International Journal of Computer Vision (2025) 133:8006–8021
https://doi.org/10.1007/s11263-025-02564-7

VLPrompt-PSG: Vision-Language Prompting for Panoptic Scene Graph Generation

Zijian Zhou[1] · Holger Caesar[2] · Qijun Chen[3] · Miaojing Shi[3,4]

Received: 9 December 2024 / Accepted: 6 August 2025 / Published online: 23 August 2025

Abstract

Panoptic scene graph generation (PSG) aims at achieving a comprehensive image understanding by simultaneously segmenting objects and predicting relations among objects. However, the long-tail problem among relations leads to unsatisfactory results in real-world applications. Prior methods predominantly rely on vision information or utilize limited language information, such as object or relation names, thereby overlooking the utility of language information. Leveraging the recent progress in Large Language Models (LLMs), we propose to use language information to assist relation prediction, particularly for rare relations. To this end, we propose the **V**ision-**L**anguage **Prompt**ing (**VLPrompt**) model, which acquires vision information from images and language information from LLMs. Then, through a prompter network based on attention mechanism, it achieves precise relation prediction. Our extensive experiments show that VLPrompt significantly outperforms previous state-of-the-art methods on the PSG dataset, proving the effectiveness of incorporating language information and alleviating the long-tail problem of relations. Code is available at https://github.com/franciszzj/VLPrompt.

Keywords Panoptic scene graph generation · large language models · visual relation detection

1 Introduction

Panoptic scene graph generation (PSG) (Yang et al., 2022) extends scene graph generation (SGG) (Lu et al., 2016) by incorporating panoptic segmentation (Kirillov et al., 2019) to capture richer and more detailed representations of images, including both "thing" (Lin et al., 2014) and "stuff" (Caesar et al., 2018) object classes. PSG constructs a directed graph to represent an image, where nodes signify objects and edges capture the relations between objects. As a bridge between vision and language (Zhu et al., 2022), PSG has a multitude of downstream applications such as visual question answering (Hildebrandt et al., 2020), image captioning (Gao et al., 2018; Chen et al., 2020), and visual reasoning (Aditya et al., 2018; Shi et al., 2019); furthermore, it can also benefit relevant fields like embodied navigation (Singh et al., 2023) and robotic action planning (Amiri et al., 2022).

Notwithstanding, the current performance of PSG (Yang et al., 2022; Zhou et al., 2023b; Wang et al., 2024; Li et al., 2024b; Hayder & He, 2024; Lorenz et al., 2024; Zhou et al., 2025) remains unsatisfactory, limiting its downstream applications. The essential reason lies in the severe long-tail problem in relation categories: for instance, in the PSG dataset (Yang et al., 2022), the top three most frequent relation categories account for over 50% of entire samples, with numerous rare relations appearing less than 1%. PSG models thus struggle to accurately predict these rare relations.

Recent methods (Lyu et al., 2022; Dong et al., 2022; Li et al., 2022b; Zhang et al., 2022; Deng et al., 2022; Zhou et al., 2023b; Yu et al., 2023; Hayder & He, 2024; Im et al., 2024; Li et al., 2024a; Lorenz et al., 2024) have made progress in addressing the long-tail problem, mainly exploiting the strength of vision information for relation prediction, whilst overlooking the language information in PSG. The integration of language information is however important to provide additional common sense knowledge for objects and

Communicated by Kaiyang Zhou.

✉ Miaojing Shi
mshi@tongji.edu.cn

[1] Department of Informatics, King's College London, London, United Kingdom

[2] Intelligent Vehicles Lab, Delft University of Technology, Delft, The Netherlands

[3] College of Electronic and Information Engineering, Tongji University, Shanghai, China

[4] Shanghai Institute of Intelligent Science and Technology, Tongji University, Shanghai, China

their relations. For example, in the two images of Fig. 1, the relation between the person and the elephant is *cleaning*. In the image 2 of Fig. 1, where the person is on the elephant's back *cleaning* it, this scenario can easily lead previous vision-only models to classify the relation as *riding*. In contrast, our vision-language model can utilize language information like "a person cleans an elephant using brushes on the back of the elephant", thus precisely predicting the relation as *cleaning*.

In SGG task, some methods (He et al., 2022; Dong et al., 2022) have recognized the importance of incorporating language information besides vision. However, the way language information is utilized in these works is limited to category names of objects or relations, providing no further context and hence not fully addressing the long-tail problem. The same observation goes to methods like Gu et al. (2019); Zareian et al. (2020), which integrate knowledge graphs into the SGG task. With the rapid development of Large Language Models (LLMs) (Achiam et al., 2023; Touvron et al., 2023b), acquiring richer language information, instead of merely the concepts of objects or relations, becomes much easier than before.

In this paper, we introduce a novel **V**ision-**L**anguage **Prompt**ing (**VLPrompt**) model, which leverages rich language information from LLMs to help predict relations between objects in visual images. The language information serves as a powerful supplement to relation prediction, especially for rare ones. Our model comprises three parts, forming a complete system implementation of our proposed idea. The first is the *vision feature extractor*, where we process the input image with a panoptic segmentation network adapted from Mask2Former (Cheng et al., 2022a) to extract features of different objects. We pair and concatenate these object features and integrate their corresponding spatial information to form the vision prompting features. In contrast, in the second part, the *language feature extractor*, we employ the chain-of-thought technique (Wei et al., 2022) to design various prompts, aiming to stimulate LLMs to propose context-rich language information for potential relations between a subject-object pair or judge a specific subject-relation-object triplet. These two functions are realized via the carefully designed relation proposer prompt and relation judger prompt. Subsequently, these language descriptions are transformed into language features using a pre-trained text encoder. Finally, in the third part, the *vision-language prompter*, we design a novel dual attention-based prompter network to facilitate the interaction between vision features and the two complimentary language features respectively, resulting into two sets of relation predictions. They are combined via a MLP-based gating network to take the strength of both for final relation prediction. The whole VLPrompt is trained end-to-end.

Extensive experiments on the PSG dataset (Yang et al., 2022) demonstrate that our VLPrompt significantly enhances the PSG performance, especially in predicting rare relations, highlighting the importance of integrating language information from LLMs to support relation prediction in the PSG task.

2 Related Work

2.1 Scene Graph Generation

Scene graph generation (SGG) (Lu et al., 2016) is a crucial task in scene understanding and has garnered widespread attention in the computer vision community. In recent years, numerous methods (Xu et al., 2017; Zellers et al., 2018; Tang et al., 2019; Lin et al., 2020; Li et al., 2022a; Shit et al., 2022; Zhang et al., 2022; Yu et al., 2023; Hayder & He, 2024; Im et al., 2024; Li et al., 2024a, 2025) have achieved notable progress. Various model architectures have been proposed, such as intricately designed message passing structures (Li et al., 2017a; Dai et al., 2017; Li et al., 2017b; Zellers et al., 2018; Gu et al., 2019; Hu et al., 2019), attention-based networks (Zheng et al., 2019; Qi et al., 2019), tree-based networks (Zhang et al., 2017; Hung et al., 2020), DETR-based networks (Li et al., 2022a; Shit et al., 2022; Cong et al., 2023) and transformer-based networks (Hayder & He, 2024; Im et al., 2024). Specifically, to address the long-tail problem, some methods enhance the prediction accuracy of rare relations through data re-sampling (Li et al., 2021) and loss re-weighting (Kang & Yoo, 2023). Relevant techniques that have been developed include constructing enhanced datasets (Zhang et al., 2022; Yu et al., 2023), grouping relations for training (Dong et al., 2022), constructing multi-stage hierarchical training (Deng et al., 2022), and designing de-bias loss functions (Yu et al., 2021; Kang & Yoo, 2023). Most methods leverage images as sole inputs. Besides, some methods (Lu et al., 2016; Hwang et al., 2018; Liao et al., 2019; Zhang et al., 2019a; Dupty et al., 2020; Zhong et al., 2021; Ye & Kovashka, 2021) have begun exploring language information or knowledge graphs in SGG; specifically, the explored language information is so far confined to basic language concepts of objects or relations.

2.2 Panoptic Scene Graph Generation

Panoptic scene graph generation (PSG) (Yang et al., 2022) has emerged as a novel task in scene understanding in recent years. Unlike SGG (Lu et al., 2016), PSG employs panoptic segmentation (Kirillov et al., 2019) instead of bounding boxes to represent objects, enabling a more comprehensive understanding. Similar to SGG methods, current methods in PSG (Yang et al., 2022; Zhou et al., 2023b; Wang et al., 2024; Li et al., 2024b; Hayder & He, 2024; Lorenz et al., 2024) also

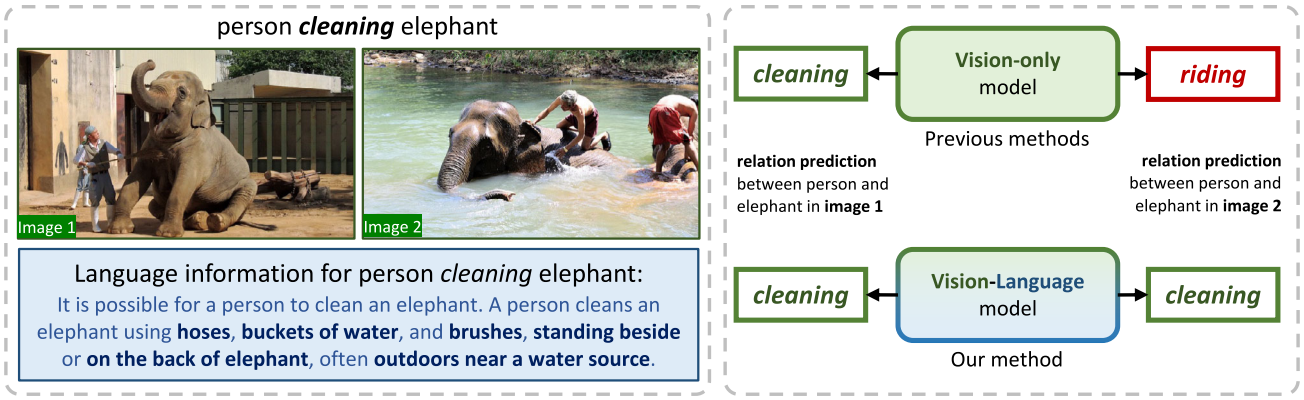

Fig. 1 Comparison between previous PSG methods and ours. **Left**: Images of "person *cleaning* elephant" in two different scenes, accompanied by snippets of descriptions about "person *cleaning* elephant" obtained from LLMs. **Right**: Previous vision-only models can predict the *cleaning* relation between the person and elephant in image 1, but often classify image 2's relation as *riding* due to the person's position on the back of the elephant. Our vision-language model, enriched with language information, precisely identifies the *cleaning* relation in both images.

