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

Abstract. Panoptic Scene Graph Generation (PSG) aims to segment objects and recognize their relations, enabling the structured understanding of an image. Previous methods focus on predicting predefined object and relation categories, hence limiting their applications in the open world scenarios. With the rapid development of large multimodal models (LMMs), significant progress has been made in open-set object detection and segmentation, yet open-set relation prediction in PSG remains unexplored. In this paper, we focus on the task of open-set relation prediction integrated with a pretrained open-set panoptic segmentation model to achieve true open-set panoptic scene graph generation (**OpenPSG**). Our OpenPSG leverages LMMs to achieve open-set relation prediction in an autoregressive manner. We introduce a relation query transformer to efficiently extract visual features of object pairs and estimate the existence of relations between them. The latter can enhance the prediction efficiency by filtering irrelevant pairs. Finally, we design the generation and judgement instructions to perform open-set relation prediction in PSG autoregressively. To our knowledge, we are the first to propose the open-set PSG task. Extensive experiments demonstrate that our method achieves state-of-the-art performance in open-set relation prediction and panoptic scene graph generation. Code is available at https://github.com/franciszzj/OpenPSG.

Keywords: Panoptic Scene Graph Generation · Open-set · Large Multimodal Models

1 Introduction

Panoptic scene graph generation (PSG) [36] aims to segment objects within an image and recognize the relations among them, thereby constructing a panoptic scene graph for a structured understanding of the image. Given its significant potential in applications such as visual question answering [13], image captioning [4,10], and embodied navigation [29], PSG has attracted considerable attentions from researchers ever since its emerging [19,33,35,46–48].

✉ Corresponding author.

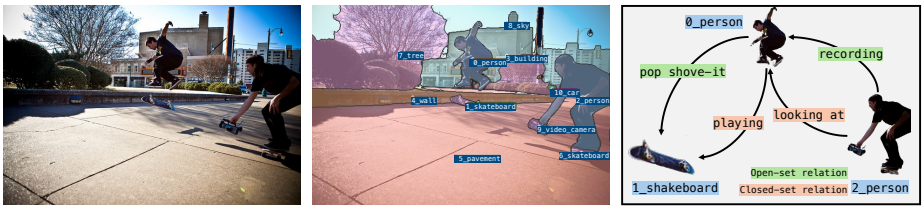

Fig. 1: The left image is the input to our OpenPSG, the middle one displays the panoptic segmentation result, and the right one shows the predicted relations between objects. Our method can predict both known (close-set) relations, *e.g.*, (0_person, *playing*, 1_skateboard), (2_person, *looking at*, 0_person), and unknown (open-set) relations, *e.g.*, (0_person, *pop shove-it*, 1_skateboard), (2_person, *recording*, 0_person).

Previous PSG methods [33, 47, 48] are only capable of predicting closed-set object and relation categories while failing to recognize objects/relations beyond predefined categories. Recently, with the advent of large multimodal models (LMMs) such as CLIP [27], BLIP-2 [18] *etc.*, a significant number of open-set prediction methods for object detection [9, 21, 37, 42, 44] and segmentation [11, 20, 39, 44] are introduced, attributing to LMMs' rich understanding of language and strong connections between vision and language. Nevertheless, open-set prediction of relations has been largely unexplored so far.

Compared to open-set object detection and segmentation, open-set relation prediction is more complex: the model is required to both understand different objects and recognize relations of object pairs based on their interactions; especially, the computation of the latter can be exponentially increased. To bridge the gap, in this paper, we focus on the open-set relation prediction.

LLMs [18, 25, 49] have demonstrated exceptional semantic analysis and understanding abilities across various multimodal tasks. In particular with the text processing, LMMs are not only good at interpreting on nouns (*i.e.* representing objects) but also pay considerable attention on predicates (*i.e.* representing relations between objects), ensuring their generated contents to be sufficiently coherent [1]. Inspired by this, we propose the **Open**-set **P**anoptic **S**cene **G**raph Generation architecture, **OpenPSG**, leveraging the capabilities of LMMs for open-set relation prediction.

To this end, we utilize a large multimodal model (*e.g.*, BLIP-2 [18]) to achieve open-set relation prediction. Specifically, our model comprises three parts. First, the *open-set panoptic segmenter*, we adapt an existing model (*e.g.*, OpenSeeD [44]) which is capable of extracting open-set object categories, masks, and visual features from the whole image, forming object pairs and pair masks. Second, the *relation query transformer*, which has two functions: extracting visual features of object pairs based on pair masks and with a special focus on pair interactions; judging the potential relations between object pairs. They are realized by two sets of queries, pair feature extraction query and relation existence estimation query. Only those object pairs that are judged to likely have

relations are fed into the third part, the *multimodal relation decoder*. This decoder directly inherits from the LMM to predict the open-set relations given an object pair in an auto-regressive manner, on condition of specifically-designed text instructions and pre-extracted pair visual features.

To the best of our knowledge, we are the first to propose the task of open-set panoptic scene graph generation, enabling the open-set prediction of both object masks and relations. Extensive experiments demonstrate that our OpenPSG achieves state-of-the-art results in the closed-set setting and exhibits outstanding performance in the open-set setting.

2 Related Work

2.1 Panoptic Scene Graph Generation

Panoptic scene graph generation stems from scene graph generation (SGG) by replacing bounding boxes with panoptic segmentation masks to represent the objects, so as to achieve a more comprehensive understanding of scenes. Following the introduction of PSG [36], a series of related works [19, 33, 36, 46–48] emerge, significantly advancing its performance. For example, Yang *et al*. [36], based on DETR [3], introduce an end-to-end framework with learnable queries to generate panoptic scene graphs. Wang *et al*. [33] design a pair proposal network to filter irrelative subject-object pairs, achieving performance improvement of the PSG. Subsequently, Zhou *et al*. [47] build a HiLo architecture based on Mask2Former [6] and devise separate branches for high- and low-frequency relations respectively, hence achieving an unbiased relation prediction method. Li *et al*. [19] re-balance the relation prediction by adaptively transferring information from high-frequency to low-frequency relations. Additionally, Zhao *et al*. [46] implement a weakly-supervised PSG method given only image-text pairs as annotations, allowing for the learning of panoptic scene graphs from image-level ground truth. Recently, Zhou *et al*. [48] leverage the rich language information inherent in large language models [1] and design effective image-text interaction modules to assist unbiased PSG. All these works are learned in the closed-set setting, in this paper, we study the open-set PSG, which has been unexplored.

