relationships.

ing deep understanding of the scene for various vision tasks such as visual reasoning [28], image captioning [40, 16], image retrieval [12, 27, 26] and visual question answering [47, 8, 34]. To improve these applications and their benefits, it is crucial to have a well performing model which generates scene graphs that corresponds to the actual visual scene.

The directional nature of a triplet in scene graph defines the subject and object in a triplet. Scene graph generation (SGG) is considered a complex problem in computer vision because of the imbalanced nature of the datasets and intricate relationship information. As stated in MOTIFS [45], combinations of at least two triplets appear in many images. So, the presence of some objects highly increase the probability of the presence of other objects. Communication i.e. information flow between the detected objects has been shown to be beneficial for improving the performance of scene graph generation models [45, 4].

The frequency distribution of the relationship within the Visual Genome dataset is long-tailed [13]. Due to this, scene graph detection models can already achieve good performance by only predicting the most frequent relationship for

the respective subject-object pair. Figure 1 illustrates the problem almost every SGG model faces. In many object pairings, the relationship is trivial and mostly possessive or geometric (e.g. *on*, *under*, or *next to*). The detailed descriptive and semantic relationships such as *jumping over* as shown in Figure 1 (given in the ground-truth) will not often be predicted since it rarely occurs. For creating models that represent less frequent relationships more precisely, an approach to handle dataset bias must be found. Since the bias in the dataset can also be beneficial, e.g. the probability for the relationship *reading* will be much higher than for *eating* if the subject-object pair given is person and book [32], traditional debiasing methods will most likely strongly harm the performance of the model. For this reason, the handling of the bias in the dataset is one of the most under-explored properties of the scene graph generation task.

In this paper, we propose a novel model for SGG. The objects present in an image are highly dependent on the presence of other objects, for instance, if there is a bike in the scene, then there will be two tyres in the same scene with a very high probability. This model uses a bidirectional GRU (BiGRU) layer to send information from every object to every other. This allows benefiting from the fact that some objects will increase the possibility for specific other objects to be present.

Subsequently to this layer that covers the object-object communication, a transformer encoder layer is used to predict the object classes. Objects and their preferred or observed relations are closely connected. To extract the information for the edge, a similar approach as in [45] is followed to specify the edge context for every detected object. We use an additional transformer encoder layer for this task of extracting the edge context features.

Using the object representations, their respective edge context is then used for the relationship prediction. In this procedure, a log-softmax function is applied to the subject-object pairwise relationship distribution. Following this Frequency Softening (FS), a Bias Adaptation (BA) approach [21] is used. The bias for every subject-object is controlled by the bias adaptation term which takes scene-specific inputs to vary the amount of added bias.

The contribution summary of the proposed BGT-Net is given in four modules to improve scene graph generation performance: (1) Object-object communication is performed using the BiGRU's. (2) A transformer encoder with scaled-dot-product attention is used to predict object classes after they have received information about the other objects present in the scene. (3) For every object, a second transformer encoder is used to gather information for the edges. (4) To tackle the bias in the relationship distribution, FS and BA [21] is adopted.

The evaluation efficacy of the proposed BGT-Net is performed on three SGG datasets: Visual Genome (VG) [13],
OpenImages (OI) [14], and Visual Relationship Detection (VRD) [22]. We perform extensive experiments and ablation study to demonstrate the effectiveness of BGT-Net. Experimental results illustrate that the proposed BGT-Net outperforms the state of the art SGG results on a common metric Recall@K and on different datasets to the best of our knowledge.

2. Related Work

From the ongoing research in scene graph generation, two different approaches for scene graph construction have developed. In the less common two-stage approach [33, 10, 4, 33], attributes of the scene graph are used in the second training step to refine the results produced by the first stage. Much more common are the one-stage approaches [4, 45, 5, 37, 39, 21, 18, 22, 17, 24] which focus only on object detection and relationship classification, while almost neglecting intrinsic features. The proposed BGT-Net follows a one step approach and has the following advantages as compared to the literature work: (1) It uses object-object communication which improves the performance in SGG; (2) It deploys transformer encoder for object and edge context prediction which has shown to be highly beneficial in optimizing the parameters of SGG; (3) It is easy to train.

The MOTIFS [45] stated in the early days of scene graph generation, that there are different combinations of triplets that appear in a lot of images. Therefore, a dependency between object appearances is present in the datasets. To leverage this information object communication has been examined in the CMAT model [4] and improved the performance of the model. All of the published works show difficulties with the bias present in the Visual Genome dataset, which is widely used in the scene graph generation task. This bias arrives from the long-tailed relationship distribution. The GPS-Net [21] tackled this problem with FS and BA which worked well compared to the previous works. The overall performance of the model could be improved as well as improvements in mean Recall@K were achieved, which gives reasoning about the positive effect of their approach in handling the dataset bias. So, motivated from GPS-Net, BGT-Net uses FS and BA. While the GPS-Net changed the way the model was built, another recent work [32] developed a Scene Graph Diagnosis toolkit that can be used on a casually built scene graph. This tool kit is based on casual inference. Drawing the counterfactual causality to the proposed graph allows inferring with the bad bias. This approach did largely improve the mean Recall@K but decreased the other metrics significantly. Similarly [38], adaptively changed the weights of the loss by using the correlation between the relationship classes. This work improved mean Recall@K but had overall quite low Recall@K results.
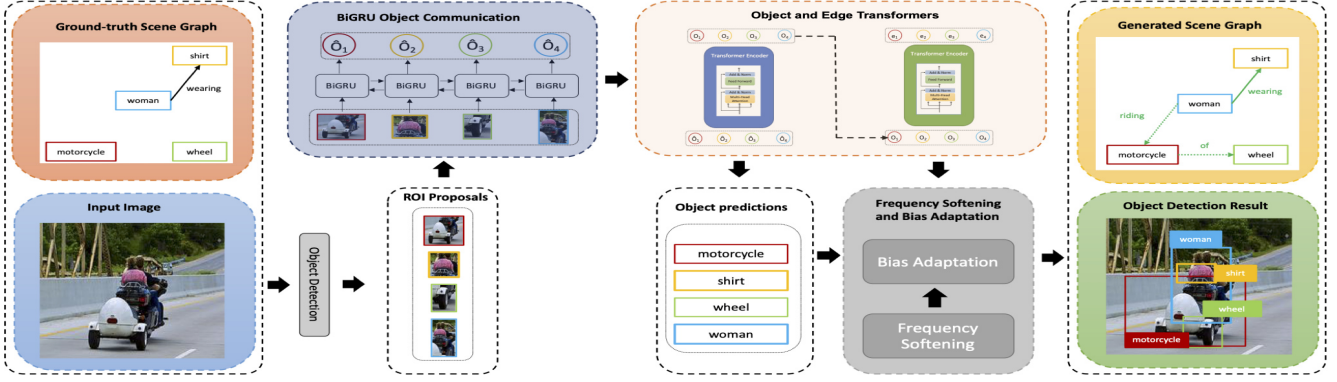
Figure 2: The framework of BGT-Net uses Faster R-CNN (with VGG-16 or ResNext-101 as the backbone) to get the visual features and spatial locations of object proposals. It includes various sub-modules for the task of SGG: (1) New technique of using BiGRU for object-object communication, (2) Novel method of using a transformer encoder with scaled-dot-product attention for predicting object classes after they have received information of the other objects present in the scene, (3) Additional transformer encoder is used to get the edge features, (4) FS and BA are used for dealing with the bias in the dataset.