mainly rely on the image input and do not utilize any language information. For instance, PSGTR (Yang et al., 2022) build a baseline PSG model by adding a relation prediction head to DETR (Carion et al., 2020). PSGFormer (Yang et al., 2022) advances PSGTR by separately modeling objects and relations in two transformer decoders and introducing an interaction mechanism. Recently, HiLo (Zhou et al., 2023b) addresses the long-tail problem by specializing different network branches in learning both high and low frequency relations. PairNet (Wang et al., 2024) develops a novel framework using a pair proposal network to filter sparse pairwise relations, improving PSG performance. TextPSG (Zhao et al., 2023) leverages language information for model training but adopts a semi-supervised approach without applying language information to a fully supervised method, resulting in poor performance. DSGG (Hayder & He, 2024) builds a transformer-based network with graph-aware queries to predict unbiased relations. In addition, some works (Yang et al., 2023b, a) extend panoptic scene graph generation to video (Yang et al., 2023b) and 4D (Yang et al., 2023a) scenes. In contrast to previous methods that rely solely on vision as input, we propose a fully supervised PSG method that leverages both vision and language inputs.

2.3 Large Language Models for Vision Tasks

LLMs have led to large improvements in natural language processing tasks (Min et al., 2023). They are normally trained on extensive text corpora by learning to autoregressively predict the next word, hence encapsulate a broad spectrum of common sense knowledge of linguistic patterns, cultural norms, and basic worldly facts. Prominent examples of LLMs include GPT series (Achiam et al., 2023), Llama series (Touvron et al., 2023a, b), Gemini (Team et al., 2023), and Claude (Anthropic, 2023), with Llama series being open-source publicly available. Given the extensive common sense information contained in LLMs, some researchers start to propose multimodal sockets to LLMs (Zhang et al., 2024a; Wu et al., 2024) and apply them to various vision tasks, such as recognition (Huang et al., 2023; Wang et al., 2023), detection (Tang et al., 2023; Zhang et al., 2023; Qi et al., 2024), segmentation (Lai et al., 2024; Zhou et al., 2023; Xia et al., 2024), visual question answering (Liu et al., 2023; Xenos et al., 2023), image reasoning (Chen et al., 2023; Zhang et al., 2024b) and robotic navigation (Tsai et al., 2023; Shah et al., 2023); nevertheless, there are no such models specifically designed for panoptic scene graph generation so far. In contrast, our method designs various prompts to stimulate LLMs to elicit rich language information to enhance relation prediction.

3 Method

In this section, we introduce our method, **VLPrompt**. Given an image $\mathcal{I} \in \mathbb{R}^{H\times W\times 3}$ and language description $\mathcal{T}$ generated from LLMs, we extract vision and language features from them to predict panoptic scene graph $\mathcal{G} = \{\mathcal{O}, \mathcal{R}\}$, *i.e.*, $\mathcal{G} = \text{VLPrompt}(\mathcal{I}, \mathcal{T})$. In $\mathcal{G}$:

- $\mathcal{O} = \{o_i\}_{i=1}^{N}$ signifies N objects segmented from the image $\mathcal{I}$. Each object is defined by $o_i = \{c, m\}$, where c belongs to one of the predefined C object categories, and m is a binary mask in $\{0, 1\}^{H\times W}$ for this object.
- $\mathcal{R} = \{r_{i,j} \mid i, j \in \{1, 2, \ldots, N\}, i \neq j\}$ denotes relations with $r_{i,j}$ being the relation between o_i and o_j. Each

r belongs to one of the predefined K relation categories (or no relation), N is the number of objects in the image.

As shown in Fig. 2, our method comprises three main components: *vision feature extractor*, *language feature extractor*, and *vision-language prompter*. The *vision feature extractor* (Sec. 3.1) adapts from a segmentation network (*e.g.*, Mask2Former (Cheng et al., 2022a)) to predict object masks and form subject-object pairs for feature extraction, which result into vision prompting features. For the *language feature extractor* (Sec. 3.2), we generate different types of descriptions by leveraging the extensive common sense knowledge embedded in LLMs through carefully designed prompts, which is beneficial for relations of different frequencies. These descriptions are then converted into language prompting features using a text encoder. Next, the vision and language prompting features are fed into a *vision-language prompter* (Sec. 3.3), where vision prompting features interact with different types of language prompting features respectively, so as to take advantage of the complimentary language information to assist relation predictions. Finally, these relation predictions are combined by a relation fusion module to achieve the final relation prediction.

3.1 Vision Feature Extractor

Given an image $\mathcal{I}$, we first leverage Mask2Former (Cheng et al., 2022a) to produce N objects with masks. They are formed into $N \times (N-1)$ subject-object pairs by pairing any two distinct ones. The purpose of the vision feature extractor is to extract vision prompting feature for each subject-object pair, which includes visual features from the segmentation network itself as well as spatial features between the subject-object pairs. In this way, we can enhance the representations of the relations between subject-object pairs.

Subject-object visual features. We first obtain the features corresponding to each object. Considering that the output feature map by the pixel decoder in Mask2Former retains rich information of the image, we use mask pooling to obtain object features corresponding to the N objects from the feature map based on each object's mask m. Then, we pair and concatenate any two distinct object features to form $N \times (N-1)$ subject-object visual features F_V^{vi}.

Subject-object spatial features. To further enhance the representations of subject-object pairs, especially their spatial relations, we are inspired by Peyre et al. (2017) to encode the spatial features into the subject-object visual features. Specifically, given o_i and o_j corresponding to subject and object, we first derive their encompassing bounding boxes $b_i = [x_i, y_i, w_i, h_i]$ and $b_j = [x_j, y_j, w_j, h_j]$, where (x, y) is the center of the bounding box, and (w, h) are the width and height. Next we construct spatial features:

$$v(o_i, o_j) = [\frac{x_j - x_i}{\sqrt{w_i h_i}}, \frac{y_j - y_i}{\sqrt{w_i h_i}}, \sqrt{\frac{w_j h_j}{w_i h_i}}, \frac{b_i \cap b_j}{b_i \cup b_j}, \frac{w_i}{h_i}, \frac{w_j}{h_j}], \quad (1)$$

where $v(o_i, o_j)$ encodes the spatial relation between o_i and o_j, such as the ratio of their bounding box sizes, the overlap between two objects and the aspect ratio of each object. Then, we use a FC layer to expand the spatial features to the same dimension as F_V^{vi}, resulting in F_V^{sp}.

Finally, we apply a FC layer to the sum of F_V^{vi} and F_V^{sp} and output vision prompting features $F_V \in \mathbb{R}^{N \times (N-1) \times D_V}$, where D_V is the vision feature dimension.

3.2 Language Feature Extractor

The purpose of the language feature extractor is to leverage the extensive common sense knowledge embedded in LLMs for providing additional language information to the PSG task, which can mitigate the long-tail problem in relation prediction. To achieve this, we need to design various prompts to elicit outputs from LLMs. On one hand, LLMs can act as a relation proposer, suggesting possible relations between two objects, which often are frequently occurring relations in the real world. On the other hand, LLMs can also serve as a relation judger, given a subject-object pair and their specific relation, LLMs make judgment and provide reasoning for this relation. This allows detailed descriptions even for rare relations. Specifically, we design two types of prompts: relation proposer prompt (RP-Prompt) for proposing and explaining potential relations given a subject-object pair; relation judger prompt (RJ-Prompt) for judging and reasoning upon a specific subject-relation-object triplet. Below, we detail how to obtain RP- and RJ-language prompting features based on the generated descriptions.

RP-language prompting feature. For RP-language prompting feature, we stimulate LLMs to guess all possible relations between two given objects o_i and o_j, along with explanation for these relations. To achieve this, we utilize the chain-of-thought technique (Wei et al., 2022): we engage in a dialogue with an LLM (*e.g.*, GPT-4 (Achiam et al., 2023)), initially informing it to act as a relation proposer and defining the task. Then an example is provided to the LLM to clarify its role. We explicitly mention the predefined K relations in the prompt, guiding the LLM to propose from them. Finally, a certain subject-object pair (o_i and o_j) is given to the relation proposer prompt. By giving this prompt to the LLM, we obtain the description for potential relations between o_i and o_j. For predefined relations that are not proposed by the LLM, we would append a template phrase by the end of the descrip-