2.2 Open-set Scene Graph Generation

Open-set SGG has been studied in recent years. Earlier works [15, 40], often termed as zero-shot SGG, focus on transferring the knowledge from known relations to unknown relations given their prior connections; for example, Kan *et al*. [15] leverage the external commonsense knowledge while Yu *et al*. [40] use knowledge graphs for the knowledge transfer. Subsequently, with the advancement of large multimodal models, various open-set SGG works [5, 12, 38, 45] have emerged; for example, He *et al*. [12] and Zhang *et al*. [45] focus on predicting relations between unknown objects in the SGG using multimodal models. Yu *et al*. [38] and Chen *et al*. [5] on the other hand focus on the open-set relation prediction in PSG, sharing the same aim with us. Specifically, Yu *et al*. [38] utilize

the CLIP model to match visual features with textual relation features for open-set relation prediction; Chen *et al.* [5] employ a student-teacher network to align visual concepts in multimodal models for predicting open-set relations. In this paper, different from previous works, we introduce an auto-regressive method based on the LMM to achieve open-set relation prediction.

2.3 Large Multimodal Models

Since the introduction of large models like the GPT series [1], recent years have witnessed rapid development in large multimodal models [18, 25, 27]. Benefiting from its nature of connecting vision and language, LMMs have significantly advanced various downstream tasks, ranging from computer vision [9,39] to natural language processing [14,28,32]. Early multimodal models, such as CLIP [27], trained on image-text paired datasets through contrastive learning to align the visual and textual information. Subsequently, owing to enlightenment of the autoregressive prediction in large language models [1,31], there has been an explosive growth of LMMs [18,24,25]; by introducing mechanisms that can transform visual information into large language models, they facilitate the communication between visual and textual information. Furthermore, this has endowed LMMs with the capability to generate free text, leading to substantial improvements in numerous multimodal tasks. In this paper, we leverage LMMs to design a multimodal relation decoder to predict relations in an open-set scenarios.

3 Task Definition

We define the task, open-set panoptic scene graph generation. Given an image $I \in \mathbb{R}^{H \times W \times 3}$, the objective of this task is to extract an open-set panoptic scene graph $G = \{O, R\}$ from the image I, where H and W are the height and width of the image. Here:

- $O = \{o_i\}_{i=1}^{N}$ represents N objects segmented from the image, each defined as $o_i = \{c, m\}$, where c is the object category that can belong to either predefined base object categories C_{base} or undefined novel object categories C_{novel}. m represents the binary mask in $\{0,1\}^{H \times W}$ of the object.
- $R = \{r_{i,j} \mid i, j \in \{1, 2, \ldots, N\}, i \neq j\}$ represents the relations between objects, where $r_{i,j}$ denotes the relation between o_i and o_j, with o_i as the subject and o_j as the object. Each relation r can belong to either predefined base relation categories K_{base} or undefined novel relation categories K_{novel}.

4 Method

As illustrated in Fig. 2, our OpenPSG comprises three components: object segmenter, relation query transformer (RelQ-Former), and multimodal relation decoder (RelDecoder). For the object segmenter (Sec. 4.1), we utilize a pretrained

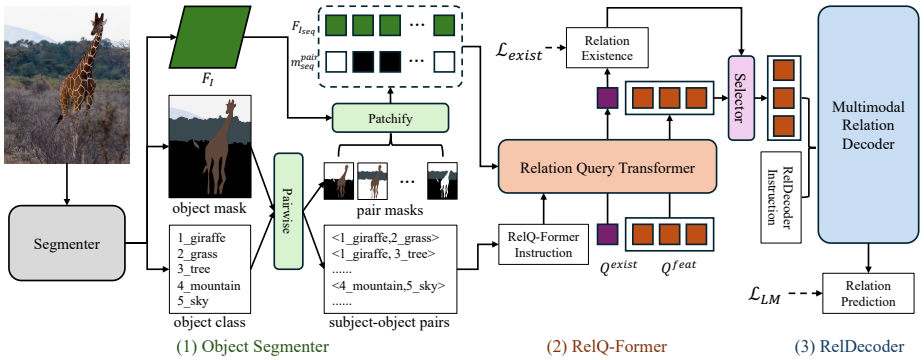

Fig. 2: The overall framework of our OpenPSG, which comprises three components: object segmenter, relation query transformer and multimodal relation decoder.

open-set panoptic segmentation model to transform the input image into object categories and masks, as well as visual feature representing the whole image. Subsequently, we input the object categories, masks, and visual feature into the RelQ-Former (Sec. 4.2). Through two sets of learnable queries and complemented by designed instructions, we obtain visual features of object pairs compatible with LMMs' input format as well as judgements on potential relation existence. Finally, only those object pairs being judged to likely have a relation are sent into the RelDecoder (Sec. 4.3) for open-set relation prediction, ultimately yielding an open-set panoptic scene graph.

4.1 Object Segmenter

Given an image I, we utilize the pretrained open-set object segmenter (*e.g.*, OpenSeeD [44]) to predict the objects O within the image and the whole-image visual feature $F_I \in \mathbb{R}^{h \times w \times D}$. Here, h and w represent the height and width of F_I, and D denotes the feature dimension. The segmenter has a similar architecture to Mask2Former [6] includeing a pixel decoder. The whole-image visual feature F_I refers to the visual feature output by the pixel decoder. Below, we develop the patchify module and pairwise module to process the output of the segmenter, generating the input for RelQ-Former.

