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

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

In this supplementary material, we provide additional results and details of our method, Selective Quad Attention Networks (SQUAT).

1. Implementation details

1.1. Code base and GPUs.

We implemented SQUAT using Pytorch [17] and some of the official code-base for BGNN [10][1]. SQUAT was trained for ∼8 hours on 4 RTX 3090 GPUs with batch size 12.

1.2. Edge selection module.

Following [19], we use simple MLP with 4 linear layers and Layer Normalization [1] with GeLU [7] activation. To capture the global statistics of the edge features $\mathcal{E} = \{f_{ij}\}_{i,j}$, we average half of the output dimensions of the first layer as a global feature g:

$$[h^l_{ij}; h^g_{ij}] = l^1(f_{ij}) \tag{1}$$

$$g = \frac{1}{|\mathcal{E}|}\sum_i \sum_j h^g_{ij}, \tag{2}$$

where l^1 is the first layer of the edge selection module and $[\cdot;\cdot]$ is the concatenation operation. The dimensions of the local part h^l_{ij} and the global part h^g_{ij} are the same. We concatenate the global feature g with each of the remaining local parts h^l_{ij} and pass into the remaining 3-layer MLP to calculate the relatedness scores s_{ij}:

$$s_{ij} = l^2([h^l_{ij}; g]), \tag{3}$$

where l^2 is the remaining 3-layer MLP. In order to remove the invalid edges, we choose top-ρ% highest relatedness score pairs $\mathcal{E}^\rho$ as the valid edges.

1.3. Training details.

To train SQUAT, we use Stochastic Gradient Descent (SGD) optimizer with a learning rate 10^{-3}. In the early stages of training, notice that the edge selection model is too naive to select the valid edges to construct feasible scene graphs and therefore causes instability during training. To make the training stable, we pre-trained the edge selection module for 2000 iterations with a learning rate of 10^{-4} freezing all other parameters, and then we trained the entire SQUAT without the node detection module.

We use the keeping ratio $\rho = 0.7$ and $\rho = 0.35$ in training time and inference time, respectively, for all the SGDet settings on the Visual Genome and the Open Images v6 datasets. Also, we use the keeping ratio $\rho = 0.9$ for the SGCls and the PredCls settings on Visual Genome. Since the background proposals do not exist in the SGCls and the PredCls settings, there are fewer invalid edges than in the SGDet setting; thus, we use a smaller keeping ratio. We use three quad attention layers for the SGDet setting and two quad attention layers for the SGCls and the PredCls settings.

2. Additional evaluations on Visual Genome

2.1. Trade-off between recall and mean recall

Since the Visual Genome dataset[2] has extremely long-tailed distribution, there is the trade-off between recall and mean recall [15, 21]. To evaluate various trade-offs of the scene graph generation methods, Zhang *et al*. [29] propose the F@K measure, the harmonic mean of recall and mean-recall, recently. Table 1 shows the R@50/100, mR@50/100, and F@50/100 on the Visual Genome dataset. SQUAT outperforms all of the state-of-the-art methods at F@50/100 measurements. It shows that although the recall of SQUAT degrades, the trade-off between the recall and the mean recall is the best in the state-of-the-art methods.

[1] https://github.com/SHTUPLUS/PySGG

[2] 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.

