Supplementary Material for OpenPSG: Open-set Panoptic Scene Graph Generation via Large Multimodal Models

Zijian Zhou[1] Zheng Zhu[2] Holger Caesar[3] Miaojing Shi[4,5]✉

[1]Department of Informatics, King's College London [2]GigaAI
[3]Intelligent Vehicles Lab, Delft University of Technology
[4]College of Electronic and Information Engineering, Tongji University
[5]Shanghai Institute of Intelligent Science and Technology, Tongji University

In this supplementary material, we provide the division method for base and novel relations in PSG dataset (Sec. A), more experiments on other datasets (Sec. B), additional experimental results (Sec. C), and instructions used in relation query transformer (Sec. D) and multimodal relation decoder (Sec. E) respectively.

A Division for base and novel relations in PSG dataset.

In the PSG dataset, there are a total of 56 predefined relations. To test the open-set performance of our model, we divide the dataset into base and novel relations at a ratio of 7:3. The novel relations, which make up 30% of the total, while the rest are classified as base relations. The detailed relations are as follows.

```
base_relations = ["over", "in front of", "beside", "on", "in", "hanging
    from", "on back of", "going down", "painted on", "walking on", "running
    on", "crossing", "lying on", "sitting on", "jumping over", "jumping
    from", "holding", "carrying", "guiding", "kissing", "drinking",
    "feeding", "catching", "picking", "chasing", "climbing", "playing",
    "touching", "pulling", "opening", "talking to", "throwing", "driving",
    "riding", "driving on", "about to hit", "swinging", "entering",
    "exiting", "enclosing", "leaning on"]
novel_relations = ["attached to", "falling off", "walking on ", "standing
    on", "flying over", "wearing", "looking at", "eating", "biting",
    "playing with", "cleaning", "pushing", "cooking", "slicing", "parked
    on", "kicking", "existing"]
all_relations = base_relations + novel_relations
```

B More datasets

GQA dataset [3] employs the same images as the VG dataset [4], but includes more comprehensive annotations of objects and relations. Following prior work [2], we conduct our experiments using the GQA200 variant, which encompasses 200 object categories and 100 relation categories.

To further validate our method, we evaluate our method on the GQA200 in both closed-set and open-set SGG scenarios (Tab. 2). Our method surpasses the previous best-performing method [2] by a sizeable margin (+5.6% in R@100 and +1.3% in mR@100).

✉ Corresponding author.

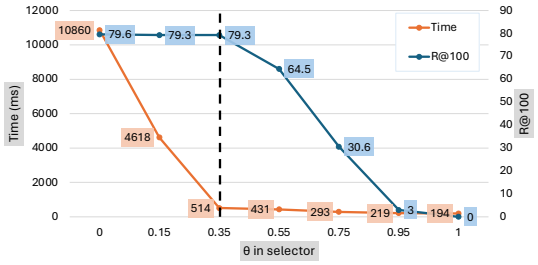

Fig. 1: θ in the selector of RelQ-Former.

Table 1: Ablation study of RelQ-Former with different numbers of layers.

	Predicate Classification					
Layer Num	R@20	mR@20	R@50	mR@50	R@100	mR@100
1	47.1	30.6	60.4	41.8	68.6	50.9
2 (Ours)	55.1	39.2	70.6	53.8	79.3	63.8
4	55.1	39.5	70.7	54.0	79.8	64.1
6	54.2	38.5	69.4	52.8	78.3	62.8

C More experimental results

Effect of θ in the selector of RelQ-Former. To select the optimal θ to balance model performance and efficiency, we experiment with different θ parameters for the selector. As shown in Fig. 1, as θ increases from 0.0 to 0.35, the time required per image decreases rapidly, whereas there is no significant change as θ increases from 0.35 to 1.0. At the same time, as θ rises from 0.0 to 0.35, there is only a minor decline in model performance; however, beyond 0.35, the performance of the model begins to decrease rapidly. Therefore, to balance performance and efficiency, we select $\theta = 0.35$, thus ensuring a short average processing time for an image while maintaining high performance.

Number of layers in RelQ-Former. To determine the number of layers for RelQ-Former, we conduct experiments with different numbers of layers, and the results are shown in Tab. 1. When we increase the number of layers from 1 to 2, the model performance clearly improves, with R@100 increasing by 10.7% and mR@100 by 12.9%. However, when the number of layers increases from 2 to 4, the model performance only sees a minimal improvement, with R@100 increasing by 0.5% and mR@100 by 0.3%. Further increasing the number of layers from 4 to 6 leads to a slight decrease in model performance. Considering that RelQ-Former requires a balanced layer number to extract features and learn relations without overwhelming computation for all possible subject-object pairs, i.e., $N \times (N-1)$ pairs. Therefore, a layer number of 2 is the optimal choice for balancing performance and efficiency.