Patchify Module. Patchify module aims to serialize visual feature F_I and object masks m, enabling them to be processed as inputs by the RelQ-Former (Sec. 4.2). Similar to the input patchify layer of vision transformer (ViT) [8], we utilize a single convolution layer to transform the extracted F_I into a sequence of visual tokens $F_{Iseq} \in \mathbb{R}^{L \times D}$, where L is the number of patches and D is the feature dimension. When the kernel size and stride of the convolution layer are both p, L is calculated as $L = \frac{h}{p} \times \frac{w}{p}$. Simultaneously, we employ nearest neighbor interpolation to each extracted object's mask m_i, where the size of m_i

is of height $\frac{h}{p}$ and width $\frac{w}{p}$, and then reshape it to a one-dimensional vector with the length L. After processing all masks in the same way, we obtain the mask sequence $m_{seq} \in \{0,1\}^{N \times L}$ for all objects.

Pairwise Module. Pairwise module aims to construct subject-object pairs. Given N objects in the image I, we pairwise all objects O into subject-object pairs $P = \{(o_i, o_j) | i, j \in \{1, 2, \ldots, N\}, i \neq j\}$. The number of subject-object pairs in P is $N \times (N-1)$, which exhibits exponential growth as N increases. Consequently, we also obtain the combined subject-object pair category set $c^{pair} \in \{(c_i, c_j) | i, j \in \{1, 2, \ldots, N\}, i \neq j\}$. We construct the mask sequences for the two objects corresponding to the indices i and j from m_{seq} for each subject-object pair by using the logical OR operation. This operation is performed for all subject-object pairs, resulting in the pair mask sequence $m_{seq}^{pair} \in \{0,1\}^{N \times (N-1) \times L}$ for the subject-object pairs, where L is the number of patches.

4.2 Relation Query Transformer

Relation query transformer, leveraging the obtained F_{Iseq}, c^{pair}, and m_{seq}^{pair}, employs two distinct types of queries, pair feature extraction query and relation existence estimation query, along with customized instructions. This approach facilitates the extraction of subject-object pair features and assesses which subject-object pairs likely have relations.

Pair Feature Extraction Query. The objective of the pair feature extraction query is to extract corresponding subject-object pair features from the whole image visual feature based on the subject-object pair masks. A common extraction method involves mask pooling [7], which extracts features for the target subject-object pair, treating each area on the subject-object pair equally. However, for features used in relation prediction, they should focus more on areas where interactions between objects occur. By leveraging attention mechanisms, we facilitate interactions among visual tokens representing different areas within the visual feature sequence F_{Iseq} of a subject-object pair. This way can enhance areas that are crucial for relation predictions. Furthermore, inspired by [24], we design an instruction to assist the this learnable query in understanding its purpose for extracting subject-object pair features.

Specifically, for each subject-object pair (o_i, o_j), we first input the pair feature extraction query $Q^{feat} \in \mathbb{R}^{E \times D}$ into a self-attention layer $(SA(\cdot))$, along with the pair instruction designed specifically for the pair feature extraction query. This pair instruction is processed through a tokenizer layer to obtain $F_{Inst}^{feat} \in \mathbb{R}^{X^{feat} \times D}$, which specifies the function of the pair feature extraction query, namely "Extracting subject-object (c_i, c_j) features from visual features according to the mask". Here E is the token number of the pair feature extraction query, and X^{feat} is the token number of the pair instruction. Note that we also incorporate the category names of the subject and object (c_i, c_j) into this

pair instruction. This operation is formulated as

$$F_{SA}^{feat} = Trunc(SA(Concat(Q^{feat}, F_{Inst}^{feat})), E), \tag{1}$$

where $Concat(\cdot)$ denotes the concatenation operation, $Trunc(\cdot)$ represents the truncation operation, and E in this truncation operation indicates that we only extract the first E features, namely the features corresponding to the pair feature extraction query. Next, we use a mask cross-attention layer ($MaskCA(\cdot)$), with F_{SA}^{feat} as the query, F_{Iseq} as key and value, and m_{seq} as the mask, to extract features corresponding to the subject-object pair, formulated as

$$F_{CA}^{feat} = MaskCA(F_{SA}^{feat}, F_{Iseq}, m_{seq}). \tag{2}$$

The features F_{CA}^{feat} are further refined through a feed-forward network ($FFN(\cdot)$), formulated as $F_{FFN}^{feat} = FFN(F_{CA}^{feat})$.

By repeating this process twice, we obtain the visual features for the subject-object pair to be input into the multimodal relation decoder, $F_I^{pair(i,j)} \in \mathbb{R}^{E \times D}$. We perform these operations in parallel for all subject-object pairs to obtain the corresponding features for all pairs.

Relation Existence Estimation Query. In addition to the pair feature extraction query, we also design a relation existence estimation query to determine whether a relation likely exists between the subject o_i and object o_j, without predicting the specific relation category. The objective is to filter out irrelevant subject-object pairs to save the computation for subsequent LMM decoding.

Specifically, for each subject-object pair (o_i, o_j), the relation existence estimation query $Q^{exist} \in \mathbb{R}^{1 \times D}$, similar to the pair feature extraction query, is input into the self-attention, mask cross-attention, and feed-forward network layers, interacting respectively with F_{Iseq}, m_{seq} and the specially designed relation instruction. The purpose of the relation instruction is to direct the relation existence estimation query towards determining whether a relation likely exists in the subject-object pair, *e.g.* "Is there a relation between o_i and o_j?" The relation instruction, after being processed by the tokenizer, results in $F_{Inst}^{exist} \in \mathbb{R}^{X^{exist} \times D}$, where X^{exist} represents the number of tokens. Eventually, the extracted features are input into a relation existence prediction layer, which includes a 2-layer MLP, and the predicted scores are normalized to $[0, 1]$ using the sigmoid function. It is worth noting that we train it using binary labels indicating whether a relation exists between objects, and a selector module specified below is utilized to perform filtering during inference.