3. Approach

The illustration of the BGT-Model can be found in Figure 2. The regions of interest and object proposals are obtained by employing a Faster R-CNN object detector [25]. Following the VG-split [37], there are 151 object categories (including 'background') and 50 relationship categories (including 'no relation'). For every proposal i, the visual feature $\hat{x}_i$ is formed by concatenating the ROI (region of interest) feature $f_i \in \mathbb{R}^{2048}$, the class confidence scores $s_i \in \mathbb{R}^{151}$, and the spatial feature of the proposal bounding box $s_i \in \mathbb{R}^4$. For the next step, the $x_i \in \mathbb{R}^{512}$ gets transformed by the projection of $\hat{x}_i$ into a 512-dimensional subspace. For the relationship classification, the union feature $u_{i,j} \in \mathbb{R}^{2048}$ for every pair of objects i and j is extracted in the object detection stage. These feature vectors representing the scene are used in the following modules of the BGT-Model. In Section 3.1, the object communication step implemented with a BiGRU is explained in detail. Section 3.2, introduces the object classification and edge information generation step using transformers. In Section 3.3, the FS of the long-tailed relationship distribution and the BA as a pre-processing for the relationship classification is described.

3.1. Object Communication

The object communication module takes the visual features x_i as input. The communication between the objects is implemented by a BiGRU. Due to the architecture of BiGRU, information from every object can flow to every other object. This information flow can be regulated by the BiGRU by learning which information shall be passed and which information shall be blocked. The output of the object communication step is therefore given by:

$$\hat{O} = BiGRU([x_i]_{i=1,2,...,n}) \qquad (1)$$

Where $\hat{O} = \{\hat{o}_1, \hat{o}_2, ..., \hat{o}_n\}$ are the object features after the communication step. The $\hat{o}_i \in \mathbb{R}^{1028}$ is obtained by concatenating the outputs $\overrightarrow{o}_i$ (left to right in BiGRU) and $\overleftarrow{o}_i$ (right to left in BiGRU), such that:

$$\hat{o}_i = [\overrightarrow{o}_i, \overleftarrow{o}_i] \in \mathbb{R}^{1028} \qquad (2)$$

3.2. Object and Edge Transformers

The output $\hat{O}$ is projected to a 512-dimensional subspace to be fed into the Object transformer encoder. This transformer encoder follows the model architecture proposed by [35], and takes the encoder block of the complete transformer model. This transformer encoder is built up by a Multi-Head attention layer, an Add & Norm layer, a Feed Forward layer, and another Add & Norm layer.

The three inputs to the Multi-Head Attention layer are the values V, the keys K and the queries Q. According to [35], these three inputs are obtained from a single input. Using three different feed-forward fully-connected layers, yields the queries, keys, and values. For every attention head i (here i=1,2,...,8) these values are calculated by:

$$Q_i = \hat{O} * W_i^Q \qquad (3)$$

$$K_i = \hat{O} * W_i^K \qquad (4)$$

$$V_i = \hat{O} * W_i^V \qquad (5)$$

Where $W_i^Q \in \mathbb{R}^{512 \times d_k}$, $W_i^K \in \mathbb{R}^{512 \times d_k}$ and $W_i^V \in \mathbb{R}^{512 \times d_v}$ are learnable parameter matrices. Also, $d_k = d_v = 64$ are the same for this application. From these values, the Scaled Dot-Product attention is calculated, such that the output of each attention head is given by:

$$Z_i = softmax(\frac{Q_i * K_i^T}{\sqrt{d_k}}) * V_i \qquad (6)$$

Concatenating Z_i's give the output of the Multi-Head

Scaled Dot-Product layer as Z. This is then fed through another fully-connected layer to bring it back to the dimension of the input matrix $\hat{O}$.

$$Z = concatenate(Z_1, Z_2, ..., Z_{n_{heads}}) \quad (7)$$

The Add & Norm layer adds the input of the previous Multi-Head Attention layer as a residual connection to the output of the attention layer. The normalization applied is a normalization layer following the approach of [2]. Here, it is suggested that the 'covariate shift' problem can be reduced by changing the mean and the variance of the summed inputs in every layer. This is followed by a Feed Forward layer with two linear transformations and a ReLU activation followed by another Add & Norm layer. This transformer encoder block gets repeated 6 times. The output Z_6 of the last repetition is then used to predict the object labels:

$$O = softmax(W_o * Z_6') \quad (8)$$

Where $W_o \in \mathbb{R}^{512 \times 151}$ predicts the object class distribution for each detected object. Repeating this procedure but using the output Z_6 of the Object transformer as input to the Edge transformer yields feature vectors having information about the edges for every object. Similarly to [45], this information then can be used in the relationship prediction step. This edge information is given formally by:

$$E = TransformerEncoder(Z_6) \quad (9)$$

Where $E = \{e_1, e_2, ..., e_n\}$ contains the edge information for every object.