Methods	PredCls			SGCls			SGDet		
	R@50 / 100	mR@50/100	F@50 / 100	R@50 / 100	mR@50/100	F@50 / 100	R@50 / 100	mR@50/100	F@50 / 100
IMP+‡ [11]	61.1 / 63.1	11.0 / 11.8	18.6 / 19.9	37.5 / 38.5	6.2 / 6.5	10.6 / 11.1	25.9 / 31.2	4.2 / 5.2	7.2 / 8.9
Motifs‡ [28]	66.0 / 67.9	14.6 / 15.8	23.9 / 25.6	39.1 / 39.9	8.0 / 8.5	13.3 / 14.0	32.1 / 36.9	5.5 / 6.8	9.4 / 11.5
Motifs‡† [28]	64.6 / 66.7	18.5 / 20.0	28.8 / 30.8	37.9 / 38.8	11.1 / 11.8	17.2 / 18.1	30.5 / 35.4	8.2 / 9.7	12.9 / 15.2
RelDN [30]	64.8 / 66.7	15.8 / 17.2	25.4 / 27.3	38.1 / 39.3	9.3 / 9.6	15.0 / 15.4	31.4 / 35.9	6.0 / 7.3	7.2 / 8.9
VCTree‡ [22]	65.5 / 67.4	15.4 / 16.6	24.9 / 26.6	38.9 / 39.8	7.4 / 7.9	12.4 / 13.2	31.8 / 36.1	6.6 / 7.7	10.9 / 12.7
MSDN [11]	64.6 / 66.6	15.9 / 17.5	25.5 / 27.7	38.4 / 39.8	9.3 / 9.7	15.0 / 15.6	31.9 / 36.6	6.1 / 7.2	10.2 / 12.0
GPS-Net [12]	65.2 / 67.1	15.2 / 16.6	24.7 / 26.6	39.2 / 37.8	8.5 / 9.1	14.0 / 14.7	31.1 / 35.9	6.7 / 8.6	11.0 / 13.9
RU-Net [14]	67.7 / 69.6	- / 24.2	- / 35.9	42.4 / 43.3	- / 14.6	- / 21.8	32.9 / 37.5	- / 10.8	- / 16.8
HL-Net [13]	60.7 / 67.0	- / 22.8	- / 34.0	42.6 / 43.5	- / 13.5	- / 20.6	**33.7 / 38.1**	- / 9.2	- / 14.8
VCTree-TDE [21]	47.2 / 51.6	25.4 / 28.7	33.0 / 36.9	25.4 / 27.9	12.2 / 14.0	16.5 / 18.6	19.4 / 23.2	9.3 / 11.1	12.6 / 15.0
Seq2Seq [15]	66.4 / 68.5	26.1 / 30.5	37.5 / 42.2	38.3 / 39.0	14.7 / 16.2	21.2 / 22.9	30.9 / 34.4	9.6 / 12.1	14.6 / 17.9
GPS-Net‡† [12]	64.4 / 66.7	19.2 / 21.4	29.6 / 32.4	37.5 / 38.6	11.7 / 12.5	17.8 / 18.9	27.8 / 32.1	7.4 / 9.5	11.7 / 14.7
JMSGG [24]	**70.8 / 71.7**	24.9 / 28.0	36.8 / 40.3	**43.4 / 44.2**	13.1 / 14.7	20.1 / 22.1	29.3 / 32.3	9.8 / 11.8	14.7 / 17.3
BGNN [10]†	59.2 / 61.3	30.4 / 32.9	**40.2 / 42.8**	37.4 / 38.5	14.3 / 16.5	20.7 / 23.1	31.0 / 35.8	10.7 / 12.6	15.9 / 18.7
SQUAT† (Ours)	55.7 / 57.9	**30.9 / 33.4**	39.7 / 42.4	33.1 / 34.4	**17.5 / 18.8**	**22.9 / 24.3**	24.5 / 28.9	**14.1 / 16.5**	**17.9 / 21.0**

Table 1. Recall, mean recall and F score of three subtasks on Visual Genome (VG) dataset with graph constraints. † denotes that the bi-level sampling [10] is applied for the model. ‡ denotes that the results are reported from the [2]. Bold numbers indicate the best performances and underlined numbers indicate the second best performances.