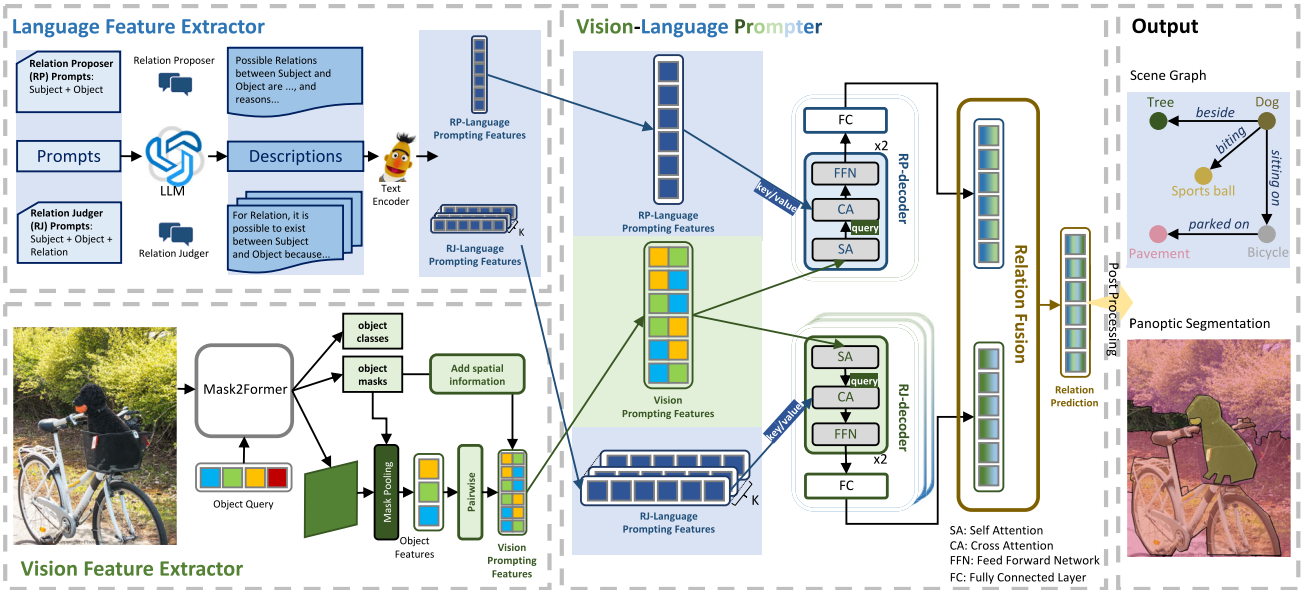

Fig. 2 The overall framework of VLPrompt, which comprises three components: the vision feature extractor, the language feature extractor and the vision-language prompter. We employ the language feature extractor to obtain LLM-based language information and integrate it into the final relation prediction through the vision-language prompter, enabling unbiased relation predictions

tion, such as "It is not likely for o_i and o_j to have relation r.", to ensure that the language description clearly distinguishes common from uncommon relations and comprehensively covers all relation categories. To encode the description into a feature interpretable by our model, we use a text encoder (*e.g.*, OpenAI Embeddings (Achiam et al., 2023)) to convert the description into the feature $F^{RP}_{L(i,j)} \in \mathbb{R}^{1\times D_L}$, namely RP-language prompting feature. This process consolidates all descriptions (each expressing multiple plausible relations for a given subject-object pair) into a single feature, allowing for a condensed reflection of the distinct attributes of common relations between subject and object.

RJ-language prompting feature. For RJ-language prompting feature, we design the relation judger prompt: we not only provide two objects o_i, o_j but also specify a relation r_k between them. By using the common sense knowledge, the LLM judges whether the relation r_k could plausibly exist between o_i and o_j and provides reason. Similar to above, we use the chain-of-thought technique (Wei et al., 2022) by telling the LLM that it serves as a relation judger; we first define the task, then give the example, and finally, the triplet (o_i-r_k-o_j) is provided. Following the same process as for RP-language prompting feature, we feed the relation judger prompt to the LLM to obtain the language description and leverage the text encoder (same as above) to encode it into the RJ-language prompting feature $F^{RJ}_{L(i,j,k)} \in \mathbb{R}^{1\times D_L}$. Different from the RP-language prompting feature, we encode each relation triplet into an individual feature, storing more detailed and subtle information for each relation, hence favouring rare relations.

By enumerating all C objects and K relations in the dataset, we derive all RP- and RJ-language prompting features. To further enhance the practicality of our method and avoid repeatedly invoking LLM at runtime, we store them in a database. Given an image with N objects, we retrieve two sets of features, $F^{RP}_L \in \mathbb{R}^{N\times(N-1)\times D_L}$ and $F^{RJ}_L \in \mathbb{R}^{N\times(N-1)\times K\times D_L}$ from the database. Note that for a specific relation r_k between all subject object pairs in this image, the RJ-language prompting feature is denoted as $F^{RJ}_{L(k)} \in \mathbb{R}^{N\times(N-1)\times D_L}$.

3.3 Vision-Language Prompter

To enable the vision prompting feature to predict relations from both macro and detailed perspectives, we let F_V interact with F^{RP}_L and F^{RJ}_L through two separate decoders, termed as RP-decoder and RJ-decoder, responsible for the interaction from F_V to F^{RP}_L and F^{RJ}_L, respectively. Each decoder contains two standard transformer decoder blocks (Vaswani et al., 2017), followed by a FC layer for relation prediction. The predictions from the two decoders are complementary: the RP-language prompting feature focuses on the condensed and distinct attributes of frequently occurring relations for a given subject-object pair; in contrast, the RJ-language prompting feature focuses on the detailed and subtle attributes of every possible relation (common or rare) for the subject-object pair. A relation fusion module consisting of a

gating network is thereby devised by the end to fuse the relation predictions from both decoders into the final one. Before feeding F_V, F_L^{RP}, and F_L^{RJ} into different decoders, we use a FC layer to transform their dimensions to a uniform dimension D. Below, we specify the RP-decoder, RJ-decoder, and the relation fusion module, respectively.

RP-decoder. The RP-decoder aims to utilize the F_L^{RP} to assist F_V in relation prediction, particularly for relations that are frequently encountered between o_i and o_j in the real world. In the first transformer decoder block, F_V is firstly fed into the self-attention layer, primarily aggregating the visual relational information in F_V. Afterwards, the self-attended F_V as query and the F_L^{RP} as key/value are engaged in the subsequent cross-attention layer, aggregating the common sense knowledge of potential relations between o_i and o_j into F_V. The output is further processed through a feed-forward network. In the second transformer decoder block, we repeat the aforementioned process. Finally, a fully connected layer is used to transform feature dimension D to the number of relations K, and a sigmoid function is applied afterwards to obtain $R^{RP} \in \mathbb{R}^{N\times(N-1)\times K}$.

RJ-decoder. The RJ-decoder aims to facilitate interaction between the RJ-language prompting feature F_L^{RJ} with the vision prompting feature F_V. Since F_L^{RJ} a group of individual language prompting feature for every relation triplet (o_i-r_k-o_j), F_V thus has the opportunity to interact with each relation's language representation independently, which can be particularly beneficial to rare relations. We conduct parallel interactions between F_V and the K triplet features contained in F_L^{RJ}. For each triplet, the interaction process between F_V and $F_{L(k)}^{RJ}$ is the same to that of the RP-decoder, except that the final FC layer is now only responsible for predicting the probability of certain relation between o_i and o_j. The FC layer is used to transform the feature dimension from D to 1. Finally, we concatenate the respective outputs to get the predictions for all K relations, a sigmoid function is applied over them to obtain $R^{RJ} \in \mathbb{R}^{N\times(N-1)\times K}$.

Relation Fusion. Upon obtaining R^{RP} and R^{RJ}, we aim to take the strength of both via a relation fusion module. We devise a gating network consisting of 3-layer MLP to generate two sets of weights, W^{RP} and W^{RJ}, each matching the shape of R^{RP} and R^{RJ}. We use the sum of F_V and F_L^{RP} as the input of the gating network, and output W^{RP}. For W^{RJ}, we use the sum of F_V and the mean of F_L^{RJ} along the relation dimension as input to the gating network. W^{RP} and W^{RJ} are used to element-wisely multiply with R^{RP} and R^{RJ} respectively and the final relation prediction R is a weighted combination:

$$R = W^{RP} \odot R^{RP} + W^{RJ} \odot R^{RJ}, \tag{2}$$

where $\odot$ is element-wise multiplication, and $R \in \mathbb{R}^{N\times(N-1)\times K}$.

Finally, the prediction R, combined with the object categories and masks predicted by the vision feature extractor, forms the panoptic scene graph $\mathcal{G}$.

3.4 Model Training

Our model training comprises two parts. The first part is the segmentation loss $\mathcal{L}_{seg}$ used in the vision feature extractor for panoptic segmentation, we simply follow the loss used in Cheng et al. (2022a). The second part is the relation loss. Since the same subject-object pair might have multiple relations, we use a binary cross-entropy loss (Su et al., 2022). To effectively train the vision-language prompter, we apply the relation loss separately to R^{RP}, R^{RJ}, and R and sum them up as the final relation loss, denoted by $\mathcal{L}_{rel}$. The final loss $\mathcal{L}$ is

$$\mathcal{L} = \lambda\mathcal{L}_{seg} + \mathcal{L}_{rel}, \tag{3}$$

where λ is the weighting coefficient. In the language feature extractor, we directly utilize pre-trained LLMs, thus eliminating the need for additional model training.

4 Experiments

4.1 Datasets

Panoptic Scene Graph (PSG) dataset (Yang et al., 2022). This is the first dataset dedicated to the PSG task, comprising 48,749 annotated images, including 2,186 test images and 46,563 training images. The dataset includes 80 "thing" (Lin et al., 2014) and 53 "stuff" categories (Caesar et al., 2018), as well as 56 relation categories.

Visual Genome (VG) dataset (Krishna et al., 2017). This dataset is widely used in the SGG task. To validate our method, we also conduct experiments on the VG dataset. Following previous works (Zellers et al., 2018; Chen et al., 2019), we use the VG-150 variant, which includes 150 object categories and 50 relation categories.

4.2 Tasks and metrics

Tasks. There are three subtasks for PSG and SGG tasks: Predicate Classification, Scene Graph Classification and Scene Graph Detection (Xu et al., 2017). We focus on Scene Graph Detection for both datasets, as it is the most challenging and comprehensive subtask, which involves localizing objects and predicting their classes and relations.

Metrics. Following previous works (Yang et al., 2022; Zhou et al., 2023b; Wang et al., 2024), we use Recall@K (R@K) and mean Recall@K (mR@K) as our metrics.