Table 2: Results on GQA: PredCls subtask in closed- and open-set scenarios.

	Closed-set		Open-set (base:novel=7:3)	
Method	R/mR@50	R/mR@100	R/mR@50	R/mR@100
SHA+GCL [2]	41.0/42.7	42.7/44.5	– / –	– / –
OpenPSG	46.2/44.1	48.3/45.8	23.6/22.7	26.2/25.0

Table 3: Results on PSG: SGDet subtask in the open-set scenario.

Method	R/mR@20	R/mR@50	R/mR@100
OpenPSG	25.9/20.9	31.6/24.0	36.7/25.4
multi-division	26.4±0.7/21.5±0.6	31.4±0.4/24.1±0.4	36.5±0.4/25.7±0.3
fuse G and J	26.3/21.2	32.0/24.6	37.2/26.3

Impact of different data divisions on the results. We conduct 5 random divisions of the base and novel classes in the PSG dataset and show that the results (mean and variance in Tab. 3: multi-division) are rather stable, attesting to the robustness of our method.

Fusion of generation and judgement instructions. Judgement instruction is used by default, while generation instruction is a variant inspired by [7] for comparison. They are not combined in the paper, but we can simply fuse their outputs using the inference fusion method in [8,9], the results in Tab. 3 (fuse G and J) show slight improvement (+0.5% in R@100) but also come with double computation cost.

D Instructions used in Relation Query Transformer

We introduce the instructions used in the relation query former (Sec. 4.2). To prevent the model from overfitting to a particular type of instruction during training, following [1,5,6], we design 10 different variations for each category of instruction. During training, one is randomly selected from these 10 options for use, whereas for inference, only the first one is chosen for evaluation.

D.1 Instructions for Pair Feature Extraction Query.

```
# {subject} is the category name of subject.
# {object} is the category name of object.
instructions_for_feat_query = [
    "Please extract features for the {subject}-{object} pair based on the
    whole visual features of the image and the masks of the {subject} and
    {object}.",
    "Based on the image's holistic visual features and the masks of both
    {subject} and {object}, please derive the features of the
    {subject}-{object} pair.",
    "Utilizing the total visual features of the image, along with the
    {subject} and {object} masks, please identify the features of the
    {subject}-{object} pair.",
```

```
    "By considering the comprehensive visual features of the image and the respective masks of the {subject} and {object}, please isolate the features specific to the {subject}-{object} pair.",
    "Taking into account the global visual features of the image and the masks designated for the {subject} and {object}, please extract the particular features of the {subject}-{object} pair.",
    "Leveraging the entire visual features of the image as well as the masks for the {subject} and {object}, please delineate the features corresponding to the {subject}-{object} pair.",
    "Drawing on the overall visual features of the image and the masks of the {subject} and {object}, please ascertain the features for the {subject}-{object} pair.",
    "By harnessing the full visual features of the image along with the masks of the {subject} and {object}, please identify the distinctive features of the {subject}-{object} pair.",
    "With reference to the comprehensive visual features of the image and the masks for the {subject} and {object}, please extract the respective features of the {subject}-{object} pair.",
    "Considering the total visual features of the image and the defined masks for the {subject} and {object}, please determine the specific features of the {subject}-{object} pair.",
]
```

D.2 Instructions for Relation Existence Estimation Query.

```
# {subject} is the category name of subject.
# {object} is the category name of object.
instructions_for_exist_query = [
    "Based on the visual features of the entire image and the masks for the {subject} and {object}, estimate whether there is a relation between the {subject} and {object}.",
    "Considering the holistic visual features of the image and the masks of the {subject} and {object}, determine whether a relation exists between the two entities.",
    "Utilizing the complete visual features of the image along with the {subject} and {object} masks, assess whether there is a relation between the {subject} and {object}.",
    "Given the overall visual features of the image and the masks for the {subject} and {object}, evaluate whether a relation exists between the {subject} and {object}.",
    "By analyzing the entire visual features of the image and the masks of the {subject} and {object}, ascertain whether there is a relation between the {subject} and {object}.",
    "With the comprehensive visual features of the image and the masks for both {subject} and {object}, deduce whether there is a relation between the {subject} and {object}.",
    "Reflecting on the total visual features of the image and the masks applied to the {subject} and {object}, gauge whether there is a relation between the {subject} and {object}.",
    "Considering the full visual features of the image and the masks of the {subject} and {object}, infer whether a relation exists between the {subject} and {object}.",
    "Leveraging the overall visual features of the image and the masks designated for the {subject} and {object}, identify whether there is a relation between the {subject} and {object}.",
    "By examining the entire visual features of the image and the masks for the {subject} and {object}, predict whether there is a relation between the {subject} and {object}.",
]
```