Methods	PredCls			SGCls			SGDet		
	ng-mR@20	ng-mR@50	ng-mR@100	ng-mR@20	ng-mR@50	ng-mR@100	ng-mR@20	ng-mR@50	ng-mR@100
IMP+‡* [11]	-	20.3	28.9	-	12.1	16.9	-	5.4	8.0
Frequency‡* [28]	-	24.8	37.3	-	13.5	19.6	-	5.9	8.9
Motifs‡* [28]	-	27.5	37.9	-	15.4	20.6	-	9.3	12.9
KERN [2]	-	36.3	49.0	-	19.8	26.2	-	11.7	16.0
GB-NET-β [27]	-	44.5	58.7	-	25.6	32.1	-	11.7	16.6
Motifs [28]	19.9	32.8	44.7	11.3	19.0	25.0	7.5	12.5	16.9
VCTree [22]	21.4	35.6	47.8	14.3	23.3	31.4	7.5	12.5	16.7
VCTree-TDE [21]	20.9	32.4	41.5	12.4	19.1	25.5	7.8	11.5	15.2
GPS-Net†* [12]	29.4	45.4	57.1	8.3	15.9	23.1	7.9	12.1	16.7
SQUAT †	**31.8**	**46.0**	**57.8**	**18.7**	**27.1**	**32.6**	**12.1**	**17.9**	**22.5**

Table 2. The scene graph generation performance of three subtasks on the Visual Genome (VG) dataset without graph constraints. † denotes that the bi-level sampling [10] is applied for the model. $*$ denotes that the model is reproduced with the authors' code. ‡ denotes that the results are reported from the [2]. Models in the first group use pre-trained Faster R-CNN with VGG16 backbone. Bold numbers indicate the best performances.

model	head	body	tail	mR@100	R@100	F@100
VCTree-TDE [21]	24.5	13.9	0.1	11.1	23.2	15.0
GPSNet† [12]	30.4	8.5	3.8	9.5	32.1	14.7
BGNN† [10]	**34.0**	12.9	6.0	12.6	**35.8**	18.6
SQUAT † (Ours)	29.5	**16.4**	**12.4**	**16.5**	28.9	**21.0**

Table 3. mR@100 on the SGDet setting for head, body, and tail classes. † denotes that the bi-level sampling is applied on the model to achieve these results. Bold numbers indicate the best performances.

2.2. Mean recall with no-graph constraints

Following [16, 28], we also evaluate SQUAT without the graph constraint, *i.e.*, each edge can have multiple relationships. For each edge, while mR@K evaluates only one predicate with the highest score, ng-mR@K evaluates all 50 predicates. As shown in Table. 2, on the Visual Genome dataset, SQUAT outperforms the state-of-the-art models. Especially, SQUAT outperforms the state-of-the-art models by a large margin of ng-mR@K on the SGDet settings as it does in the evaluation of mR@K.

2.3. Recall for head, body, and tail classes

Following [10], we split the relationship classes into three sets according to the number of relationship instances: head (more than 10k), body (0.5k$\sim$10k), and tail (less than 0.5k) classes. Table 3 shows the mR@100 for each group. SQUAT outperforms the state-of-the-art methods for body and tail classes by a large margin. Especially for the tail classes, SQUAT achieves twice mR@100 as that of BGNN.

model	simple	moderate	complex	mR@100
BGNN [18]	15.52	12.71	9.87	12.46
SQUAT	19.54	16.80	13.28	16.47
Gain (%)	25.90	32.18	34.55	32.18

Table 4. mR@100 on the simple, moderate, and complex sets.

It shows that the scene graphs from SQUAT have more meaningful predicates, *i.e.*, tail classes such as 'walking in', instead of general predicates, *i.e.*, head classes such as 'on'.

2.4. Recall on simple, moderate, and complex scenes

As shown in Tables 1 and 2 in the main paper, the SQUAT shows exceptionally high performance on the most complicated task, *i.e.*, SGDet, and the most complex dataset, *i.e.*, Visual Genome. Furthermore, to analyze the performance on the complexity of the scene, we divide the image sets in the Visual Genome into three disjoint sets according to the number of objects in the scene: simple (≤ 9), moderate ($10 \sim 16$), and complex (≥ 17). As shown in Table 4, the SQUAT shows a higher performance gain on the more complex images; the SQUAT is more effective for realistic and complex scenes.