While Recall@K is biased towards frequent classes, mean Recall@K gives all classes the same weight.

4.3 Implementation details

In our experiments, we use Mask2Former (Cheng et al., 2022a) pretrained on COCO (Lin et al., 2014) dataset to initialize the panoptic segmentation network in the vision feature extractor, while the vision-language prompter is trained from scratch. In language feature extractor, we utilize by default GPT-4 (Achiam et al., 2023) as the LLM, and employ OpenAI Embedding Service (Achiam et al., 2023) as the text encoder. We use GPT-4 via the standard OpenAI API to extract language features for the PSG dataset, which takes about one day. For the non-overlapping portion of the VG dataset, extraction requires an additional 14 hours. We store the extracted language prompting features in a database and then retrieve the RP-language prompting feature using "sub#obj", and the RJ-language prompting feature using "sub#rel#obj". We adopt the same data augmentation settings following previous methods (Yang et al., 2022; Zhou et al., 2023b). To train our model, we use the AdamW (Loshchilov & Hutter, 2019), with a learning rate of $1e^{-4}$ and weight decay of $5e^{-2}$. We set λ to 0.1 in our final loss function. Our model is trained for 12 epochs with a step scheduler reducing the learning rate to $1e^{-5}$ at epoch 6 and further to $1e^{-6}$ at epoch 10. The training takes approximately 18 hours on four A100 GPUs, with a batch size of 1 for each GPU. The inference of our model follows the same forward process in training.

4.4 Comparison to the state-of-the-art

PSG. Tab. 1 reports the performance of our method compared to previous state-of-the-art methods on the PSG dataset (Yang et al., 2022). Previous methods rely solely on vision inputs, *i.e.*, images, while ours utilizes both vision and language inputs. For a fair comparison, we use the same ResNet-50 (He et al., 2016) backbone for all methods in the vision feature extractor. Our method shows superior performance compared to all previous methods. Specifically, VLPrompt achieves substantial improvements over previous methods, with gains of +5.3 in R@20, +3.9 in mR@20, +4.8 in R@50, +6.3 in mR@50, +2.4 in R@100, and +3.6 in mR@100.

VG. Tab. 2 reports the performance of our method compared to previous methods on the VG dataset (Krishna et al., 2017). To adapt our method to the VG dataset, *i.e.*, a bounding box-based SGG task, we first use a Segment Anything Model (SAM) (Kirillov et al., 2023) with VG dataset's ground-truth bounding box annotations as prompts to transform the VG dataset into a dataset suitable for instance segmentation tasks. We then train a Mask2Former on this instance segmentation dataset, enabling our method to be adapted to the VG dataset. As shown in Tab. 2, our method surpasses previous vision-only and vision-language methods on the mR@K metric, achieving improvements of +1.9 in mR@20, +1.2 in mR@50, and +1.5 in mR@100. This indicates that incorporating language information effectively enhances the prediction performance for rare relations and alleviates the long-tail problem. Additionally, our method achieves a +2.0 improvement on R@20 and comparable performance on R@50 and R@100, and while it performs slightly lower than DSGG on R@K (*e.g.*, -0.7 on R@50, -2.0 on R@100), it significantly surpasses DSGG on mR@K (*e.g.*, +4.0 on mR@50, +2.5 on mR@100), thereby demonstrating superior overall performance.

4.5 Analysis

Qualitative analysis. As shown in Fig. 3, with the inclusion of language information, VLPrompt successfully predicts challenging relations, which are the highlighted in yellow. Fig. 5 showcases additional visualization results of our VLPrompt. For each example, the top shows the results of panoptic segmentation, the bottom left displays the ground truth, and the bottom right shows the top 10 relation prediction results. Based on visualization results, our VLPrompt exhibits precise capabilities in relation prediction, thereby enhancing scene understanding.

Failure case analysis. In addition to the high-quality visual results shown above, we also conduct a detailed failure case analysis, as illustrated in Fig. 6. We observe that when multiple objects of the same category appear in close proximity within a scene, the model tends to mispredict the relations between them. For example, in the first image from the left, 0_person is only *wearing* 3_baseball_glove, but the model incorrectly predicts that they are also *wearing* 2_baseball_glove. A similar misprediction occurs for 1_person. In the second image, only 0_person is actually *flying* the kite, while 1_person and 2_person are merely *looking at* it. When similar objects are located too close together, the segmentation model may mistakenly identify them as a single instance. For example, in the fourth image, the model predicts that both 0_person and 1_person are *riding* 2_horse, whereas 2_horse should be segmented into two separate horses instead of one. Although incorporating language knowledge extracted from LLMs significantly alleviates the long-tail problem, reasoning about fine-grained relations between spatially adjacent objects remains a challenging issue that requires further investigation.

Alleviating the long-tail problem. In SGG and PSG tasks, mR@K is often used as a reflection for a method's ability to solve long-tail problem (Tang et al., 2020). To further validate it, in PSG dataset, we split relations occurring over 1000 times as common relations, and those under 1000 as rare relations, resulting into 21 common and 35 rare relations.

Table 1 Comparison between our VLPrompt and other methods on the PSG dataset. Our method shows superior performance compared to all previous methods. Note **bold** indicates the best result, and underline indicates the second-best result

Method	Model Input	Scene Graph Detection					
		R@20	mR@20	R@50	mR@50	R@100	mR@100
IMP (Xu et al., 2017)	Vision	16.5	6.5	18.2	7.1	18.6	7.2
MOTIF (Zellers et al., 2018)	Vision	20.0	9.1	21.7	9.6	22.0	9.7
VCTree (Tang et al., 2019)	Vision	20.6	9.7	22.1	10.2	22.5	10.2
GPSNet (Lin et al., 2020)	Vision	17.8	7.0	19.6	7.5	20.1	7.7
PSGTR (Yang et al., 2022)	Vision	28.4	16.6	34.4	20.8	36.3	22.1
PSGFormer (Yang et al., 2022)	Vision	18.0	14.8	19.6	17.0	20.1	17.6
PairNet (Wang et al., 2024)	Vision	29.6	24.7	35.6	28.5	39.6	30.6
HiLo (Zhou et al., 2023b)	Vision	34.1	23.7	40.7	30.3	43.0	33.1
DSGG (Hayder & He, 2024)	Vision	32.7	30.8	42.8	38.8	50.0	43.4
DSFormer (Lorenz et al., 2024)	Vision	–	27.2	–	30.7	–	50.1
VLPrompt (ours)	Vision + Language	**39.4**	**34.7**	**47.6**	**45.1**	**52.4**	**53.7**

Table 2 Comparison between our VLPrompt and other methods on the VG dataset. Our method surpasses all previous methods on the mR@K while achieving comparable performance on the R@K, demonstrating its effectiveness in alleviating the long-tail problem

Method	Model Input	Scene Graph Detection					
		R@20	mR@20	R@50	mR@50	R@100	mR@100
MOTIF (Zellers et al., 2018)	Vision	21.7	–	31.0	6.7	35.1	7.7
VCTree (Tang et al., 2019)	Vision	22.0	–	30.2	6.7	34.6	8.0
Transformer (Tang et al., 2020)	Vision	–	–	30.0	7.4	34.3	8.8
GPSNet (Lin et al., 2020)	Vision	22.3	–	30.3	5.9	35.0	7.1
IETrans (Zhang et al., 2022)	Vision	–	–	25.9	14.6	28.1	16.5
SVRP (He et al., 2022)	Vision + Language	–	–	31.8	10.5	35.8	12.8
HiLo (Zhou et al., 2023b)	Vision	–	–	25.6	15.8	27.9	18.0
DSGG (Hayder & He, 2024)	Vision	–	–	**32.9**	13.0	**38.5**	17.3
EGTR (Im et al., 2024)	Vision	22.4	8.8	28.2	14.0	31.7	18.3
SBG (Li et al., 2025)	Vision	–	–	27.0	13.8	31.3	16.1
VLPrompt (ours)	Vision + Language	**24.4**	**10.7**	32.2	**17.0**	36.4	**19.8**

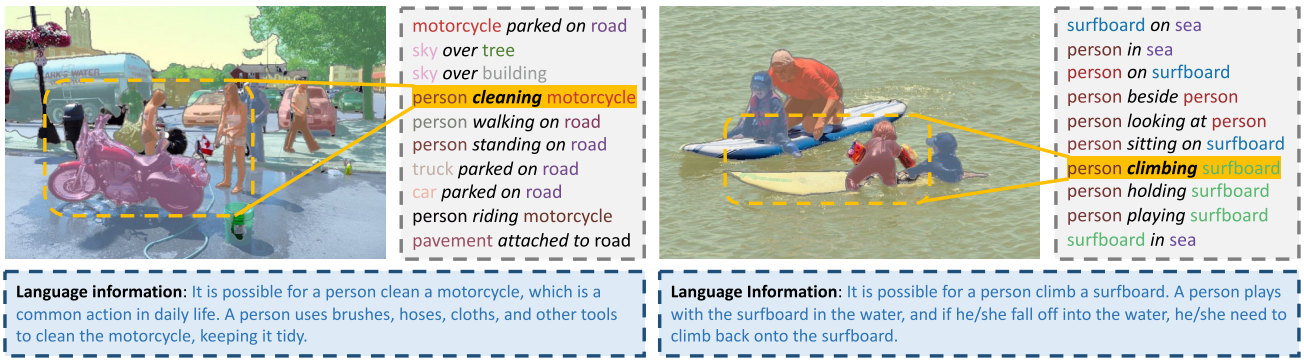

Fig. 3 Visualization results of our VLPrompt. We show two examples. For each example, the top left displays the predicted segmentation results, the top right shows the top 10 predicted relation triplets (all are correct relation triplets), and bottom is the language snippet utilized for predicting the highlighted triplets in yellow (color figure online)

Table 3 Alleviating the long-tail problem on PSG dataset. Incorporating LLM-based language information boosts performance on common and rare relations, with greater gains for rare ones, validating its effectiveness in addressing the long-tail problem

	Common relations		Rare relations	
	w/o LLM	w/ LLM	w/o LLM	w/ LLM
mR@100	54.7	**57.0** (+2.3)	37.9	**51.7** (+13.8)

Table 4 Alleviating the long-tail problem on VG dataset

	Common relations		Rare relations	
	w/o LLM	w/ LLM	w/o LLM	w/ LLM
mR@100	21.6	**23.2** (+1.6)	9.7	**16.4** (+6.7)

As shown in Tab. 3, our method's integration of language information (w/ LLM) leads to 2.3 increase in mR@100 for common relations and 13.8 increase for rare relations on the PSG dataset. Tab. 4 reveals consistent findings on the VG dataset. The significant improvement in the latter highlights the effectiveness in enhancing rare relation prediction. Additionally, as shown in Fig. 4, we present relation/predicate-wise performance improvements on PSG dataset, further highlighting the effectiveness of our method across different relations. Note, w/o LLM is our method without language information (Sec. 3). In this setting, we remove the cross-attention module and rely solely on vision features for relation prediction.

