Supplementary Materials for From Pixels to Graphs: Open-Vocabulary Scene Graph Generation with Vision-Language Models

A. More Experiment Configurations

Datasets and tasks We evaluate our method on both SGG task and downstream VL-tasks. For SGG task, we use two large-scale SGG benchmarks: Panoptic Scene Graph Generation (PSG) [9], Visual Genome (VG) [3]. We mainly adopt the data splits from the previous work [5, 9, 11]. For Visual Genome [3] dataset, we take the same split protocol as [8, 11] where 62,723 images are used for training, 26,446 for test, and 5,000 images sampled from the training set for validation. The most frequent 150 object categories and 50 predicates are adopted for evaluation. The Panoptic Scene Graph Generation [9] has 4,4967 images are used for training, 1,000 for test, and 3,000 images sampled from the training set for validation. There 133 object categories and 56 predicates categories in total. We use it bound box annotation for SGG task rather than segmentation masks.

For the open-vocabulary predicate SGG settings, we randomly select 30% predicate categories as novel class. For VL tasks, we inspect our model on VL-task which potentially need the visual scene representation, such as visual grounding on RefCOCO/+/g [7, 10], visual question answering on GQA [2], and image captioning on COCO image caption [1].

Implementation Details We initialize our PGSG by using the BLIP [4] model with ViT-B/16 as the visual backbone and BERT$_{base}$ as the text decoder. For scene graph training process, we use the image size of 384 times 384, an AdamW [6] optimizer with lr = 1e-5, weight decay of 0.02 with a cosine scheduler.We increase the learning rate of position adaptors to 1e-4 for faster convergence. We train our model on 4 A100 GPUs with 50 epoch. For downstream tasks fine-tuning, we following the training setup of BLIP [4]. We use the image encoder and text decoder of PGSG model, and the text encoder and word embedding remain the same as in the pre-trained BLIP model. During the scene graph sequence generation, we generate M=32 number of sequences which length is L=24. For category amplifier β_i, we set this hyper-parameter as 5.0 for entity categories and 1.0 for predicate classification.

B	M	Zs Trp.	Standard SGG				
		R@50/100	mR50	R50	wmAP		score_wtd
					phr	rel	
R101	RelDN	-	39.7	72.1	28.7	29.1	38.6
	HOTR	-	36.8	52.6	21.5	19.4	26.8
	SGTR	-	38.6	59.1	36.9	38.7	42.8
ViT-B*	SGTR	19.4/31.6	30.5	52.6	**28.0**	**22.7**	**30.8**
	PGSG	**23.1/38.6**	**40.7**	**62.0**	27.8	19.7	28.7

Table 1. **The close-vocabulary SG-Det performance on Open-Image V6.**

Prompt	Open Vocab SGG			ZS Trp.
	Novel+Base		Novel	
	mR@50/100	R@50/100	mR@50/100	R@50/100
A	8.2/10.5	14.5/16.4	2.3/7.0	3.6/6.4
B	**9.3/11.7**	**17.7/20.4**	3.7/8.6	**4.4/7.6**
C	9.1/10.1	16.3/19.4	**4.1/9.0**	4.1/6.6

Table 2. **Ablation study on different prompt for SGG task on PSG dataset.** A: ”A visual scene of: ”; B: ”Describe the image by relationships:”; C: ”A picture of: ”