3.3. Frequency Softening(FS), Bias Adaptation(BA)

To handle the long-tailed relationship distribution present in the Visual Genome dataset, the procedure of softening this distribution and adapting the bias term for every subject-object pair form [21] is adopted. The used features in this step is different than in the GPS-Net [21], but the principle stays the same. The softening of the relationship distribution is done by applying a log-softmax function to the original distribution. The softened frequency distribution is therefore given by Eq. 10. The probability distribution $p_{i \to j}$ of every object pair i and j must be softened separately. Based on this definition, softening does not take any information of the respective visual scene into account.

$$\tilde{p}_{i \to j} = log \, softmax(p_{i \to j}) \quad (10)$$

BA is used to get a case-specific adaptation of the above term. BA allows us to adjust the bias in the relationship prediction step. This adaptation term takes the appearance of the subject-object pair i, j into account, by using their union feature $u_{i,j}$. The BA term d can be calculated by:

$$d = W_p * u_{i,j} \quad (11)$$

Where $W_p \in \mathbb{R}^{2048}$ is the transformation matrix and $u_{i,j}$ is the union feature of the subject-object pair. In the relationship prediction step, this bias term can be used as follows:

$$p_{i,j} = softmax(W_r(o_i' * o_j' * u_{i,j}) + d \odot \tilde{p}_{i \to j}) \quad (12)$$

In this equation, the bias $d \odot \tilde{p}_{i \to j}$ can be adjusted by changing d accordingly. Here, $o_i' = [o_i, e_i]$ (same for o_j') with '[-,-]' denotes the concatenation function representing the object features obtained in earlier steps. W_r is the classifier that projects the features to the relationship class dimension. $\odot$ denotes the Hadamard Product and $*$ the fusion function. The fusion function for $(x * y)$ is given by:

$$(x * y) = (W_x x + W_y y) - (W_x x - W_y y) \odot (W_x x - W_y y) \quad (13)$$

With the parameter matrices W_x and W_y, the fusion function is proposed to learn to count objects in images. [51]. The predicted relationship between objects i and j is given by:

$$r_{i,j} = argmax(p_{i,j}(r)) \quad (14)$$

Where r lies in the set of relationship classes including background BG.

4. Experiments

We performed experiments using three different datasets, i.e., Visual Genome (VG) [13], Visual Relationship Detection (VRD) [22], and OpenImages (OI) [14].

4.1. Visual Genome

Visual Genome dataset [13] is the most frequently used dataset for the SGG task. We use the same data statistics and evaluation metrics as widely used by the state of the art in this field, i.e., 150 object categories and 50 relationship categories are used. 70% of the dataset is used for training and 30% for testing. An additional 5000 images are taken from the training set and are used for validation. As used by the state of the art, we also employed Faster R-CNN [25] with VGG-16 or ResNext-101 as a backbone to get the characteristics of object proposals. To keep the fairness in the comparison with state-of-the-art, we chose same experimental factors as chosen by [21].

Performance Diagnosis: The scene graph generation model is evaluated in three different sub-tasks: (1) Predicate classification, (2) Scene graph classification, and (3) Scene graph generation. These are the three protocols for which the model's performance is evaluated separately. Predicate Detection is used to specify the relation of given objects. This protocol evaluates the set of possible relations between a pair of given objects. The prediction of relationships without the effect of object detection is examined. In Phrase Detection, the input is an image and the outputs are triplets of subject-predicate-object. Additionally, one bounding box must have an overlap of at least 0.5 with the corresponding ground truth. For Relation Detection, the same input and output as in Phrase Detection is used. In this case, not only one but two bounding boxes of the pair of objects must have at least 0.5 overlap with the ground truth.

Metrics. The evaluated metrics for the diagnosis of the model performance is Recall at K (R@K), no graph constraint Recall at K (nGR@K), zero-shot Recall at K (zsR@K), and mean Recall (mR@K), where K=20, 50, and 100, respectively.

Object Detector: A pretrained Faster R-CNN object detector with a VGG-16 net [9] or ResNeXt-101-FPN [36] as a backbone is used which is taken from [21]. This detector was trained on the VG dataset, with batch size 8 and initial learning rate $8 * 10^{-3}$ which is decayed at the $30k^{th}$ and $40k^{th}$ iteration by the factor of 10. After training this detector on 4 2080Ti GPU, 28.14 mAP (with 0.5 IoU) was achieved.

Scene Graph Generation: The scene graph generation is trained with an SGD optimizer [3]. The learning rate is set at 10^{-3} for all three protocols. This learning rate was decayed by a factor of 10 twice after hitting a validation performance plateau. Per-Class non-maximal suppression (NMS) was applied with 0.5 IoU. 160 RoIs for each image were sampled. In contrast to previous works, we also considered non-overlapping regions for relationship prediction. To generalize the scene graph generation task, we used similar settings as used in literature [21]. For model training, an RTX Titan was used. The batch size was set to 12 and the learning rate started at 10^{-3} and was reduced two times by a factor of 10 after hitting a validation performance plateau. The number of solver iterations was set to 18000.