Comparison with large multimodal models. Large multimodal models (LMMs) currently demonstrate great performance across various multimodal tasks and can also predict relations with specific instructions. To further validate the performance of LMMs on PSG task, we test three popular LMMs GPT-4o, GPT-4V and Llama 3.2-11B on the PSG test set. To ensure fairness in comparison, we allow LMMs to use the segmentation results output by our method, thus ensuring same segmentation performance. Specifically, following the approach of Yang et al. (2023c), we attach the panoptic segmentation results extracted by our model to the original image and input it to LMMs. This allows LMMs to obtain object segmentation results and the corresponding categories consistent with our method. Subsequently, we use LMMs to predict the relations for all object pairs. From the experimental results, as shown in Tab. 5, we observe that as more advanced LMMs are used, their performance steadily improves. However, a noticeable performance gap still remains compared to our PSG model. This trend indicates that LMMs hold strong potential for achieving better results. In turn, we believe that future advances in LMMs may further promote progress in panoptic scene graph generation.

4.6 Ablation study

4.6.1 Vision Feature Extractor

Object features from pixel decoder. To validate the superiority of obtaining object features from the pixel decoder (PixelDec) of Mask2Former, we experiment with an alternative approach: acquiring corresponding object features from the transformer decoder (TsfmDec) of Mask2Former. This is a common practice in the literature (Cong et al., 2023). Experiments (PixelDec→TsfmDec) in Tab. 6 show that the feature from the pixel decoder performs 3.4 better in mR@100 than that from the transformer decoder, as the pixel decoder contains more comprehensive vision information.

Mask pooling for object feature extraction. Common methods to obtain object features from the pixel decoder include mask pooling (MaskPool) and bounding box pooling (BboxPool) (Girshick, 2015). We validate our choice of mask pooling through ablation experiments (MaskPool→BboxPool). As shown in Tab. 6, the performance using mask pooling is 1.1 higher in mR@100 than that using bbox pooling. Mask pooling is more suitable for the mask prediction in the PSG task.

Concatenating object features into a pair feature. Common methods for merging the features of two objects for their relation prediction include concatenation (Concat.) (Zhang et al., 2019b) and subtraction (Sub.) (Cheng et al., 2022b). Notably, addition of the features cannot be used due to the inherent order requested between the subject and object. Results (Concat.→Sub.) in Tab. 6 indicate that concatenation outperforms subtraction by 4.8 on the mR@100.

Spatial feature. To validate the effect of adding spatial features, we conduct an ablation study by removing these features. The results (w/o Spatial Feat.) in Tab. 6 show that this leads to a decrease of -0.7 in R@100 and -1.6 in mR@100. This is because spatial features provide the model with additional spatial interaction between the subject and object (Sec. 3.1), thereby enhancing the vision feature for relation prediction.

4.6.2 Language Feature Extractor

Chain-of-thought for prompt. By carefully designing prompts with the chain-of-thought technique, LLMs can produce rich and accurate descriptions. However, if we do not use chain-of-thought and instead directly ask LLMs questions, such as replacing the relation proposer prompt with "What are the possible relations between subject and object? And why?" or the relation judger prompt with "Could this relation be possible between subject and object? Why?", we find that the outputs from LLMs become much less predictable and often not as expected. We experiment without chain-of-thought (w/o CoT) and the results in Tab. 7 show that the

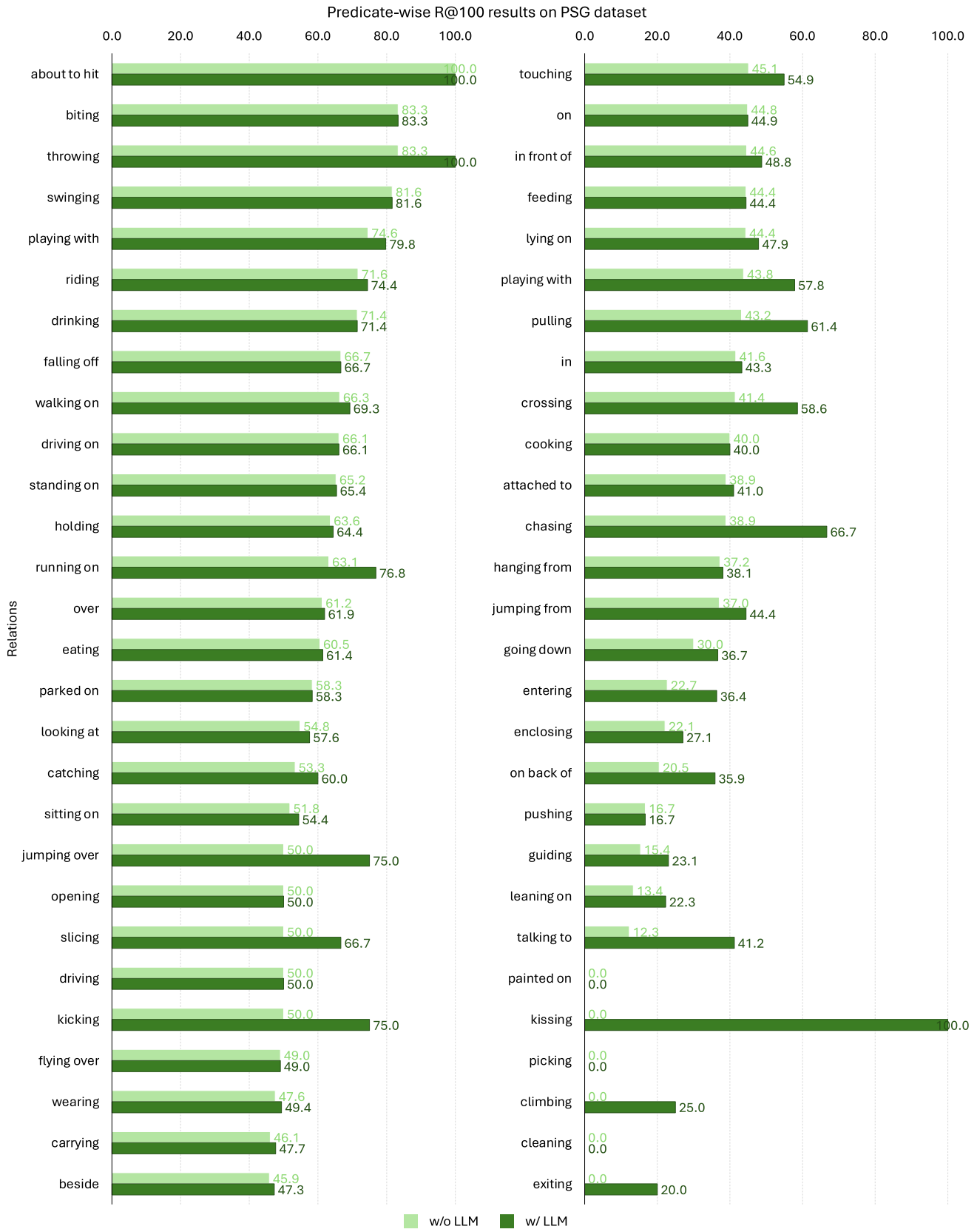

Fig. 4 Predicate-wise R@100 results on the PSG dataset show performance improvements across various relations with LLM integration

Fig. 5 Visualization of panoptic segmentation (top), ground truth (bottom left), and top 10 relation predictions (bottom right) demonstrates VLPrompt's precision in relation prediction, enhancing scene understanding

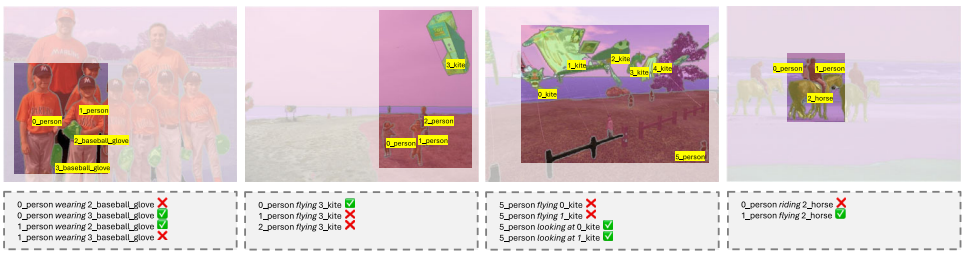

Fig. 6 Visualization of typical failure cases. We visualize typical failure cases of VLPrompt, which mainly occur when multiple instances of the same object category appear in the scene. In such cases, the model often mismatches the relations between different objects

Table 5 Comparison with LMMs in PSG task. Large multimodal models like GPT-4V, not tailored for PSG task, underperform compared to task-specific PSG models