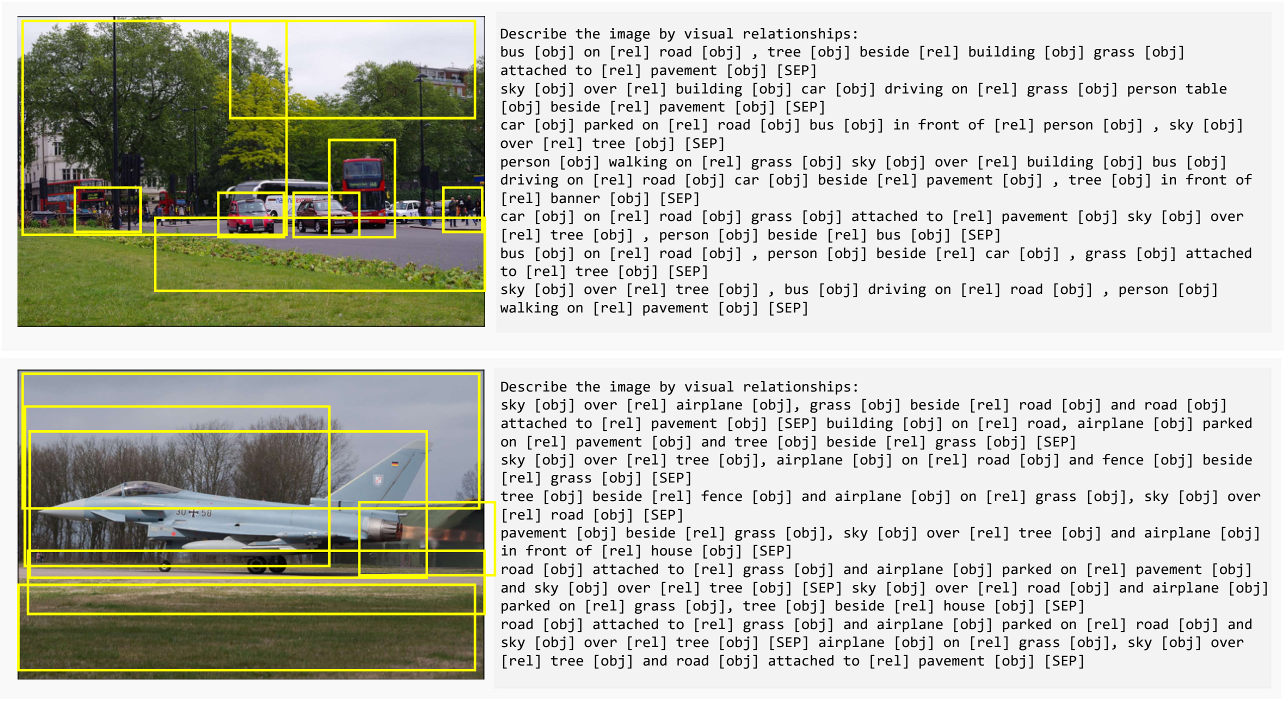
B. More Experimental Results

In this section, we propose mre Experimental results, includes quantitative and quantitative analysis of our method.

C. Quantitative Results

C.1. Close-vocabulary SGG on OpenImage V6

In Tab. 1, we present the close-vocabulary SG-Det performance on OpenImage V6 across various visual backbones and zero-shot triplet (**Zs Trp.**) scenarios. With the same backbone as BLIP ViT-B, our PGSG achieves comparable performance with baseline SGTR in a standard close-vocabulary setting and reasonable performance with the previous one-stage SGG method, which has a larger input resolution ResNet-101 backbone. For compositional generalization setting, zero-shot triplet SGG, our method achieves a remarkable 7.0 improvement over the SGTR baseline.



```
Describe the image by visual relationships:
bus [obj] on [rel] road [obj] , tree [obj] beside [rel] building [obj] grass [obj]
attached to [rel] pavement [obj] [SEP]
sky [obj] over [rel] building [obj] car [obj] driving on [rel] grass [obj] person table
[obj] beside [rel] pavement [obj] [SEP]
car [obj] parked on [rel] road [obj] bus [obj] in front of [rel] person [obj] , sky [obj]
over [rel] tree [obj] [SEP]
person [obj] walking on [rel] grass [obj] sky [obj] over [rel] building [obj] bus [obj]
driving on [rel] road [obj] car [obj] beside [rel] pavement [obj] , tree [obj] in front of
[rel] banner [obj] [SEP]
car [obj] on [rel] road [obj] grass [obj] attached to [rel] pavement [obj] sky [obj] over
[rel] tree [obj] , person [obj] beside [rel] bus [obj] [SEP]
bus [obj] on [rel] road [obj] , person [obj] beside [rel] car [obj] , grass [obj] attached
to [rel] tree [obj] [SEP]
sky [obj] over [rel] tree [obj] , bus [obj] driving on [rel] road [obj] , person [obj]
walking on [rel] pavement [obj] [SEP]
```

```
Describe the image by visual relationships:
sky [obj] over [rel] airplane [obj], grass [obj] beside [rel] road [obj] and road [obj]
attached to [rel] pavement [obj] [SEP] building [obj] on [rel] road, airplane [obj] parked
on [rel] pavement [obj] and tree [obj] beside [rel] grass [obj] [SEP]
sky [obj] over [rel] tree [obj], airplane [obj] on [rel] road [obj] and fence [obj] beside
[rel] grass [obj] [SEP]
tree [obj] beside [rel] fence [obj] and airplane [obj] on [rel] grass [obj], sky [obj] over
[rel] road [obj] [SEP]
pavement [obj] beside [rel] grass [obj], sky [obj] over [rel] tree [obj] and airplane [obj]
in front of [rel] house [obj] [SEP]
road [obj] attached to [rel] grass [obj] and airplane [obj] parked on [rel] pavement [obj]
and sky [obj] over [rel] tree [obj] [SEP] sky [obj] over [rel] road [obj] and airplane [obj]
parked on [rel] grass [obj], tree [obj] beside [rel] house [obj] [SEP]
road [obj] attached to [rel] grass [obj] and airplane [obj] parked on [rel] road [obj] and
sky [obj] over [rel] tree [obj] [SEP] airplane [obj] on [rel] grass [obj], sky [obj] over
[rel] tree [obj] and road [obj] attached to [rel] pavement [obj] [SEP]
```