4.2. OpenImages

The training and validation sets of the OpenImages dataset contain 53,953 and 3,234 images. For comparison, we use the same Faster R-CNN detector with a pre-trained ResNeXt-101-FPN backbone as used by [21, 50]. Also, the same data processing and evaluation metrics are used as in these previous works [21, 50]. The evaluation metrics are Recall@50, weighted mean average precision (AP) of relationships wmAP$_{rel}$, and weighted mean AP of phrase wmAP$_{phr}$. The final score is given by $0.2 * R@50 + 0.4 * wmAP_{rel} + 0.4 * wmAP_{phr}$, which was adopted from the OpenImages challenge formula [49], where the mAP was replaced by its weighted counterpart. The replacement of the mAP with the wmAP was done [49] due to the extreme predicate class imbalance. The wmAP is achieved by scaling each predicate category by their relative ratios in the val set from the mAP. Important to note is that the wmAP$_{rel}$ evaluates the AP of the predicted triplet where both the subject and object boxes have an IoU of at least 0.5 with ground truth. The wmAP$_{phr}$ is quite similar but is utilized for the union box of the subject and the object.

4.3. Visual Relation Detection

The Visual Relation Detection (VRD) dataset was introduced by [22]. We adopt the same detectors as [49]. Specifically, we use a pre-trained Faster R-CNN detector with VGG-16 backbone trained on the COCO dataset [20]. The

evaluation process is given by [22] and the metrics used are R@50 and R@100.

4.4. Implementation Details

We follow the same implementation parameters as used by [21]. To ensure a fair comparison with state of the art, we used VGG-16 and ResNext-101 as backbone. We use the 10^{-3} as the learning rate and 6 as the batch-size which is the same as used by [21]. We use SGD with momentum as the optimizer for the training process. We use the relationship between overlapped bounding boxes and subject-object pairs for the SGDet process. NMS with an IoU of 0.3 is used and the topmost 64 object proposals are chosen. The ratio of 3:1 is maintained during training between the subject-object pairs with and without any relationship.

4.5. Comparisons with State-of-the-Art Methods

Visual Genome: BGT-Net outperforms all the compared previous literature work as shown in Table 1 on R@K for all values of K . BGT-Net performs better than a recent model named GPS-net [21] by 6% and by 12% on average at R@50 and R@100 over the three protocols when VGG-19 or ResNext-101 is used as the base for the Faster-RCNN, respectively. It also outperforms when an individual evaluation protocol is compared. The improvement is by 16.6% for SGDet, 17.3% for SGCls, and for 0.2% for PredCls. When compared to the classic MOTIFS [45], it showed an improvement of 7.3% and 14% on average at R@50 and R@100 over the three protocols when VGG-19 or ResNext-101 is used as a backbone for the Faster-RCNN, respectively. BGT-Net outperforms FREQ [45], VCTREE-SL [33], VCTREE-HL [33], GB-NET [44] , NODIS [43], CMAT [4], KERN [5], Graph R-CNN [39], IMP [37] by 14.9%, 4.3%, 3.9%, 5.6%, 3.2%, 3.2%, 6.3%, 40.9%, 85.4%, respectively, on average at R@50 and R@100 over the three protocols when VGG-19 is used as a backbone for the Faster-RCNN and by 22.2%, 11%, 10.5%, 12.3%, 9.8%, 9.8%, 13.1%, 49.9%, 97.3%, respectively, on average at R@50 and R@100 over the three protocols when ResNext-101 is used as a backbone for the Faster-RCNN.

We evaluate Mean Recall of BGT-Net to understand its performance on the class imbalance problem. So, we study its performance by conducting experiments to calculate its Mean Recall [5, 33, 21]. We see in Figure 3 and Table 2 that BGT-Net performs well considering the Mean Recall evaluation metric. The mean of the Mean Recall over all the three evaluation metrics (SGDet, SGCls, PredCls) is 15.5 for BGT-Net and hence outperforms GB-Net and GPS-Net which are the best state-of-the-art on class imbalance handling having good performance on both mean R@K and R@K results. This gives a positive indication that BGT-Net can tackle the problem of class imbalance while simultaneously giving high R@K as compared to the other existing

	SGdet			SGCls			PredCls			
Model	*R@20*	*R@50*	*R@100*	*R@20*	*R@50*	*R@100*	*R@20*	*R@50*	*R@100*	Mean
MOTIFS ⋆ [45]	21.4	27.2	30.3	32.9	35.8	36.5	58.5	65.2	67.1	43.7
FREQ ⋆ [45]	20.1	26.2	30.1	29.3	32.3	32.9	53.6	60.6	62.2	40.7
VCTREE-SL ⋆ [33]	21.7	27.7	31.1	35.0	37.9	38.6	59.8	66.2	67.9	44.9
VCTREE-HL ⋆ [33]	22.0	27.9	31.3	35.2	38.1	38.8	60.1	66.4	68.1	45.1
GB Net ⋆ [44]	-	26.3	29.9	-	37.3	38	-	66.6	68.2	44.4
NODIS ⋆ [43]	21.5	27.4	30.7	36	39.8	40.7	58.9	66	67.9	45.4
GPS-Net ⋆ [21]	22.6	28.4	31.7	36.1	39.2	40.1	60.7	66.9	68.8	45.9
CMAT ⋆ [4]	22.1	27.9	31.2	35.9	39	39.8	60.2	66.4	68.1	45.4
KERN ⋆ [5]	-	27.1	29.8	-	36.7	37.4	-	65.8	67.6	44.1
Graph R-CNN ⋆ [39]	-	11.4	13.7	-	29.6	31.6	-	54.2	59.1	33.3
IMP ⋆ [37]	-	3.44	4 .24	-	21.72	24.38	-	44.75	53.08	25.3
BGT-Net (no BiGRU) ⋆	**23.61**	**30.4**	**34.81**	**33.81**	**37.22**	**38.12**	**57.98**	**64.75**	**66.63**	**46.9**
BGT-Net ⋆	**23.1**	**28.6**	**32.2**	**38.0**	**40.9**	**43.2**	**60.9**	**67.3**	**68.9**	**46.9**
BGT-Net ⋄	**25.5**	**32.8**	**37.3**	**41.7**	**45.9**	**47.1**	**60.9**	**67.1**	**68.9**	**49.9**

Table 1: Recall@K Model comparison with state-of-the-arts on the VG dataset. We compare R@20, R@50, and R@100. For some literature work, the R@20 is not given. We used '−' which denotes that the result is unavailable. So, the mean is calculated using values of R@50 and R@100 to have fair comparison with state of the art. Models using the same VGG backbone are denoted with '⋆' and the BGT-Net with ResNext-101 background is marked with '⋄'.