Method	R@20	mR@20	R@50	mR@50	R@100	mR@100
VLPrompt	**39.4**	**34.7**	**47.6**	**45.1**	**52.4**	**53.7**
GPT-4o (Achiam et al., 2023)	30.9	28.6	40.2	38.9	45.9	44.1
GPT-4V (Achiam et al., 2023)	16.9	10.1	20.4	12.8	22.5	13.1
Llama 3.2-11B (Dubey et al., 2024)	14.8	6.9	16.9	8.2	18.1	9.5

Table 6 Ablation study for the Vision Feature Extractor

Method	R@20	mR@20	R@50	mR@50	R@100	mR@100
VLPrompt	**39.4**	**34.7**	**47.6**	**45.1**	**52.4**	**53.7**
PixelDec → TsfmDec	37.4	32.1	45.8	43.7	50.1	50.3
MaskPool → BboxPool	38.6	33.7	46.4	44.8	51.0	52.6
Concat. → Sub.	35.4	29.8	44.1	42.3	48.6	48.9
w/o Spatial Feat.	39.0	33.0	46.6	44.3	51.7	52.1

Table 7 Ablation study for the Language Feature Extractor

Method	R@20	mR@20	R@50	mR@50	R@100	mR@100
VLPrompt	**39.4**	**34.7**	**47.6**	**45.1**	**52.4**	**53.7**
w/o CoT	36.8	27.9	44.3	38.8	48.6	45.7
w/o template	39.0	33.7	47.0	43.8	51.6	52.2
$\text{Ext}^{\text{RP}} \rightarrow \text{Ext}^{\text{RJ}}$	39.1	34.2	47.5	45.1	52.3	53.5
$\text{Ext}^{\text{RJ}} \rightarrow \text{Ext}^{\text{RP}}$	38.9	31.2	46.8	41.7	51.0	49.9
$\text{Swap}(\text{Ext}^{\text{RP}}, \text{Ext}^{\text{RJ}})$	36.2	29.4	44.3	38.6	48.7	46.4
Llama3-8B + Bert	39.0	33.5	47.3	44.8	52.0	53.2
Compression	38.3	33.1	46.5	44.6	51.2	51.1

performance of relation prediction significantly drops, *i.e.*, a -8.0 decrease in mR@100. This further illustrates the rationale and importance of using chain-of-thought technology for designing prompts.

Template phrase in RP-prompt. The template phrase appended to the end of the RP-prompt helps the model clearly distinguish common relations from less plausible ones, thereby preventing performance degradation on high-frequency relations during prediction. To validate this design, we conduct an ablation study by removing the template phrase. As shown in Tab. 7 (w/o template), excluding the template leads to a slight performance drop (-0.8 R@100, -1.5 mR@100), indicating that explicitly guiding the model to differentiate between common and uncommon relations through prompt design is beneficial.

Different feature extraction methods. To validate our design of different feature extract methods for RP- and RJ-language prompting features, we study their effects. We offer three variants: 1) For RP-language prompting features, we adopt the same way as the RJ-language prompting feature to extract feature for each relation triplet individually, we denote this variant as $\text{Ext}^{\text{RP}} \rightarrow \text{Ext}^{\text{RJ}}$ in the Tab. 7. It shows that performance is slightly lower than our original VLPrompt, but with increased computation. 2) For RJ-language prompting features, we adopt the same way as the RP-language prompting feature to extract all relation triplets into one feature, denoted by $\text{Ext}^{\text{RJ}} \rightarrow \text{Ext}^{\text{RP}}$ in the Tab. 7. We observe a clear drop in mR@K, as this would condense the features of relations between subject and object while dropping the subtle details, which can be especially disadvantageous for rare relations. 3) We swap the feature extraction methods between Ext^{RP} and Ext^{RJ}, denoted by $\text{Swap}(\text{Ext}^{\text{RP}}, \text{Ext}^{\text{RJ}})$ in Tab. 7. We observe a substantial decrease in both metrics. Our original feature extraction is specifically designed on one side to focus on the condensed and distinct attributes of commonly occurring relations; on the other side to focus on the detailed and subtle attributes of all possible especially rare relations for a subject-object pair.

Different LLMs and text encoders. To further assess the effects of different LLMs and text encoders on model performance, we attempt to replace the GPT-4 with Llama3-8B (Touvron et al., 2023b) and the OpenAI Embedding Service with Bert (Devlin et al., 2018). When using Bert, we take the mean of the embeddings of all output tokens as the feature. The experimental results (Llama3-8B + Bert) in Tab. 7 reveal that using Llama3-8B as the LLM and Bert as the text encoder results in only a slight decrease in performance: -0.4 in R@100 and -0.5 in mR@100. We review the descriptions output by Llama3-8B and compare them with those from GPT-4, finding no significant differences in quality, more details can be found in supplementary materials. This suggests that, with carefully designed prompts, open-source LLMs with reduced parameters also work for our method, thus validating the flexibility of our method.

Compress the language prompting feature. At runtime, the language prompting feature occupies 290.57MB of memory, which is manageable. To further enhance the model's applicability in real-world scenarios, such as when there are more object and relation categories, we compress the language prompting feature to 1/4 of its original size using an encoder-decoder models. Results (Compression) shown in Tab. 7 indicate that compressing the language prompting feature to 1/4 leads to only a minor performance decrease, while the memory required by the model during runtime is reduced to 1/4 of the original, which further demonstrates the practicality of our method in real-world settings.

4.6.3 Vision-Language Prompter

Language information. To validate the efficacy of incorporating language information, we conduct a comparative experiment by replacing all language prompting features in the vision-language prompter with vision prompting features. This approach ensures that all other factors remain constant while assessing the impact of language information. As shown in Tab. 8, the experimental results (w/o Language) reveal a significant decrease in the mR@100 by -9.5 when language information is removed, which demonstrates the substantial impact of language information in mitigating the long-tail problem.

Table 8 Ablation study for the Vision-Language Prompter

Method	R@20	mR@20	R@50	mR@50	R@100	mR@100
VLPrompt	**39.4**	**34.7**	**47.6**	**45.1**	**52.4**	**53.7**
w/o Language	37.1	26.8	45.9	37.2	50.0	44.2
RP-decoder	38.4	30.6	46.6	41.2	50.9	49.6
RJ-decoder	37.9	34.2	45.3	44.6	50.4	52.8
Uncompressed F_L^{RJ} for W^{RJ}	39.3	34.7	47.5	44.9	52.1	53.7

Results of RP-decoder and RJ-decoder. We test the relation prediction performance of both RP-decoder and RJ-decoder separately. The results (RP-decoder and RJ-decoder) in Tab. 8 show that the RP-decoder outperforms the RJ-decoder in the R@K (*e.g.*, 50.9 *v.s.* 50.4 for R@100), while the RP-decoder scores lower in mR@K compared to the RJ-decoder (*e.g.*, 49.6 *v.s.* 52.8 for mR@100). This indicates that the RP-decoder is good at predicting frequent relation classes, whereas the RJ-decoder excels in rare relation classes, which indicates that they are complementary.

Relation fusion. As shown in Tab. 8, when the outputs of the two branches are combined via our relation fusion strategy (*i.e.*, VLPrompt), the final prediction outperforms either individual branch, demonstrating the effectiveness of our design. Specifically, the RP-decoder achieves higher R@K but lower mR@K scores compared to the RJ-decoder, as it receives RP-language prompting features that encode the most frequent relations for each subject-object pair. This makes the model more biased toward high-frequency relations, resulting in strong performance on common categories but poor generalization to rare ones, due to the long-tail distribution of relation labels. In contrast, the RJ-decoder, which uses RJ-language prompting features to independently judge each relation category, achieves better mR@K at the cost of slightly reduced R@K, as it improves rare-relation prediction by sacrificing some performance on frequent ones. These complementary behaviors highlight the importance of integrating both decoders, allowing the model to leverage their respective strengths and achieve overall optimal performance.

Fusion weights. The input feature F_L^{RJ} used to compute the fusion weights W^{RJ} is compressed along the relation dimension, and this operation is applied solely within the gate network. Our empirical results show that this compression has minimal impact on performance, as the gate network is only responsible for generating fusion weights. To validate this design choice, we conduct an ablation study. As shown in Tab. 8 (Uncompressed F_L^{RJ} for W^{RJ}), using the uncompressed F_L^{RJ} yields performance comparable to its compressed counterpart. This confirms that the compressed version of F_L^{RJ} offers a more computationally efficient implementation without sacrificing performance.

The number of decoder blocks. To elucidate the reason that both RP-decoder and RJ-decoder use a 2-block transformer decoder, we vary different numbers of blocks in Tab. 9. We observe that when increasing the transformer decoder blocks from 1 to 2, there is a significant improvement in model performance. However, increasing from 2 to 4 blocks does not change much for the performance. Further increasing the decoder blocks to 12 leads to a notable decrease in performance, suggesting that too many decoder blocks may cause the model to overfit. Considering both performance and speed, we choose 2 blocks.

4.6.4 Efficiency Analysis

In Tab. 10, we compare our model's efficiency with the previous state-of-the-art models, evaluating computational floating point operations per second (FLOPS), parameter size and inference speed on the same A100 GPU. We observe that although our method has higher FLOPS compared to HiLo, it matches HiLo in prediction speed, and significantly outperforms HiLo in performance (see Tab. 1).

5 Limitation

While our method achieves significant performance improvements on the PSG task, there are still several limitations that warrant further exploration.

First, our approach requires the pre-extraction of language information from LLMs to enhance relation prediction. Although the extraction process is efficient and cost-effective, it introduces a dependency on external models and preprocessing, which may limit scalability in certain deployment scenarios.

Second, our method currently focuses on closed-set relation prediction and does not support open-set relation prediction. This restricts its applicability to real-world settings where unseen relations frequently occur. Although we significantly mitigate the long-tail issue within the closed-set setting, bridging the gap to open-set generalization remains an important challenge.