3. SQUAT with off-the-shelf method

To reduce the biases of the scene graph generation datasets, many off-the-shelf methods [3–6, 8, 9, 20, 23, 25, 26, 29] are proposed. For a fair comparison, we do not compare the off-the-shelf methods with SQUAT in the main paper. We applied Internal and External Data Transfer (IETrans) and reweighting (Rwt) [29], which are the state-of-the-art off-the-shelf learning methods for scene graph generation, to the SQUAT. For efficiency, we only report a model with the best performance for each off-the-shelf method. As shown in Table 5, without careful hyper-parameter search, SQUAT+IETrans+Rwt model outperforms VCTree+IETrans+Rwt model and outperforms other off-the-shelf methods with Motifs [28], Transformer [21], and VCTree [22]. It shows that other off-the-shelf learning methods can be adopted for SQUAT to improve its performance.

4. Additional qualitative results

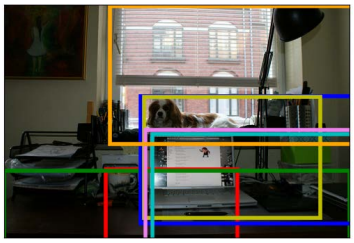
In Fig. 1 and 2, we show the qualitative results for SQUAT model. We also compare the results of SQUAT with the results from ablated models: model without node updates and model without edge updates. The full SQUAT model shows the most informative scene graph compared to the other ablated models. There are some false positives, such as ('mouth of elephant', 'eye of elephant') in Fig. 1 bottom and ('glasses on man', 'man and woman') in Fig. 2

model	SGDet mR@20	SGDet mR@50	SGDet mR@100
VCTree [22]	4.9	6.6	7.7
VCTree+TDE [21]	6.3	8.6	10.3
VCTree+PCPL ‡ [25]	8.1	10.8	12.6
VCTree+DLFE [3]	8.6	11.8	13.8
VCTree+TDE+EBM [20]	7.1	9.69	11.6
Transformer+BPL+SA [6]	10.7	13.5	15.6
Transformer+HML [4]	11.4	15.0	17.7
GPSNet+IETrans+Rwt [29]	-	16.2	18.8
SQUAT +IETrans+Rwt [29]	**12.0**	**16.3**	**19.1**

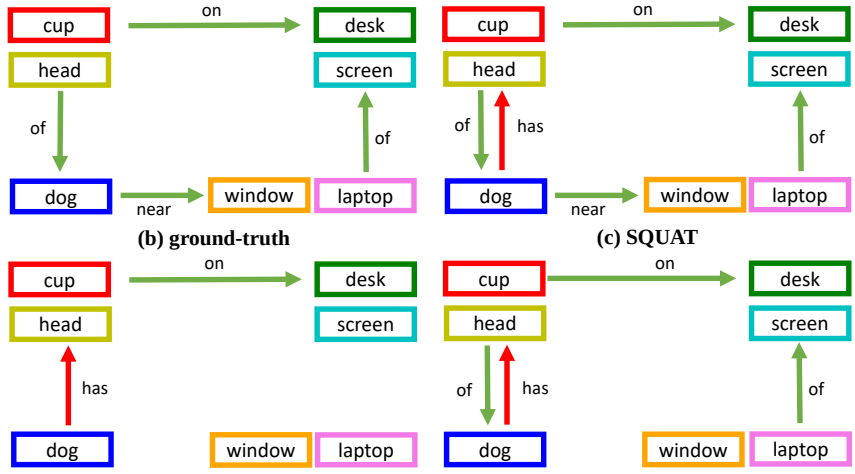
Table 5. The ablation study with the off-the-shelf learning methods on Visual Genome (VG) dataset with graph constraint. ‡ denotes that the results are reported from the [3]. The other results are from each of the original papers.

top, however, such errors are often caused by the incompleteness of the dataset, and hence it can be seen as a true positive. 'dog near window' in Fig. 1 top, 'zebra behind elephant' in Fig. 1 bottom, and 'man standing on sidewalk' in Fig. 2 bottom are predicted by only the full SQUAT model. It shows that quad attention modules can capture more informative contextual information.