	SGdet	SGCls	PredCls	
Model	*mR@100*	*mR@100*	*mR@100*	Mean
GB Net ⋆ [44]	8.5	13.4	**24**	15.3
IMP ⋆ [37]	4.8	6.0	10.5	7.1
GPS-Net ⋆ [21]	**9.8**	12.6	22.8	15.1
VCTREE-HL ⋆ [33]	8.0	10.8	19.4	12.8
MOTIFS ⋆ [45]	6.6	8.2	15.3	10.0
KERN ⋆ [5]	7.3	10	19.2	12.2
BGT-Net ⋆	9.6	**13.7**	23.2	**15.5**

Table 2: Comparison on mR@100 between various methods across all 50 relationship categories.

solutions to best of our knowledge.

Why BiGRU? We investigated BGT-Net without BiGRU but BGT-Net (no BiGRU) has lower SGCls and PredCls results (see Table 1). The deciding factor for using BiGRU in BGT-Net is mR@K and nGR@K. Both these metrics significantly improve for all three: SGDet, SGCls, PredCls when BGT-Net is with BiGRU as shown in Table 3.

Also shown in Table 3, BGT-Net has high improvement on zsR@K as compared to [37] and [45] which shows that it is able to better detect those subject-predicate-object combinations which are not present in the training set.

OpenImages: The results in Table 5 show that BGT-Net performs very well on the OpenImages dataset. The overall score is 0.64 points higher than the GPS-Net model performance. This increase in performance is achieved by an overall increase in performance in all three evaluated metrics R@K, wmAP$_{rel}$, and wmAP$_{phr}$. An evaluation of the per-class AP is also shown. In this evaluation, some classes were chosen as in GPS-Net [21] and RelDN [49] to show class-specific performance. The performances of GPS-Net and the BGT-Net are quite close. For "holds", "interacts with" and "wears", the GPS-Net shows a higher AP while for the others the BGT-Net shows the highest AP. The overall performance of the BGT-Net outperforms the state-of-the-art performance of the GPS-Net model.

Visual Relationship Detection: The evaluation results on VRD Dataset are illustrated in Table 4. The BGT-Net uses

Figure 3: The increase in R@100 in PredCls of BGT-Net compared with the MOTIFS [45] and VCTREE [33]. The Top-35 relationship categories are selected according to their alphabetic occurrence.

the same Detector as RelDN and GPS-Net. The BGT-Net outperforms the state-of-the-art models in all three evaluation metrics. So, BGT-Net shows the best to the date performance on the VRD dataset as well to the best of our knowledge, and which is better than GPS-Net and RelDN (previous best performing networks).

4.6. Ablation Studies

To evaluate and analyze our proposed BGT-Net, we conducted a number of ablations as shown in Table 6, Table 7, and Table 8.

Network Performance with a different combination of Modules. In this study, we evaluate the effectiveness of the network in the presence of the three modules, i.e., Transformer, BiGRU, and FS, individually and collectively. The performance of the network increases with the presence of

	SGdet			SGCls			PredCls		
Model	$nGR@20$	$nGR@50$	$nGR@100$	$nGR@20$	$nGR@50$	$nGR@100$	$nGR@20$	$nGR@50$	$nGR@100$
GB Net ⋆ [44]	-	29.3	35	-	46.9	50.3	-	**83.5**	**90.3**
CMAT ⋆ [4]	23.7	31.6	36.8	**41**	**48.6**	**52**	**68.9**	83.2	90.1
KERN ⋆ [5]	-	30.9	35.8	-	45.9	49	-	65.8	67.6
BGT-Net (no BiGRU) ⋄	24.23	32.88	39.06	38.55	46.28	50.04	65.92	80.51	87.82
BGT-Net ⋄	**27.24**	**36.91**	**43.72**	**47.83**	**57.67**	**62.29**	**69.1**	**83.71**	**90.55**
Model	$mR@20$	$mR@50$	$mR@100$	$mR@20$	$mR@50$	$mR@100$	$mR@20$	$mR@50$	$mR@100$
GB Net ⋆ [44]	-	7.1	8.5	-	12.7	13.4	-	**22.1**	**24**
GPS-Net⋆[21]	-	-	**9.8**	-	-	12.6	-	-	22.8
KERN ⋆[5]	-	6.4	7.3	-	9.4	10	-	17.7	19.2
BGT-Net (no BiGRU) ⋄	4.62	6.55	7.85	7.31	9.14	9.71	12.14	15.59	17.05
BGT-Net ⋄	**5.69**	**7.81**	9.25	**10.41**	**12.77**	**13.61**	**16.8**	**20.56**	**22.98**
Model	$zsR@20$	$zsR@50$	$zsR@100$	$zsR@20$	$zsR@50$	$zsR@100$	$zsR@20$	$zsR@50$	$zsR@100$
Motifs ⋆[31]	0	0.05	0.11	0.32	0.91	1.39	1.35	3.63	5.36
IMP ⋆[37]	0.18	0.38	0.77	2.01	3.03	3.92	12.17	17.66	20.25
BGT-Net ⋄	**1.22**	**2.38**	**3.42**	**4.8**	**7.37**	**8.78**	**12.23**	**18.31**	**21.51**

Table 3: nGR@K, mR@K and zsR@K comparison with state-of-the-art on the VG dataset. The version of the BGT-Net, i.e., the BGT-Net (no BiGRU) is also compared.