Selector. The selector module implemented by 2-layer MLP is set to filter irrelvant subject-object pairs. Only those with a score higher than the threshold θ can be input into the multimodal relation decoder. Compared to predicting for all subject-object pairs, this can enables a 20× speedup in our experiment.

4.3 Multimodal Relation Decoder

Multimodal relation decoder aims to utilize the subject-object pair feature $F_I^{i,j}$ extracted by the aforementioned modules, combined with an instruction guiding it to achieve open-set relation prediction. Inspired by [39, 41], we first design a *generation instruction* to perform open-set relation prediction in an autoregressive manner. This works well, yet we find that it tends to favor common relations more or less. Therefore, we further design a *judgement instruction*, leveraging the LMMs' strong analytical and judgment capabilities. The judgement instruction also utilizes an autoregressive manner but to judge whether a specific relation exists between objects, thereby simplifying the complexity of open-set relation prediction. Next, we specify the two instructions, respectively.

Generation Instruction. For the generation instruction, we follow the instruction design used in open-set object recognition [39], utilizing "What are the relations between c_i and c_j?". Here, c_i and c_j respectively refers to the name of the subject and object. We convert this instruction into features $F_{inst}^{gen} \in \mathbb{R}^{X^{gen} \times D}$ using the tokenizer, where X^{gen} is the token number of the generation instruction. We input the features of this generation instruction F_{inst}^{gen} together with the subject-object pair features $F_I^{pair(i,j)}$ into the multimodal relation decoder $Dec(\cdot)$, predicting all possible relations in an autoregressive way, formulated as

$$r_{i,j} = Dec(Concat(F_I^{pair(i,j)}, F_{inst}^{gen})). \tag{3}$$

If multiple relations are predicted, they are separated by the delimiter "[SEP]".

Judgement Instruction. Unlike generation instruction, the judgement instruction guides relation decoder to judge, based on a given relation name, whether this relation exists between the subject and object. For example, "Please judge between c_i and c_j whether there is a relation r_k". In this case, we only need the multimodal relation decoder to answer "Yes" or "No" to determine the existence of this relation. Note that inputting the complete judgement instruction for each relation into the decoder can be costly. Therefore, we place the relation name at the end of the instruction. During inference we divide the judgement instruction into two parts: the section before the relation name, transformed into F_{inst}^{judge} through the tokenizer, and the relation name itself, processed into F_{inst}^{rel}. Benefiting from the autoregressive manner for open-set relation prediction, we initially input the subject-object pair feature $F_I^{pair(i,j)}$ and F_{inst}^{judge} into the multimodal relation decoder, formulated as

$$F_{prefix}^{(i,j)} = Dec(Concat(F_I^{pair(i,j)}, F_{inst}^{judge})), \tag{4}$$

which is then cached for subsequent calculations for each relation. For each relation r_k, the multimodal relation decoder only needs to process F_{prefix} and $F_{inst}^{rel(k)}$ to achieve relation prediction, formulated as:

$$J_{i,j,k} = Dec(Concat(F_{prefix}^{(i,j)}, F_{inst}^{rel(k)})), \tag{5}$$

where $J_{i,j,k}$ represents the judgement for (o_i, r_k, o_j) triplet. When $J_{i,j,k}$ is "Yes", it indicates that the relation r_k exists between o_i and o_j; otherwise, it does not exist. Through this approach, we can maintain the same prediction time as with the generation instruction.

We perform the aforementioned process for all subject-object pairs that may have a relation, ultimately achieving open-set relation prediction. For method using generation instruction, we denote it as OpenPSG-G, and for those using judgement instruction, as OpenPSG-J. In next section, OpenPSG by default refers to the latter.

4.4 Loss Function

During the model training, there are two losses involved: the binary cross-entropy loss $\mathcal{L}_{exist}$ for estimating the existence of a relation using the relation existence estimation query in the relation query transformer, and the cross-entropy loss $\mathcal{L}_{LM}$ consistent with language model training used by the multimodal relation decoder. The total loss is: $\mathcal{L} = \lambda\mathcal{L}_{exist} + \mathcal{L}_{LM}$, where λ is a weight factor.

5 Experiments

5.1 Datasets

Panoptic Scene Graph (PSG) dataset [36] is constructed based on the COCO dataset [2, 22], consisting of 48,749 annotated images: 46,563 for training and 2,186 for testing. It encompasses 80 "thing" object categories [22] and 53 "stuff" object categories [2], as well as 56 relation categories.

Visual Genome (VG) dataset [17] is a widely used dataset in the SGG task. To further validate our method, we follow previous works [5, 38] and test our method on the VG-150 variant, which contains 150 object categories and 50 relation categories.

5.2 Tasks and metrics

Tasks. In both PSG and SGG, there are three distinct subtasks: Predicate Classification (PredCls), Scene Graph Classification (SGCls), and Scene Graph Detection (SGDet) [34]. In PredCls, the categories and locations of objects within the image are given, and only the relation categories between the objects need to be predicted. SGCls requires predicting both the categories of objects and the relations between them, given the locations of objects within the image. SGDet requires the simultaneous prediction of object categories, locations, and relations between objects, based on the given image. In this paper, we focus on the PredCls and SGDet subtasks. The PredCls excludes the influence of segmentation performance and only compares the relation prediction performance, while the SGDet considers the combined results for both object segmentation and relation prediction.