Table 9 Ablation study for the number of decoder blocks

Block Number	R@20	mR@20	R@50	mR@50	R@100	mR@100
1	35.1	29.8	44.8	39.5	47.2	46.9
2	**39.4**	34.7	**47.6**	45.1	**52.4**	53.7
4	38.9	**34.8**	47.4	**45.2**	52.1	**53.9**
12	18.4	14.5	27.5	23.0	33.2	29.4

Table 10 Analysis for efficiency. Our method achieves significant performance improvements without introducing notable efficiency issues, maintaining a comparable level to previous approaches

Method	FLOPS (G)	Parameters (G)	Inference Speed (ms)
PSGTR (Yang et al., 2022)	461.3	44.2	140
HiLo (Zhou et al., 2023b)	229.4	58.7	156
VLPrompt (ours)	386.7	49.1	152

In future work, we plan to explore building multimodal large language models that can jointly model visual and relational information. Such models have the potential to directly perform open-set relation prediction, thus further enhancing the generalization and applicability of panoptic scene graph generation systems.

6 Conclusion

In this work, we introduce VLPrompt, the novel method to incorporate language information generated by LLMs to enhance the PSG task performance. VLPrompt utilizes the chain-of-thought method in designing prompts, enabling LLMs to generate rich descriptions for relation prediction. Additionally, we develop a prompter network based on attention mechanisms to facilitate comprehensive interaction between vision and language information, achieving high-quality relation prediction. Experiments demonstrate that our method significantly outperforms the current state-of-the-art on the PSG dataset and mitigates the long-tail problem for relations. In future work, we plan to explore the use of LLMs for open-set relation prediction and further refine the model by distillation to enhance efficiency, enabling broader application in downstream tasks.

Acknowledgements This work was supported by the Fundamental Research Funds for the Central Universities.

Data Availability Panoptic Scene Graph (PSG) dataset is available at http://psgdataset.org/.
Visual Genome (VG) dataset is available at https://homes.cs.washington.edu/~ranjay/visualgenome/index.html.

Declarations

Competing interests The authors declare that they have no competing interests.

References

Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., Avila, R., Babuschkin, I., Balaji, S., Balcom, V., Baltescu, P., Bao, H., Bavarian, M., Belgum, J., Bello, I., et al. (2023). Gpt-4 technical report. arXiv preprint arXiv:2303.08774.

Aditya, S., Yang, Y., Baral, C., Aloimonos, Y., & Fermüller, C. (2018). Image understanding using vision and reasoning through scene description graph. *Computer Vision and Image Understanding, 173*, 33–45.

Amiri, S., Chandan, K., & Zhang, S. (2022). Reasoning with scene graphs for robot planning under partial observability. *IEEE Robotics and Automation Letters, 7*(2), 5560–5567.

Anthropic (2023) Claude. https://claude.ai/chats.

Caesar, H., Uijlings, J.,& Ferrari, V. (2018). Coco-stuff: Thing and stuff classes in context. In *CVPR*, (pp 1209–1218).

Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., & Zagoruyko, S. (2020). End-to-end object detection with transformers. In *ECCV*, (pp 213–229).

Chen, L., Zhang, H., Xiao, J., He, X., Pu, S., & Chang, S. (2019). Counterfactual critic multi-agent training for scene graph generation. In *ICCV*, (pp 4613–4623).

Chen, L., Li, B., Shen, S., Yang, J., Li, C., Keutzer, K., Darrell, T., & Liu, Z. (2023). Large language models are visual reasoning coordinators. *NeurIPS, 36*, 70115–70140.

Chen, S., Jin, Q., Wang, P., & Wu, Q. (2020). Say as you wish: Fine-grained control of image caption generation with abstract scene graphs. In *CVPR*, (pp 9962–9971).

Cheng, B., Misra, I., Schwing, A. G., Kirillov, A., & Girdhar, R. (2022a). Masked-attention mask transformer for universal image segmentation. In *CVPR*, (pp 1290–1299).

Cheng, J., Wang, L., Wu, J., Hu, X., Jeon, G., Tao, D., & Zhou, M. (2022). Visual relationship detection: A survey. *IEEE Transactions on Cybernetics, 52*(8), 8453–8466.

Cong, Y., Yang, M. Y., & Rosenhahn, B. (2023). Reltr: Relation transformer for scene graph generation. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 45*(9), 11169–11183.

Dai, B., Zhang, Y., & Lin, D. (2017). Detecting visual relationships with deep relational networks. In *CVPR*, (pp 3076–3086).

Deng, Y., Li, Y., Zhang, Y., Xiang, X., Wang, J., Chen, J., & Ma, J. (2022). Hierarchical memory learning for fine-grained scene graph generation. In *ECCV*, (pp 266–283).

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

Dong, X., Gan, T., Song, X., Wu, J., Cheng, Y., & Nie, L. (2022). Stacked hybrid-attention and group collaborative learning for unbiased scene graph generation. In *CVPR*, (pp 19427–19436).

Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Yang, A., Fan, A., et al. (2024). The llama 3 herd of models. arXiv preprint arXiv:2407.21783.

Dupty, M. H., Zhang, Z., & Lee, W. S. (2020). Visual relationship detection with low rank non-negative tensor decomposition. In *AAAI*, (pp 10737–10744).

Gao, L., Wang, B., & Wang, W. (2018). Image captioning with scene-graph based semantic concepts. In *ICMLC*, (pp 225–229).

Girshick, R. (2015). Fast r-cnn. In *ICCV*, (pp 1440–1448).

Gu, J., Zhao, H., Lin., Z., Li, S., Cai, J., & Ling, M. (2019). Scene graph generation with external knowledge and image reconstruction. In *CVPR*, (pp 1969–1978).

Hayder, Z., & He, X. (2024). Dsgg: Dense relation transformer for an end-to-end scene graph generation. In *CVPR*, (pp 28317–28326).

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *CVPR*, (pp 770–778).

He, T., Gao, L., Song, J., & Li, Y.-F. (2022). Towards open-vocabulary scene graph generation with prompt-based finetuning. In *ECCV*, (pp 56–73).

Hildebrandt, M., Li, H., Koner, R., Tresp, V., & Günnemann, S. (2020). Scene graph reasoning for visual question answering. IEEE Transactions on Multimedia.

Hu, Y., Chen, S., Chen, X., Zhang, Y., & Gu, X. (2019). Neural message passing for visual relationship detection. In *ICMLW*, (pp 0).

Huang, X., Huang, Y. J., Zhang, Y., Tian, W., Feng, R., Zhang, Y., Xie, Y., Li, Y., & Zhang, L. (2023). Inject semantic concepts into image tagging for open-set recognition. arXiv preprint arXiv:2310.15200.

Hung, Z. S., Mallya, A., & Lazebnik, S. (2020). Contextual translation embedding for visual relationship detection and scene graph generation. *IEEE transactions on pattern analysis and machine intelligence, 43*(11), 3820–3832.

Hwang, S. J., Ravi, S. N., Tao, Z., Kim, H. J., Collins, M. D., & Singh, V. (2018). Tensorize, factorize and regularize: Robust visual relationship learning. In *CVPR*, (pp 1014–1023).

Im, J., Nam, J., Park, N., Lee, H., & Park, S. (2024). Egtr: Extracting graph from transformer for scene graph generation. In *CVPR*, (pp 24229–24238).

Kang, H., & Yoo, C. D. (2023). Skew class-balanced re-weighting for unbiased scene graph generation. *Machine Learning and Knowledge Extraction, 5*(1), 287–303.

Kirillov, A., He, K., Girshick, R., Rother, C., & Dollár, P. (2019). Panoptic segmentation. In *CVPR*, (pp 9404–9413).

Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A. C., Lo, W.-Y. et al. (2023). Segment anything. In *ICCV*, (pp 4015–4026).

Krishna, R., Zhu, Y., Groth, O., Johnson, J., Hata, K., Kravitz, J., Chen, S., Kalantidis, Y., Li, L.-J., Shamma, D. A., Bernstein, S. M., & Fei-Fei, L. (2017). Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International journal of computer vision, 123*(1), 32–73.

Lai, X., Tian, Z., Chen, Y., Li, Y., Yuan, Y., Liu, S., & Jia, J. (2024). Lisa: Reasoning segmentation via large language model. In *CVPR*, (pp 9579–9589).

Li, J., Wang, Y., Guo, X., Yang, R., & Li, W. (2024a). Leveraging predicate and triplet learning for scene graph generation. In *CVPR*, (pp 28369–28379).

Li, L., Ji, W., Wu, Y., Li, M., Qin, Y., Wei, L., & Zimmermann, R. (2024b). Panoptic scene graph generation with semantics-prototype learning. In *AAAI*, (pp 3145–3153).

Li, R., Zhang, S., Wan, B., & He, X. (2021). Bipartite graph network with adaptive message passing for unbiased scene graph generation. In *CVPR*, (pp 11109–11119).

Li, R., Zhang, S., & He, X. (2022a). Sgtr: End-to-end scene graph generation with transformer. In *CVPR*, (pp 19486–19496).

Li, W., Zhang, H., Bai, Q., Zhao, G., Jiang, N., & Yuan, X. (2022b). Ppdl: Predicate probability distribution based loss for unbiased scene graph generation. In *CVPR*, (pp 19447–19456).

Li, Y., Ouyang, W., Wang, X., & Tang, X. (2017a). Vip-cnn: Visual phrase guided convolutional neural network. In *CVPR*, (pp 1347–1356).

Li, Y., Ouyang, W., Zhou, B., Wang, K., & Wang, X. (2017b). Scene graph generation from objects, phrases and region captions. In *ICCV*, (pp 1261–1270).

Li, Y., Wang, T., Wu, K., Wang, L., Guo, X., & Wang, W. (2025). Fine-grained scene graph generation via sample-level bias prediction. In *ECCV*, (pp 18–35).

Liao, W., Rosenhahn, B., Shuai, L., & Ying Yang, M. (2019). Natural language guided visual relationship detection. In *CVPRW*, (pp 0).

Lin, T. Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., & Zitnick, C. L. (2014). Microsoft coco: Common objects in context. In *ECCV*, (pp 740–755).