	Predicate Detection	Relation Detection		Phrase Detection		
Model	$R@50$	$R@50$	$R@100$	$R@50$	$R@100$	Mean
VTransE [48]	44.8	19.4	22.4	14.1	15.2	23.2
ViP-CNN [11]	-	17.3	20.0	22.8	27.9	22
VRL [19]	-	18.2	20.8	21.4	22.6	20.8
KL distilation [42]	55.2	19.2	21.3	23.1	24.0	28.6
MF-URLN [46]	58.2	23.9	26.8	31.5	36.1	35.3
Zoom-Net ⋄ [41]	50.7	18.9	21.4	24.8	28.1	28.8
CAI + SCA-M ⋄ [41]	56.0	19.5	22.4	25.2	28.9	30.4
RelDN ⋆ [49]	-	25.3	28.6	31.3	36.4	30.4
GPS-Net ⋆ [21]	63.4	27.8	31.7	33.8	39.2	39.2
BGT-Net ⋆	**64.1**	**28.5**	**31.9**	**34.4**	**39.4**	**39.6**

Table 4: Comparison on the VRD dataset [22]. '⋆' and '⋄' denote the models using the same object detector. The object detector with a VGG-16 backbone trained on COCO is used as in RelDN and GPS-Net to have fair comparison.

all three modules. Firstly, we evaluated for an individual module and then permutated the modules with each other to make a combination to evaluate the performance of different resulting configurations. As illustrated in Table 6, for all the three evaluation protocols, i.e., SGDet, SGCls, and PredCls, our proposed network with all the three modules outperforms the other network configurations with individual modules. When modules are used together, the network performance improves which shows that each individual module plays a significant role in predicting objects and their relationships.

The FS and BA were adapted from [20]. In [21], they performed ablation study with and without FS and BA. It was shown that these modules improve the PredCls for all three R@20, R@50, and R@100. We included in our ablation the effect of using FS which showed that zsR@20, zs@50, and zsR@100 improved drastically for all three: SGDet, SGCls, PredCl, when using FS as shown in Table 6.

Performance with different number of Transformer Heads. We performed this ablation to validate the optimized number of transformer heads in the network. As illustrated in Table 7, we can see that when 1, 2, or 6 number of transformer heads are used, the network with 6 transformer heads performs better in all the experiments than others for all the three evaluation protocols. This ablation shows that the number of transformer heads also effect the performance of the model and hence this factor is critical

while designing the network for the SGG. This study motivated us to use six transformer heads in the novel BGT-Net.

Performance with different number of Bidirectional GRU Layers. We compare networks with different number of BiGRU layers. To keep the comparison fair, we use all other same parameters in the experiments except the number of BiGRU layers. As shown in Table 7, it is evident that increasing the number of BiGRU layers does not significantly improve the performance, but it does increase the computational power and training time. Hence, in the BGT-Net, we only use one BiGRU layer.

Qualitative Results. In Figure 4, Left: shows the qualitative results. In SGCls, the bounding boxes are given and the model has to predict the object class and the relationships. In SGDet, no information is given. In SGDet up to 160 objects in an image can be detected but to keep the illustrations clean not every object detection is shown. It detected relationships present in the ground truth along with the additional feasible relationships. Figure 4 Top: in SGCls the predicted scene graph fully corresponds to the ground truth scene graph for this image. There is no additional relationship predicted between other objects. Looking at the scene graph for SGDet, the difference between these two protocols can be seen very well. Also, the performance of the model is really good in this case. Additionally to the ground truth objects, the object "motorcycle" and "person" are detected. These two detections are correct and feasible. While the only ground truth relation (woman - wearing - shirt) is still being detected, three other relationships that are totally feasible are detected (woman - on - motorcycle), (person - on - motorcycle) and (wheel - on - motorcycle). The performance of the BGT-Net on this image is outstanding. No problems or specialties in the SGDet can be found.

In Figure 4 Right: In SGDet, the model even fails to give the correct relationship (car - on - track). But, it correctly detects the whole train, which was not specified in the ground truth and the correct relationship (train - on - track). It is special in this case and it might also be in a lot of other

Model	R@50	$wmAP_{rel}$	$wmAP_{phr}$	$score_{wtd}$	AP$_{rel}$ per class								
					at	on	holds	plays	interacts with	wears	hits	inside of	under
RelDN, L_0 [49]	74.67	34.63	37.89	43.94	32.40	36.51	41.84	36.04	40.43	5.70	55.40	44.17	25.00
RelDN [49]	74.94	35.54	38.52	44.61	32.90	37.00	43.09	41.04	44.16	7.83	51.04	44.72	50.00
GPS-Net [21]	77.29	38.78	40.15	47.03	35.10	38.90	**51.47**	45.66	**44.58**	**32.35**	71.71	47.21	57.28
BGT-Net	**77.98**	**39.56**	**40.75**	**47.67**	**36.23**	**39.05**	50.96	**46.78**	44.56	31.45	**72.17**	**48.03**	**57.64**

Table 5: Comparison on OpenImages dataset [15]. The BGT-Net uses a ResNext-101 backbone. Additionally, the same data processing and evaluation metrics as [21, 49] are followed to a ensure fair comparison.

	Transformer	GRU	FS	R @ 20	R @ 50	R @ 100	nG R @ 20	nG R @ 50	nG R @ 100	zs R @ 20	zs R @ 50	zs R @ 100	mR @ 20	mR @ 50	m R @ 100
	x	-	-	23.61	30.4	34.81	24.23	32.88	39.06	0	0	0	4.62	6.55	7.85
SGDet	-	x	-	24.1	31.2	35.5	25.37	34.59	41.14	0	0	0.03	4.07	5.49	6.51
	-	-	x	22.3	28.17	32.56	25.04	34.58	41.23	0.95	1.27	2.21	4.47	5.98	7.65
	x	x	x	24.68	31.87	36.18	26.23	35.87	42.46	1.22	2.38	3.42	5.69	7.81	9.25
	x	-	-	33.81	37.22	38.12	38.55	46.28	50.04	0.15	0.45	0.7	7.31	9.14	9.71
SGCls	-	x	-	40.03	44	45.02	45.61	54.82	59.26	0.19	0.69	0.99	7.92	9.85	10.55
	-	-	x	35.63	38.92	39.77	39.41	47.92	55.56	3.25	4.99	5.78	8.2	10.65	11.34
	x	x	x	41.72	45.69	46.74	47.96	57.42	61.92	4.12	6.72	8.06	10.41	12.77	13.61
	x	-	-	57.98	64.75	66.63	65.92	80.51	87.82	0.64	2.06	3.69	12.14	15.59	17.05
PredCls	-	x	-	58.52	65.19	67.08	65.76	80.38	87.83	0.68	2.52	4.46	12.33	15.79	17.16
	-	-	x	56.73	63.48	65.53	64.87	79.56	87.2	11.9	17.78	20.97	12.05	15.22	16.46
	x	x	x	58.71	65.25	67.1	67.27	82.05	89.29	12.06	18.22	21.49	14.36	17.88	19.44