Furthermore, to validate our method's capability in open-set relation prediction, we divide the dataset into base relations and novel relations at a ratio of 7:3. For the PSG dataset, please refer to the supplementary material for the specific division method. For the division of the VG dataset, we follow the practice in previous works [5, 38]. In the open-set scenarios, our model is trained with data only from base relations and tested on both base and novel relations. It is noteworthy that the test sets for open-set and closed-set are same.
Metrics. Following previous works [36, 47], we use Recall@K (R@K) and mean Recall@K (mR@K) as our evaluation metrics. Additionally, in open-set scenarios, we also report the R@K and mR@K metrics for base and novel relations.

5.3 Implementation details

In our experiments, we utilize the pretrained OpenSeeD [44] as the open-set object segmenter. The patch size p of the patchify module is set to 8. Within the relation query transformer, the length E of the pair feature extraction query is 32, and the threshold θ used to filter subject-object pairs is set to 0.35. In the multimodal relation decoder, we employ the decoder of BLIP-2 [18]. During model training, the weight factor λ for the loss is set to 10. We adopt the same data augmentation strategies as in previous methods [36,47]. To train our model, we use the AdamW [26] optimizer with a learning rate of $1e^{-4}$ and a weight decay of $5e^{-2}$. Our model is trained for a total of 12 epochs, reducing the learning rate to $1e^{-5}$ at the 8_{th} epoch. The experimental platform uses four A100 GPUs. Note that during training we freeze the parameters of the object segmenter and multimodal relation decoder but only train the proposed RelQ-Former.

5.4 Comparison to the state of the art

PSG dataset. Tab. 1 consists of two parts, comparing the performance of our method against previous methods in closed-set and open-set scenarios under the subtasks of predicate classification and scene graph detection. For the first part, in the closed-set scenario, our method significantly surpasses previous methods. For the predicate classification subtask, only methods that predict segmentation and relation separately are applicable [36], and our method achieves a substantial improvement compared to previous best results, for instance, a 26.6% increase in R@100 and a 25.0% increase in mR@100. For scene graph detection, our method also achieves a significant increase of 9.0% over the best previous method [47] in R@100 and a 17.0% increase in mR@100. This indicates that in the closed-set scenario, our method demonstrates significant performance improvements. For the second part, in the open-set scenario, we train the model only on base relations and test it on all relations. Notably, for predicate classification, our method even outperforms previous methods trained on all relations, which demonstrates the superiority of our method. For example, compared with previous best results, we achieve a 39% improvement in R@100 and a 7.2% improvement in mR@100. For the scene graph detection subtask, our method with judgement instruction remains very competitive compared to [36] trained on all relations.

Table 1: Comparison between our OpenPSG and other methods on the PSG dataset in both closed-set and open-set scenarios. Our method shows superior performance compared to all previous methods.

	Predicate Classification			Scene Graph Detection		
Method	R/mR@20	R/mR@50	R/mR@100	R/mR@20	R/mR@50	R/mR@100
	Train on all relations (closed-set)					
IMP [34]	31.9/9.6	36.8/10.9	38.9/11.6	16.5/6.5	18.2/7.1	18.6/7.2
Motifs [43]	44.9/20.2	50.4/22.1	52.4/22.9	20.0/9.1	21.7/9.6	22.0/9.7
VCTree [30]	45.3/20.5	50.8/22.6	52.7/23.3	20.6/9.7	22.1/10.2	22.5/10.2
GPSNet [23]	31.5/13.2	39.9/16.4	44.7/18.3	17.8/7.0	19.6/7.5	20.1/7.7
PSGTR [36]	– / –	– / –	– / –	28.4/16.6	34.4/20.8	36.3/22.1
PSGFormer [36]	– / –	– / –	– / –	18.0/14.8	19.6/17.0	20.1/17.6
ADTrans [19]	– /29.0	– /36.2	– /38.8	26.0/26.4	29.6/29.7	30.0/30.0
PairNet [33]	– / –	– / –	– / –	29.6/24.7	35.6/28.5	39.6/30.6
HiLo [47]	– / –	– / –	– / –	34.1/23.7	40.7/30.3	43.0/33.1
OpenPSG	**55.1**/**39.2**	**70.6**/**53.8**	**79.3**/**63.8**	**38.1**/**32.3**	**46.8**/**40.9**	**52.0**/**50.1**
	Train on base relations (open-set)					
OpenPSG	45.1/29.1	55.5/38.7	61.5/46.0	25.9/20.9	31.6/24.0	36.7/25.4

Table 2: Comparison between our OpenPSG and other methods on the VG dataset in both closed-set and open-set scenarios with predicate classification subtask.

	Train on all relations (closed-set)		Train on base relations (open-set)	
Method	R/mR@50	R/mR@100	R/mR@50	R/mR@100
Motifs [43]	65.2/15.9	67.1/17.2	– / –	– / –
VCTree [30]	66.4/16.8	68.1/19.4	– / –	– / –
Cacao+Epic [38]	– /39.0	– /40.8	– /16.5	– /21.8
OvSGTR [5]	36.4/ –	42.4/ –	22.9/ –	26.7/ –
OpenPSG	**60.2**/**45.8**	**71.4**/**50.3**	**25.7**/**21.5**	**30.6**/**27.2**

VG dataset. To further validate our method on the VG dataset, we compare to two closed-set SGG methods [30, 43] and two recent open-set SGG works [5, 38]. Since our method relies on an object segmentation model while [5, 30, 38, 43] rely on an object detector, for a fair comparison, we only present the results for predicate classification subtask. Tab. 2 shows the results in both closed-set and open-set scenarios. In the closed-set scenario, compared with previous closed-set methods [30, 43], our method only performs a few points lower on R@50, yet yields significantly better results on the other metrics. In addition, compared with previous best open-set methods [5, 38], our method improves by 29.0% in R@100 and by 9.5% in mR@100. In open-set scenario, our method improves upon [5, 38] by 3.9% in R@100 and by 5.4% in mR@100. These results demonstrate the effectiveness of our method in both closed-set and open-set relation prediction.