In Fig. 3, we show the qualitative results for the edge selection module in SQUAT. The edge selection module successfully selects the valid edges. In particular, the edge selection module removes the edges between the background and the foreground, *e.g.*, most of the edges of 'sunhat' and 'scarf' are removed in Fig. 3 (a) and (b), respectively. Also, the edges between the boxes which denote the same objects are removed. For example, the edges of ('tea', 'coffee') and ('mug', 'coffee cup') are removed in Fig. 3 (d). It shows that the edge selection module successfully removed invalid edges and helps the informative message passing in the quad attention module.

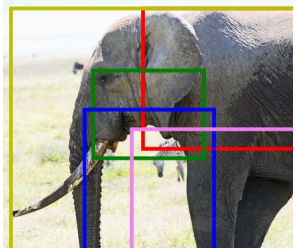


(a) detection results

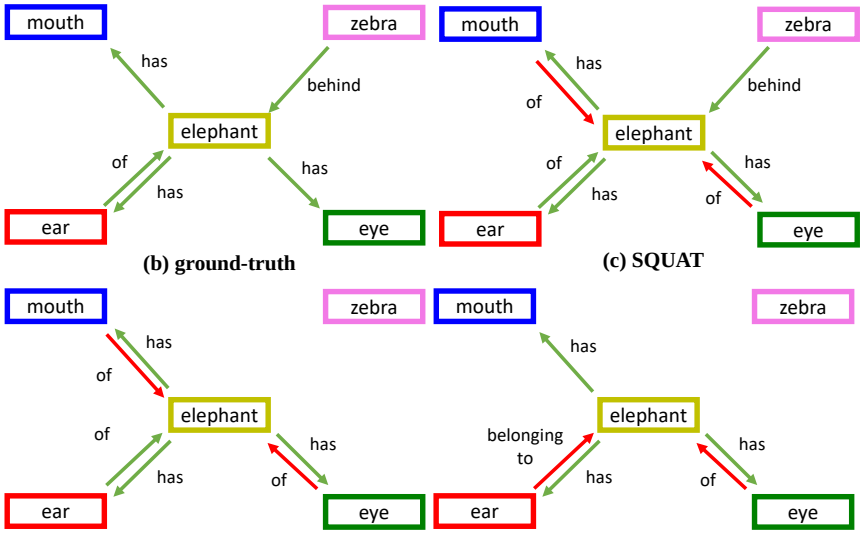


(b) ground-truth (c) SQUAT

(d) w.o. edge update (e) w.o. node update



(a) detection results



(b) ground-truth (c) SQUAT

(d) w.o. edge update (e) w.o. node update

Figure 1. The qualitative results for SQUAT. (a) The detection results from pre-trained Faster R-CNN [18]. (b) The ground-truth scene graph. (c) The results from full SQUAT. (d) The results from SQUAT without edge update, *i.e.*, the edge-to-edge and the edge-to-node attentions. (e) The results from SQUAT without node update, *i.e.*, the node-to-edge and the node-to-node attentions. Full SQUAT shows more informative scene graphs than the other ablated models. The green arrows denote the true positives and the red arrows denote the false positives.