Table 6: Ablation study performed on evaluation of three models. All the experimental factors are kept same except the ones which are being evaluated to have fair comparison. The evaluation is conducted using Recall@K, nGR@K, zsR@K, and mR@K.

	Transformer Heads	R @ 20	R @ 50	R @ 100	nG R @ 20	nG R @ 50	nG R @ 100	zs R @ 20	zs Recall @ 50	zs R @ 100	mR @ 20	mR @ 50	m R @ 100
	1	23.78	31.15	35.4	25.76	34.8	41.78	0.99	2.11	2.67	4.75	7.32	7.96
SGDet	2	24.54	31.77	36.11	26.05	35.64	42.39	1.07	2.18	3.37	5.57	7.52	8.8
	6	**24.68**	**31.87**	**36.18**	**26.23**	**35.87**	**42.46**	**1.22**	**2.38**	**3.42**	**5.69**	**7.81**	**9.25**
	1	38.62	42.54	43.66	45.01	52.1	58.78	2.67	4.31	4.89	8.21	9.98	10.32
SGCls	2	39.37	43.42	44.51	45.25	54.51	59.17	3.32	5.37	6.48	9.01	11.14	11.89
	6	**41.72**	**45.69**	**46.74**	**47.96**	**57.42**	**61.92**	**4.12**	**6.72**	**8.06**	**10.41**	**12.77**	**13.61**
	1	58.03	64.8	66.89	66.23	81.06	87.89	11.06	17.39	19.45	12.55	16.21	17.51
PredCls	2	58.45	64.98	67.03	66.89	81.66	88.34	11.9	18.11	21.34	12.91	16.75	18.92
	6	**58.71**	**65.25**	**67.1**	**67.27**	**82.05**	**89.29**	**12.06**	**18.22**	**21.49**	**14.36**	**17.88**	**19.44**

Table 7: Various R@K performance for the different numbers of Transformer Heads in BGT-Net using VG dataset.

	Bi-GRU	R @ 20	R @ 50	R @ 100	nG R @ 20	nG R @ 50	nG R @ 100	zs R @ 20	zs Recall @ 50	zs R @ 100	mR @ 20	mR @ 50	m R @ 100
	1	**25.54**	**32.87**	**37.3**	**27.24**	**36.91**	**43.72**	**4.8**	**7.37**	**8.78**	**10.41**	**12.77**	**13.61**
SGDet	2	24.54	31.77	36.11	26.05	35.64	42.39	1.07	2.18	3.37	9.91	12.28	13.12
	6	23.93	31.01	35.37	25.47	35.02	41.63	1.12	2.11	3.1	5.49	7.53	8.86
	1	41.69	**45.96**	**47.06**	47.83	**57.67**	**62.29**	2.67	4.31	4.89	8.21	9.98	10.32
SGCls	2	**41.72**	45.69	46.74	**47.96**	57.42	61.92	**4.12**	**6.72**	**8.06**	**9.01**	**11.14**	**11.89**
	6	41.08	45.57	46.23	46.98	57.12	61.3	4.02	6.56	7.79	8.54	11.81	13.01
	1	**59.21**	**65.68**	**67.45**	**67.69**	**82.42**	**89.45**	**12.23**	**18.31**	**21.51**	**14.7**	**18.46**	**20.08**
PredCls	2	58.71	65.25	67.1	67.27	82.05	89.29	12.06	18.22	21.49	14.36	17.88	19.44
	6	58.22	65.45	66.74	66.91	81.87	88.69	11.83	18.07	21.1	14.22	17.59	19.21

Table 8: Various R@K performance comparison in relation to the number of BiGRU layers present in network when trained on VG dataset.

Figure 4: Qualitative results showing scene graphs generated by the BGT-Net. For both examples, (a) shows the scene graph generated in the SGDet protocol, and (b) the one generated in SGCls. The green arrow denotes that the detection object or relationship corresponding to the ground-truth. Orange arrows denote the detections that are not available in ground-truth but do represent the image properly. Red marks errors that are used for undetected relationships or wrongly detected objects.

images that the model detects many objects that were not shown in the ground truth. It also shows a lot of relationships between these additionally shown objects. But, most likely with a higher amount of detected objects in an image, these triplets, that are not in the ground truth, leads the model to miss some of the relationships that would also be found in the ground truth.

5. Conclusion

We proposed a novel method BGT-Net to address the main challenges in SGG. BGT-Net solves the problems by 1) using the object-object communication by employing Bi-directional GRUs; 2) using transformer encoder with scaled-dot-product attention for predicting object classes after they have received feature information from other objects; 3) getting edge feature from second transformer encoder; 4) Utilising Frequency Softening and Bias Adaptation for dealing the with bias in the SGG. We validated the effectiveness of the proposed BGT-Net using extensive experiments and conducting elaborative ablation studies using three open-source datasets.

References

[1] Jose M Alvarez, Theo Gevers, Yann LeCun, and Antonio M Lopez. Road scene segmentation from a single image. In *European Conference on Computer Vision*, pages 376–389. Springer, 2012.

[2] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization, 2016.

[3] Léon Bottou. Stochastic gradient descent tricks. In *Neural networks: Tricks of the trade*, pages 421–436. Springer, 2012.