Table 3: Ablation study of comparison of different segmenters.

	Predicate Classification				Scene Graph Detection		
Segmenter	PQ	R/mR@20	R/mR@50	R/mR@100	R/mR@20	R/mR@50	R/mR@100
OpenSeeD [44]	**55.1**	**55.1/39.2**	**70.6/53.8**	**79.3/63.8**	**38.1/32.3**	**46.8/40.9**	**51.9/50.1**
Mask2Former [6]	51.7	54.8/39.1	70.4/52.9	78.9/63.7	36.1/30.3	44.8/38.4	48.4/47.8

5.5 Ablation study

To validate the effectiveness of each module in our method, we conduct ablation studies, using the model trained on all relations with judgement instruction for relation prediction as the baseline unless otherwise specified. To eliminate interference, we validate the object segmenter and modules in RelQ-Former under closed-set settings, and test two types of instructions in the multimodal relation decoder under open-set settings.

Different segmenters. Our method is based on a pretrained segmenter. To verify the impact of different segmenters on model performance, we experiment with two options: the closed-set Mask2Former [6] and the open-set OpenSeeD [44]. As shown in Tab. 3, OpenSeeD outperforms Mask2Former on the Panoptic Quality (PQ) [16] metric (55.1 *vs*. 51.7) in PSG test set in closed-set senario. For the predicate classification subtask, the results show that using OpenSeeD is slightly better than using Mask2Former, with R@100 higher by 0.4% and mR@100 by 0.1%. For the scene graph detection subtask, OpenSeeD is a better segmenter than Mask2Former, with an increase of 3.5% in R@100 and 2.3% in mR@100, due to its superior object segmentation capability (see PQ in Tab. 3).

Subject-object pair features extraction. To validate the effectiveness of the RelQ-Former in extracting subject-object pair features via the attention mechanism, we design an experiment by extracting subject-object pair features using mask pooling. Specifically, for mask pooling, we derive object features by applying mask pooling to their respective positions in the visual features and then concatenate these to form the subject-object pair features. As shown in Tab. 4, it demonstrates that our attention mechanism outperforms mask pooling based pair feature extraction method by 5.2% in R@100 and by 4.7% in mR@100. This indicates that our method using RelQ-Former can better focus the extracted features on the interactions between objects, thereby enhancing the performance for relation prediction.

Selector in RelQ-Former. To validate the efficacy of the selector, we set θ to 0, meaning that during inference, we predict relations for all subject-object pairs. As shown in Tab. 4, we find that when we set θ to 0, the model performance is approximately the same. However, under these conditions, the model takes 20 times longer to run on the whole PSG test set. This indicates that our selector allows for a significant improvement in computational efficiency with a small sacrifice in performance. For more details, please refer to supplementary material.

Relation Existence Loss. To evaluate the impact of the relation existence estimation loss (Sec. 4.4) on the model, we set the training λ to 0, thereby

Table 4: Ablation study of comparison of different designs in RelQ-Former.

Method	Predicate Classification					
	R@20	mR@20	R@50	mR@50	R@100	mR@100
RelQ-Former	**55.1**	**39.2**	**70.6**	**53.8**	**79.3**	**63.8**
Change to Mask Pooling	51.5	36.3	65.9	49.3	74.1	59.1
Set $\theta = 0$ in Selector	56.8	40.7	70.9	54.0	79.6	63.9
Set $\lambda = 0$ in Loss	54.9	38.7	69.8	53.6	78.5	63.2

Table 5: Ablation study of different instruction types in multimodal relation decoder under various base-to-novel relation ratios. OpenPSG-G: using generation instruction. OpenPSG-J: using judgement instruction.

base:novel	OpenPSG-G			OpenPSG-J		
	R/mR@20	R/mR@50	R/mR@100	R/mR@20	R/mR@50	R/mR@100
7:3	40.1/24.5	49.4/32.4	57.1/36.8	45.1/29.1	55.5/38.7	61.5/46.0
6:4	34.9/21.3	46.3/29.2	52.9/32.0	40.6/25.1	53.9/35.3	57.0/43.5
5:5	25.4/14.0	38.6/23.2	40.1/24.2	36.4/19.2	48.9/29.3	50.3/40.8
4:6	19.8/11.2	32.9/14.3	33.2/18.8	29.8/17.3	40.9/21.6	44.3/34.7
3:7	13.0/10.7	17.6/13.0	18.5/14.4	22.8/15.7	28.3/19.7	35.1/23.7

removing this loss for model training. We discover that this loss not only aids in training a relation existence classifier, but also has beneficial effect on the model. As shown in Tab. 4, setting λ to 0 leads to a decrease in R@100 by 0.8% and in mR@100 by 0.6%.

Analysis of instruction types in multimodal relation decoder. To further analyze the two types of instructions, generation instruction and judgement instruction of open-set relation prediction in our multimodal relation decoder (Sec. 4.3), we conduct a series of experiments by adjusting the proportion of novel relation categories, conducting tests with base:novel ratios of 7:3, 6:4, 5:5, 4:6, and 3:7, to validate the performance, and results shown in Tab. 5. First, under the same base:novel ratio, the method using judgement instruction (OpenPSG-J) consistently outperforms the one using generation instruction (OpenPSG-G). For example, at a base:novel ratio of 7:3, OpenPSG-J is 4.4% higher than OpenPSG-G in R@100, and is 9.2% higher in mR@100. Second, the results indicate that as the proportion of novel relations increases, the performance of both OpenPSG-G and OpenPSG-G gradually decreases. For OpenPSG-G, R@100 decreases from 57.1% at a base:novel ratio of 7:3 to 18.5% at a base:novel ratio of 3:7, a drop of 38.6%. OpenPSG-J's R@100 decreases by 26.4%, which is less than the decrease for OpenPSG-G, indicating that OpenPSG-J has a stronger capability for relation prediction in an open world.