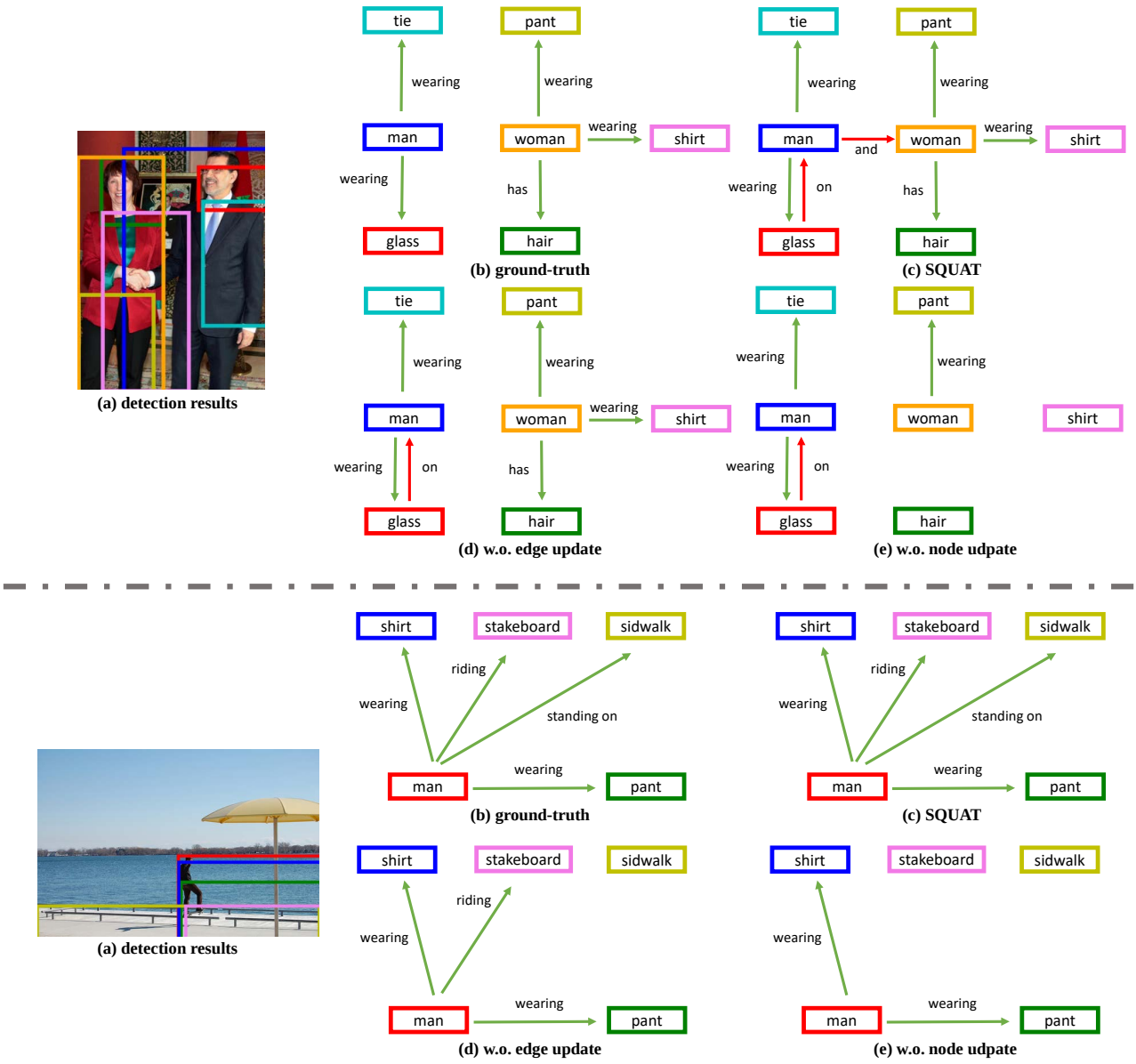


Figure 2. The qualitative results for SQUAT. (a) The detection results from pre-trained Faster R-CNN [18]. (b) The ground-truth scene graph. (c) The results from full SQUAT. (d) The results from SQUAT without edge update, *i.e.*, the edge-to-edge and the edge-to-node attentions. (e) The results from SQUAT without node update, *i.e.*, the node-to-edge and the node-to-node attentions. Full SQUAT shows more informative scene graphs than the other ablated models. The green arrows denote the true positives and the red arrows denote the false positives.