[4] Long Chen, Hanwang Zhang, Jun Xiao, Xiangnan He, Shiliang Pu, and Shih-Fu Chang. Counterfactual critic multi-agent training for scene graph generation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4613–4623, 2019.

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

[6] Ciarán O Conaire, Noel E O'Connor, and Alan F Smeaton. An improved spatiogram similarity measure for robust object localisation. In *2007 IEEE International Conference on Acoustics, Speech and Signal Processing-ICASSP'07*, volume 1, pages I–1069. IEEE, 2007.

[7] Martin C Cooper. The tractability of segmentation and scene analysis. *International Journal of Computer Vision*, 30(1):27–42, 1998.

[8] Shalini Ghosh, Giedrius Burachas, Arijit Ray, and Avi Ziskind. Generating natural language explanations for visual question answering using scene graphs and visual attention. *arXiv preprint arXiv:1902.05715*, 2019.

[9] Song Han, Jeff Pool, John Tran, and William Dally. Learning both weights and connections for efficient neural network. In *Advances in neural information processing systems*, pages 1135–1143, 2015.

[10] Roei Herzig, Moshiko Raboh, Gal Chechik, Jonathan Berant, and Amir Globerson. Mapping images to scene graphs with permutation-invariant structured prediction. In *Advances in Neural Information Processing Systems*, pages 7211–7221, 2018.

[11] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9:1735–80, 12 1997.

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

[13] 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 Fei-Fei Li. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *CoRR*, abs/1602.07332, 2016.

[14] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper R. R. Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Tom Duerig, and Vittorio Ferrari. The open images dataset V4: unified image classification, object detection, and visual relationship detection at scale. *CoRR*, abs/1811.00982, 2018.

[15] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper R. R. Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Tom Duerig, and Vittorio Ferrari. The open images dataset V4: unified image classification, object detection, and visual relationship detection at scale. *CoRR*, abs/1811.00982, 2018.

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

[17] 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.

[18] 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.

[19] Xiaodan Liang, Lisa Lee, and Eric P. Xing. Deep variation-structured reinforcement learning for visual relationship and attribute detection. *CoRR*, abs/1703.03054, 2017.

[20] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[21] Xin Lin, Changxing Ding, Jinquan Zeng, and Dacheng Tao. Gps-net: Graph property sensing network for scene graph generation, 2020.

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

[23] Constantine Papageorgiou and Tomaso Poggio. A trainable system for object detection. *International journal of computer vision*, 38(1):15–33, 2000.

[24] 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 Conference on Computer Vision and Pattern Recognition*, pages 3957–3966, 2019.

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

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

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

[28] Jiaxin Shi, Hanwang Zhang, and Juanzi Li. Explainable and explicit visual reasoning over scene graphs. In *Proceedings*

of the IEEE Conference on Computer Vision and Pattern Recognition, pages 8376–8384, 2019.

[29] Josephine Sullivan, Andrew Blake, Michael Isard, and John MacCormick. Bayesian object localisation in images. *International Journal of Computer Vision*, 44(2):111–135, 2001.

[30] Christian Szegedy, Alexander Toshev, and Dumitru Erhan. Deep neural networks for object detection. In *Advances in neural information processing systems*, pages 2553–2561, 2013.

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

[32] Kaihua Tang, Yulei Niu, Jianqiang Huang, Jiaxin Shi, and Hanwang Zhang. Unbiased scene graph generation from biased training. *arXiv preprint arXiv:2002.11949*, 2020.

[33] Kaihua Tang, Hanwang Zhang, Baoyuan Wu, Wenhan Luo, and Wei Liu. Learning to compose dynamic tree structures for visual contexts. *CoRR*, abs/1812.01880, 2018.

[34] 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.

[35] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *CoRR*, abs/1706.03762, 2017.

[36] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1492–1500, 2017.

[37] Danfei Xu, Yuke Zhu, Christopher B. Choy, and Li Fei-Fei. Scene graph generation by iterative message passing. *CoRR*, abs/1701.02426, 2017.

[38] 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.

[39] 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.

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

[41] 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. *CoRR*, abs/1807.04979, 2018.

[42] Ruichi Yu, Ang Li, Vlad I. Morariu, and Larry S. Davis. Visual relationship detection with internal and external linguistic knowledge distillation. *CoRR*, abs/1707.09423, 2017.

[43] Cong Yuren, Hanno Ackermann, Wentong Liao, Michael Ying Yang, and Bodo Rosenhahn. Nodis: Neural ordinary differential scene understanding. *arXiv preprint arXiv:2001.04735*, 2020.

[44] Alireza Zareian, Svebor Karaman, and Shih-Fu Chang. Bridging knowledge graphs to generate scene graphs. In *Proceedings of the European conference on computer vision (ECCV)*, August 2020.

[45] 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.

[46] Yibing Zhan, Jun Yu, Ting Yu, and Dacheng Tao. On exploring undetermined relationships for visual relationship detection. *CoRR*, abs/1905.01595, 2019.

[47] Cheng Zhang, Wei-Lun Chao, and Dong Xuan. An empirical study on leveraging scene graphs for visual question answering, 2019.

[48] Hanwang Zhang, Zawlin Kyaw, Shih-Fu Chang, and Tat-Seng Chua. Visual translation embedding network for visual relation detection. *CoRR*, abs/1702.08319, 2017.

[49] Ji Zhang, Kevin J. Shih, Ahmed Elgammal, Andrew Tao, and Bryan Catanzaro. Graphical contrastive losses for scene graph generation. *CoRR*, abs/1903.02728, 2019.

[50] Ji Zhang, Kevin J Shih, Ahmed Elgammal, Andrew Tao, and Bryan Catanzaro. Graphical contrastive losses for scene graph parsing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 11535–11543, 2019.

[51] Yan Zhang, Jonathon S. Hare, and Adam Prügel-Bennett. Learning to count objects in natural images for visual question answering. *CoRR*, abs/1802.05766, 2018.