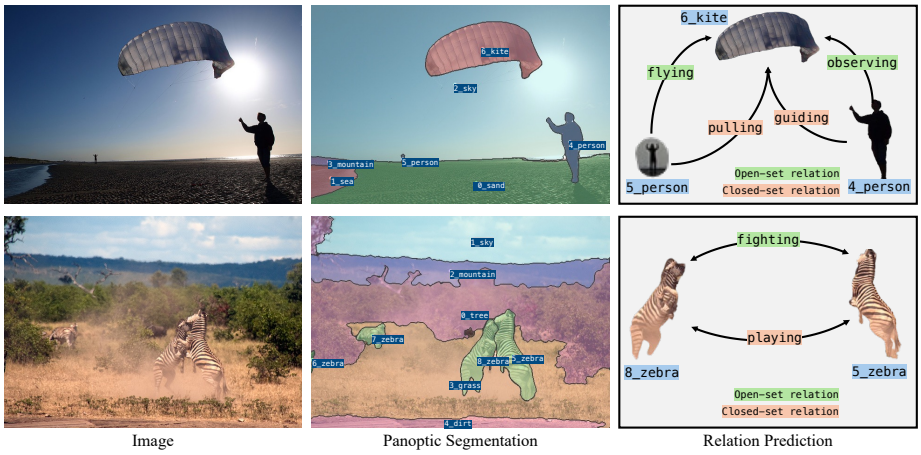

Fig. 3: Visualization results produced by our OpenPSG. The left image is the input to our OpenPSG, the middle one displays the panoptic segmentation result, and the right one shows the predicted relations between objects.

5.6 Visualization

As shown in Fig. 3, our OpenPSG method can predict relations not defined in the dataset, such as "*flying*", "*observing*", and "*fighting*", which qualitatively demonstrates the excellent open-set relation prediction capability of our method.

6 Conclusion

In this paper, we propose the open-set PSG task and introduce the OpenPSG to accomplish open-set relation prediction. With the help of large multimodal models, our method employs an autoregressive approach to predict open-set relations. Additionally, we have developed a relation query transformer which contains pair feature extraction and relation existence estimation queries, one for extracting features of subject-object pairs, the other for predicting the existence of relations between them to filter out irrelevant pairs. Furthermore, we design generation and judgement instructions to enable the multimodal relation decoder to predict open-set relations. Extensive experiments demonstrate that our method achieves excellent performance in open-set relation prediction. In the future, we plan to employ model distillation to reduce the model size, thus enhancing the practicality in various real-world scenarios while ensuring its ability to predict open-set relations.

Acknowledgments

The authors would like to thank Prof. Tomasz Radzik for helpful discussions. Computing resources provided by King's Computational Research, Engineering and Technology Environment (CREATE). This work was supported by the Fundamental Research Funds for the Central Universities.

References

1. Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F.L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., et al.: Gpt-4 technical report. arXiv preprint arXiv:2303.08774 (2023) 2, 3, 4
2. Caesar, H., Uijlings, J., Ferrari, V.: Coco-stuff: Thing and stuff classes in context. In: CVPR (2018) 9
3. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: ECCV. pp. 213–229 (2020) 3
4. Chen, S., Jin, Q., Wang, P., Wu, Q.: Say as you wish: Fine-grained control of image caption generation with abstract scene graphs. In: CVPR (2020) 1
5. Chen, Z., Wu, J., Lei, Z., Zhang, Z., Chen, C.: Expanding scene graph boundaries: Fully open-vocabulary scene graph generation via visual-concept alignment and retention. arXiv preprint arXiv:2311.10988 (2023) 3, 4, 9, 10, 11
6. Cheng, B., Misra, I., Schwing, A.G., Kirillov, A., Girdhar, R.: Masked-attention mask transformer for universal image segmentation. In: CVPR. pp. 1290–1299 (2022) 3, 5, 12
7. Dai, J., He, K., Sun, J.: Convolutional feature masking for joint object and stuff segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3992–4000 (2015) 6
8. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020) 5
9. Du, Y., Wei, F., Zhang, Z., Shi, M., Gao, Y., Li, G.: Learning to prompt for open-vocabulary object detection with vision-language model. In: CVPR. pp. 14084–14093 (2022) 2, 4
10. Gao, L., Wang, B., Wang, W.: Image captioning with scene-graph based semantic concepts. In: ICMLC (2018) 1
11. Ghiasi, G., Gu, X., Cui, Y., Lin, T.Y.: Scaling open-vocabulary image segmentation with image-level labels. In: ECCV. pp. 540–557. Springer (2022) 2
12. He, T., Gao, L., Song, J., Li, Y.F.: Towards open-vocabulary scene graph generation with prompt-based finetuning. In: ECCV (2022) 3
13. Hildebrandt, M., Li, H., Koner, R., Tresp, V., Günnemann, S.: Scene graph reasoning for visual question answering. arXiv preprint arXiv:2007.01072 (2020) 1
14. Joshi, M., Levy, O., Weld, D.S., Zettlemoyer, L.: Bert for coreference resolution: Baselines and analysis. arXiv preprint arXiv:1908.09091 (2019) 4
15. Kan, X., Cui, H., Yang, C.: Zero-shot scene graph relation prediction through commonsense knowledge integration. In: Machine Learning and Knowledge Discovery in Databases. Research Track: European Conference, ECML PKDD 2021, Bilbao, Spain, September 13–17, 2021, Proceedings, Part II 21. pp. 466–482. Springer (2021) 3