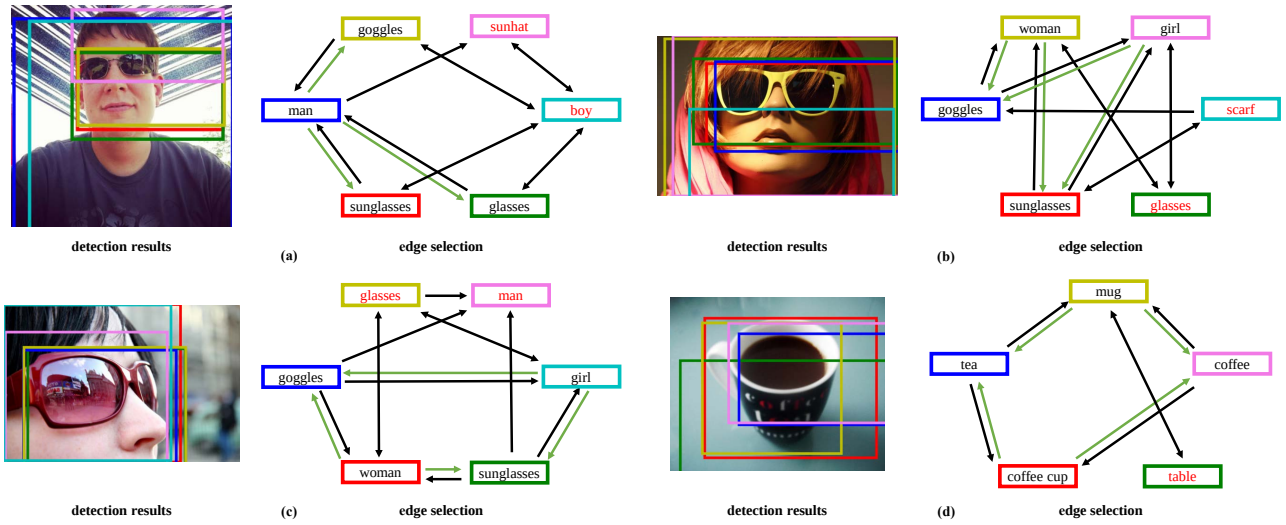


Figure 3. The qualitative results for the edge selection module on the Open Images v6 dataset. The graph denotes the results of the $\mathrm{ESM}^{\mathrm{Q}}$ and the green arrows denote the valid edges. The boxes with the red class denote the incorrect prediction or the background.

References

[1] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. 1

[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

[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. 3

[4] Youming Deng, Yansheng Li, Yongjun Zhang, Xiang Xiang, Jian Wang, Jingdong Chen, and Jiayi Ma. Hierarchical memory learning for fine-grained scene graph generation. *arXiv preprint arXiv:2203.06907*, 2022. 3

[5] 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. 3

[6] 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. 3

[7] Dan Hendrycks and Kevin Gimpel. Bridging nonlinearities and stochastic regularizers with gaussian error linear units. 2016. 1

[8] 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. 3

[9] 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. 3

[10] 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

[11] 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. 2

[12] 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

[13] 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. 2

[14] 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. 2

[15] 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

[16] Alejandro Newell and Jia Deng. Pixels to graphs by associative embedding. *Advances in neural information processing systems*, 30, 2017. 2

[17] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. 1

[18] 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. 4, 5

[19] 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. 1

[20] 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. 3

[21] 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. 1, 2, 3

[22] 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. 2, 3

[23] 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. 3

[24] Minghao Xu, Meng Qu, Bingbing Ni, and Jian Tang. Joint modeling of visual objects and relations for scene graph gen-

eration. *Advances in Neural Information Processing Systems*, 34, 2021. 2

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

[26] 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. 3

[27] 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

[28] 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. 2, 3

[29] Ao Zhang, Yuan Yao, Qianyu Chen, Wei Ji, Zhiyuan Liu, Maosong Sun, and Tat-Seng Chua. Fine-grained scene graph generation with data transfer. In *ECCV*, 2022. 1, 3

[30] Ji Zhang, Kevin J Shih, Ahmed Elgammal, Andrew Tao, and Bryan Catanzaro. Graphical contrastive losses for scene graph generation. 2019. 2