E Instructions used in Multimodal Relation Decoder

We introduce the instructions used in the multimodal relation decoder (Sec. 4.3). Same as Sec. D, to avoid model overfitting to a particular type of instruction during training, we design 10 different variations for each type of instruction. During training, one is randomly selected from these 10 options for use, whereas for inference, only the first one is chosen for evaluation [39].

E.1 Generation Instructions.

```
# {subject} is the category name of subject.
# {object} is the category name of object.
instruction_for_generation = [
    "Please determine what the relation is between {subject} and {object}.",
    "Please ascertain the relation between the {subject} and {object}.",
    "Please identify what relations exists between the {subject} and
    {object}.",
    "Please decide what kind of relation is present between the {subject}
    and the {object}.",
    "Please deduce the relation between the {subject} and the {object}.",
    "Please establish what the relation is between the {subject} and
    {object}.",
    "Please clarify the relation between the {subject} and {object}.",
    "Please determine the type of relation existing between the {subject}
    and the {object}.",
    "Please pinpoint the kind of relation between the {subject} and
    {object}.",
    "Please evaluate what the relation is between the {subject} and the
    {object}.",
]
```

E.2 Judgement Instructions.

```
# {subject} is the category name of subject.
# {object} is the category name of object.
# {relation} is the category name of relation.
instructions_for_judgement = [
    "Please judge between {subject} and {object} whether there is a relation
    {relation}.",
    "Please determine if there exists a relation between the {subject} and
    {object}, termed {relation}.",
    "Please ascertain whether there is a relation between the {subject} and
    {object}, identified as {relation}.",
    "Please evaluate if a relation between the {subject} and {object} can be
    classified as {relation}.",
    "Please judge whether there is a relation between the {subject} and
    {object} referred to as {relation}.",
    "Please decide if there is a relation between the {subject} and {object}
    denoted as {relation}.",
    "Please establish whether there is a relation between the {subject} and
    {object}, described as {relation}.",
    "Please conclude whether a relation exists between the {subject} and
    {object}, designated as {relation}.",
    "Please investigate whether there is a relation between the {subject}
    and {object}, recognized as {relation}.",
    "Please analyze if there exists a relation between the {subject} and
    {object}, characterized as {relation}.",
]
```

References

1. Dai, W., Li, J., Li, D., Tiong, A.M.H., Zhao, J., Wang, W., Li, B., Fung, P.N., Hoi, S.: Instructblip: Towards general-purpose vision-language models with instruction tuning. NeurIPS (2024) 3
2. Dong, X., Gan, T., Song, X., Wu, J., Cheng, Y., Nie, L.: Stacked hybrid-attention and group collaborative learning for unbiased scene graph generation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 19427–19436 (2022) 1, 3
3. Hudson, D.A., Manning, C.D.: 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. pp. 6700–6709 (2019) 1
4. Krishna, R., Zhu, Y., Groth, O., Johnson, J., Hata, K., Kravitz, J., Chen, S., Kalantidis, Y., Li, L.J., Shamma, D.A., et al.: Visual genome: Connecting language and vision using crowdsourced dense image annotations. International journal of computer vision pp. 32–73 (2017) 1
5. Liu, H., Li, C., Wu, Q., Lee, Y.J.: Visual instruction tuning. NeurIPS **36** (2024) 3
6. Yu, Q., Shen, X., Chen, L.C.: Towards open-ended visual recognition with large language model. arXiv preprint arXiv:2311.08400 (2023) 3
7. Zhang, H., Li, F., Zou, X., Liu, S., Li, C., Yang, J., Zhang, L.: A simple framework for open-vocabulary segmentation and detection. In: ICCV. pp. 1020–1031 (2023) 3
8. Zhou, Z., Shi, M., Caesar, H.: Hilo: Exploiting high low frequency relations for unbiased panoptic scene graph generation. arXiv preprint arXiv:2303.15994 (2023) 3
9. Zhou, Z., Shi, M., Caesar, H.: Vlprompt: Vision-language prompting for panoptic scene graph generation. arXiv preprint arXiv:2311.16492 (2023) 3