Figure 1. **The visualization of scene graph sequence prediction of PGSG.**

C.2. Sensitivity of different Prefix Prompts

We also study the different prompt structures for generating the scene graph, as shown in Tab. 2. We experiment with the PGSG framework with different prefix instructions for the scene graph generation task. The results show that more specific instructions yield a slight improvement in performance, which indicates that our method has robustness for different instructions.

C.3. Time Complexity Comparison With Previous Method

Despite potential inference time increases due to the self-regression generation with a large model, we have effectively mitigated this issue by reducing output size. We achieve a boost in inference speed reasonable open-vocabulary SGG performance, in the Tab. 4 of the main paper. We also compare the inference times with other SGG methods, as shown in Tab. 3. The results demonstrate that PGSG attains comparable time efficiency while maintaining its competitive open-vocabulary SGG performance.

C.4. Time Complexity

Despite potential inference time increases due to the self-regression generation with a large model, we have effectively mitigated this issue by reducing output size. We achieve a boost in inference speed reasonable open-vocabulary SGG performance, in the Tab. 4 of the main paper. We also compare the inference times with other SGG methods, as shown in Tab. 3. The results demonstrate that PGSG attains comparable time efficiency while maintaining its competitive open-vocabulary SGG performance.

M	VCTree	GPS-Net	BGNN	PGSG	PGSG*
Time	1.69	1.02	1.32	4.8	1.8

Table 3. **the inference speed (Second per image) comparison with previous two-stage SGG methods**

D. Qualitative Results

We also present the qualitative analysis for the PGSG framework to take a close look at the sequence generation-based SGG framework. In Fig. 1, we show a few examples of generated sequences from our validation set of the PSG dataset. At inference time, the VLM generates scene graph sequences with entity-aware tokens as indicators by using several short token sequences with nucleus sampling, which are able to obtain diverse visual relations. The following entity grounding module extracts the boxes for each entity within the sequences.

References

[1] Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325*, 2015. 1

[2] Drew A Hudson and Christopher D Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6700–6709, 2019. 1

[3] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International journal of computer vision*, 123(1):32–73, 2017. 1

[4] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International Conference on Machine Learning*, pages 12888–12900. PMLR, 2022. 1

[5] 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

[6] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 1

[7] Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Camburu, Alan L Yuille, and Kevin Murphy. Generation and comprehension of unambiguous object descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 11–20, 2016. 1

[8] Danfei Xu, Yuke Zhu, Christopher B Choy, and Li Fei-Fei. Scene graph generation by iterative message passing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition(CVPR)*, pages 5410–5419, 2017. 1

[9] Jingkang Yang, Yi Zhe Ang, Zujin Guo, Kaiyang Zhou, Wayne Zhang, and Ziwei Liu. Panoptic scene graph generation. *arXiv preprint arXiv:2207.11247*, 2022. 1

[10] Licheng Yu, Patrick Poirson, Shan Yang, Alexander C Berg, and Tamara L Berg. Modeling context in referring expressions. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II 14*, pages 69–85. Springer, 2016. 1

[11] 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