16. Kirillov, A., He, K., Girshick, R., Rother, C., Dollár, P.: Panoptic segmentation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 9404–9413 (2019) 12
17. 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) 9
18. Li, J., Li, D., Savarese, S., Hoi, S.: Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. arXiv preprint arXiv:2301.12597 (2023) 2, 4, 10
19. Li, L., Ji, W., Wu, Y., Li, M., Qin, Y., Wei, L., Zimmermann, R.: Panoptic scene graph generation with semantics-prototype learning. arXiv preprint arXiv:2307.15567 (2023) 1, 3, 11
20. Liang, F., Wu, B., Dai, X., Li, K., Zhao, Y., Zhang, H., Zhang, P., Vajda, P., Marculescu, D.: Open-vocabulary semantic segmentation with mask-adapted clip. In: CVPR. pp. 7061–7070 (2023) 2
21. Lin, C., Sun, P., Jiang, Y., Luo, P., Qu, L., Haffari, G., Yuan, Z., Cai, J.: Learning object-language alignments for open-vocabulary object detection. arXiv preprint arXiv:2211.14843 (2022) 2
22. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: ECCV (2014) 9
23. Lin, X., Ding, C., Zeng, J., Tao, D.: Gps-net: Graph property sensing network for scene graph generation. In: CVPR (2020) 11
24. Liu, H., Li, C., Li, Y., Lee, Y.J.: Improved baselines with visual instruction tuning. arXiv preprint arXiv:2310.03744 (2023) 4, 6
25. Liu, H., Li, C., Wu, Q., Lee, Y.J.: Visual instruction tuning. NeurIPS **36** (2024) 2, 4
26. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. In: ICLR (2019) 10
27. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: ICML. pp. 8748–8763 (2021) 2, 4
28. Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., Liu, P.J.: Exploring the limits of transfer learning with a unified text-to-text transformer. The Journal of Machine Learning Research **21**(1), 5485–5551 (2020) 4
29. Singh, K.P., Salvador, J., Weihs, L., Kembhavi, A.: Scene graph contrastive learning for embodied navigation. In: ICCV (2023) 1
30. Tang, K., Zhang, H., Wu, B., Luo, W., Liu, W.: Learning to compose dynamic tree structures for visual contexts. In: CVPR (2019) 11
31. Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., et al.: Llama: Open and efficient foundation language models. arXiv preprint arXiv:2302.13971 (2023) 4
32. Wang, C., Liu, X., Song, D.: Language models are open knowledge graphs. arXiv preprint arXiv:2010.11967 (2020) 4
33. Wang, J., Wen, Z., Li, X., Guo, Z., Yang, J., Liu, Z.: Pair then relation: Pair-net for panoptic scene graph generation. arXiv preprint arXiv:2307.08699 (2023) 1, 2, 3, 11
34. Xu, D., Zhu, Y., Choy, C.B., Fei-Fei, L.: Scene graph generation by iterative message passing. In: CVPR (2017) 9, 11

35. Yang, J., Wang, C., Liu, Z., Wu, J., Wang, D., Yang, L., Cao, X.: Focusing on flexible masks: A novel framework for panoptic scene graph generation with relation constraints. In: ACM MM. pp. 4209–4218 (2023) 1
36. Yang, J., Ang, Y.Z., Guo, Z., Zhou, K., Zhang, W., Liu, Z.: Panoptic scene graph generation. In: ECCV. pp. 178–196 (2022) 1, 3, 9, 10, 11
37. Yao, L., Han, J., Liang, X., Xu, D., Zhang, W., Li, Z., Xu, H.: Detclipv2: Scalable open-vocabulary object detection pre-training via word-region alignment. In: CVPR. pp. 23497–23506 (2023) 2
38. Yu, Q., Li, J., Wu, Y., Tang, S., Ji, W., Zhuang, Y.: Visually-prompted language model for fine-grained scene graph generation in an open world. arXiv preprint arXiv:2303.13233 (2023) 3, 9, 10, 11
39. Yu, Q., Shen, X., Chen, L.C.: Towards open-ended visual recognition with large language model. arXiv preprint arXiv:2311.08400 (2023) 2, 4, 8
40. Yu, X., Chen, R., Li, J., Sun, J., Yuan, S., Ji, H., Lu, X., Wu, C.: Zero-shot scene graph generation with knowledge graph completion. In: 2022 IEEE International Conference on Multimedia and Expo (ICME). pp. 1–6. IEEE (2022) 3
41. Yue, K., Chen, B.C., Geiping, J., Li, H., Goldstein, T., Lim, S.N.: Object recognition as next token prediction. arXiv preprint arXiv:2312.02142 (2023) 8
42. Zang, Y., Li, W., Zhou, K., Huang, C., Loy, C.C.: Open-vocabulary detr with conditional matching. In: ECCV. pp. 106–122. Springer (2022) 2
43. Zellers, R., Yatskar, M., Thomson, S., Choi, Y.: Neural motifs: Scene graph parsing with global context. In: CVPR (2018) 11
44. 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) 2, 5, 10, 12
45. Zhang, Y., Pan, Y., Yao, T., Huang, R., Mei, T., Chen, C.W.: Learning to generate language-supervised and open-vocabulary scene graph using pre-trained visual-semantic space. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2915–2924 (2023) 3
46. Zhao, C., Shen, Y., Chen, Z., Ding, M., Gan, C.: Textpsg: Panoptic scene graph generation from textual descriptions. In: ICCV. pp. 2839–2850 (2023) 1, 3
47. Zhou, Z., Shi, M., Caesar, H.: Hilo: Exploiting high low frequency relations for unbiased panoptic scene graph generation. arXiv preprint arXiv:2303.15994 (2023) 1, 2, 3, 10, 11
48. Zhou, Z., Shi, M., Caesar, H.: Vlprompt: Vision-language prompting for panoptic scene graph generation. arXiv preprint arXiv:2311.16492 (2023) 1, 2, 3
49. Zhu, D., Chen, J., Shen, X., Li, X., Elhoseiny, M.: Minigpt-4: Enhancing vision-language understanding with advanced large language models. arXiv preprint arXiv:2304.10592 (2023) 2