Lin, X., Ding, C., Zeng, J., & Tao, D. (2020). Gps-net: Graph property sensing network for scene graph generation. In *CVPR*, (pp 3746–3753).

Liu, X., Tang, W., Ni, X., Lu, J., Zhao, R., Li, Z., & Tan, F. (2023). What large language models bring to text-rich vqa? arXiv preprint arXiv:2311.07306.

Lorenz, J., Pest, A., Kienzle, D., Ludwig, K., & Lienhart, R. (2024). A fair ranking and new model for panoptic scene graph generation. In *ECCV*, (pp 148–164).

Loshchilov, I., & Hutter, F. (2019). Decoupled weight decay regularization. In *ICLR*, (pp 1–18).

Lu, C., Krishna, R., Bernstein, M., & Fei-Fei, L. (2016). Visual relationship detection with language priors. In *ECCV*, (pp 852–869).

Lyu, X., Gao, L., Guo, Y., Zhao, Z., Huang, H., Shen, H. T., & Song, J. (2022). Fine-grained predicates learning for scene graph generation. In *CVPR*, (pp 19467–19475).

Min, B., Ross, H., Sulem, E., Veyseh, A. P. B., Nguyen, T. H., Sainz, O., Agirre, E., Heintz, I., & Roth, D. (2023). Recent advances in natural language processing via large pre-trained language models: A survey. *ACM Computing Surveys, 56*(2), 1–40.

Peyre, J., Sivic, J., Laptev, I., & Schmid, C. (2017). Weakly-supervised learning of visual relations. In *ICCV*, (pp 5179–5188).

Qi, L., Chen, Y. W., Yang, L., Shen, T., Li, X., Guo, W., Xu, Y., & Yang, M. -H. (2024). Generalizable entity grounding via assistance of large language model. arXiv preprint arXiv:2402.02555.

Qi, M., Li, W., Yang, Z., Wang, Y., & Luo, J. (2019). Attentive relational networks for mapping images to scene graphs. In *CVPR*, (pp 3957–3966).

Shah, D., Osiński, B., Levine, S., & Ichter, B. (2023). Lm-nav: Robotic navigation with large pre-trained models of language, vision, and action. In: Conference on robot learning, (pp 492–504).

Shi, J., Zhang, H., & Li, J. (2019). Explainable and explicit visual reasoning over scene graphs. In *CVPR*, (pp 8376–8384).

Shit, S., Koner, R., Wittmann, B., Paetzold, J., Ezhov, I., Li, H., Pan, J., Sharifzadeh, S., Kaissis, G., Tresp, V., & Menze, B. (2022). Relationformer: A unified framework for image-to-graph generation. In *ECCV*, (pp 422–439).

Singh, K. P., Salvador, J., Weihs, L., & Kembhavi, A. (2023). Scene graph contrastive learning for embodied navigation. In *ICCV*, (pp 10884–10894).

Su, J., Zhu, M., Murtadha, A., Pan, S., Wen, B., & Liu, Y. (2022). Zlpr: A novel loss for multi-label classification. arXiv preprint arXiv:2208.02955.

Tang, J., Zheng, G., Yu, J., & Yang, S. (2023). Cotdet: Affordance knowledge prompting for task driven object detection. In *ICCV*, (pp 3068–3078).

Tang, K., Zhang, H., Wu, B., Luo, W., & Liu, W. (2019). Learning to compose dynamic tree structures for visual contexts. In *CVPR*, (pp 6619–6628).

Tang, K., Niu, Y., Huang, J., Shi, J., & Zhang, H. (2020). Unbiased scene graph generation from biased training. In *CVPR*, (pp 3716–3725).

Team, G., Anil, R., Borgeaud, S., Alayrac, J. -B., Yu, J., Soricut, R., Schalkwyk, J., Dai, A. M., Hauth, A., Millican, K., Silver, D., Johnson, M., Antonoglou, I., Schrittwieser, J., Glaese, J., Chen, J., Pitler, E., Lillicrap, T., Lazaridou, A., Firat, O., et al. (2023). Gemini: a family of highly capable multimodal models. arXiv preprint arXiv:2312.11805.

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M. -A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., Rodriguez, A., Joulin, A., Grave, E., & Lample, G. (2023a). Llama: Open and efficient foundation language models. arXiv preprint . arXiv:2302.13971

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., Bikel, D., Blecher, L., Ferrer, C.C., Chen, M., Cucurull, G., Esiobu, D, Fernandes, J., Fu, J., Fu, W., et al. (2023b). Llama 2: Open foundation and fine-tuned chat models. arXiv preprint arXiv:2307.09288.

Tsai, Y. H. H., Dhar, V., Li, J., Zhang, B., & Zhang, J. (2023). Multimodal large language model for visual navigation. arXiv preprint arXiv:2310.08669.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł, & Polosukhin, I. (2017). *Attention is all you need. NeurIPS, 30*, 6000–6010.

Wang, J., Wen, Z., Li, X., Guo, Z., Yang, J., & Liu, Z. (2024). Pair then relation: Pair-net for panoptic scene graph generation. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 46*(12), 10452–10465.

Wang, W., Shi, M., Li, Q., Wang, W., Huang, Z., Xing, L., Chen, Z., Li, H., Zhu, X., Cao, Z., Chen, Y., Lu, T., Dai, J., & Qiao, Y. (2023). The all-seeing project: *Towards panoptic visual recognition and understanding of the open world. ICLR* (pp 1–33).

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E., Le, Q. V., Zhou, D., et al. (2022). Chain-of-thought prompting elicits reasoning in large language models. *NeurIPS, 35*, 24824–24837.

Wu, J., Li, X., Xu, S., Yuan, H., Ding, H., Yang, Y., Li, X., Zhang, J., Tong, Y., Jiang, X., et al. (2024). Towards open vocabulary learning: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 46*(7), 5092–5113.

Xenos, A., Stafylakis, T., Patras, I., & Tzimiropoulos, G. (2023). A simple baseline for knowledge-based visual question answering. arXiv preprint arXiv:2310.13570.

Xia, Z., Han, D., Han, Y., Pan, X., Song, S., & Huang, G. (2024). Gsva: Generalized segmentation via multimodal large language models. In *CVPR*, (pp 3858–3869).

Xu, D., Zhu, Y., Choy, C. B., & Fei-Fei, L. (2017). Scene graph generation by iterative message passing. In *CVPR*, (pp 5410–5419).

Yang, J., Ang, Y. Z., Guo, Z., Zhou, K., Zhang, W., & Liu, Z. (2022). Panoptic scene graph generation. In *ECCV*, (pp 178–196).

Yang, J., Cen, J., Peng, W., Liu, S., Hong, F., Li, X., Zhou, K., Chen, Q., & Liu, Z. (2023). 4d panoptic scene graph generation. *NeurIPS, 36*, 69692–69705.

Yang, J., Peng, W., Li, X., Guo, Z., Chen, L., Li, B., Ma, Z., Zhou, K., Zhang, W., Loy, C. C. & Liu, Z. (2023b). Panoptic video scene graph generation. In *CVPR*, (pp 18675–18685).

Yang, J., Zhang, H., Li, F., Zou, X., Li, C., & Gao, J. (2023c). Set-of-mark prompting unleashes extraordinary visual grounding in gpt-4v. arXiv preprint arXiv:2310.11441.

Ye, K., & Kovashka, A. (2021). Linguistic structures as weak supervision for visual scene graph generation. In *CVPR*, (pp 8289–8299).

Yu, J., Chai, Y., Wang, Y., Hu, Y., & Wu, Q. (2021). Cogtree: Cognition tree loss for unbiased scene graph generation. In *IJCAI*, (pp 1274–1280).

Yu, Q., Li, J., Wu, Y., Tang, S., Ji, W., & Zhuang, Y. (2023). Visually-prompted language model for fine-grained scene graph generation in an open world. ICCV (pp 21560–21571).

Zareian, A., Karaman, S., & Chang, S. F. (2020). Bridging knowledge graphs to generate scene graphs. In *ECCV*, (pp 606–623).

Zellers, R., Yatskar, M., Thomson, S., & Choi, Y. (2018). Neural motifs: Scene graph parsing with global context. In *CVPR*, (pp 5831–5840).

Zhang, A., Yao, Y., Chen, Q., Ji, W., Liu, Z., Sun, M., & Chua, T. -S. (2022). Fine-grained scene graph generation with data transfer. In *ECCV*, (pp 409–424).

Zhang, A., Zhao, L., Xie, C. W., Zheng, Y., Ji, W., & Chua, T. -S. (2023). Next-chat: An lmm for chat, detection and segmentation. arXiv preprint arXiv:2311.04498.

Zhang, H., Kyaw, Z., Chang, S. F., & Chua, T. -S. (2017). Visual translation embedding network for visual relation detection. In *CVPR*, (pp 5532–5540).

Zhang, J., Kalantidis, Y., Rohrbach, M., Paluri, M., Elgammal, A., & Elhoseiny, M. (2019a). Large-scale visual relationship understanding. In *AAAI*, (pp 9185–9194).

Zhang, J., Shih, K. J., Elgammal, A., Tao, A., & Catanzaro, B. (2019b). Graphical contrastive losses for scene graph parsing. In *CVPR*, (pp 11535–11543).

Zhang, J., Huang, J., Jin, S., & Lu, S. (2024). Vision-language models for vision tasks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 46*(8), 5625–5644.

Zhang, T., Li, X., Fei, H., Yuan, H., Wu, S., Ji, S., Loy, C. C., & Yan, S. (2024). Omg-llava: Bridging image-level, object-level, pixel-level reasoning and understanding. *NeurIPS, 37*, 71737–71767.

Zhao, C., Shen, Y., Chen, Z., Ding, M., & Gan, C. (2023). Textpsg: Panoptic scene graph generation from textual descriptions. In *ICCV*, (pp 2839–2850).

Zheng, S., Chen, S., & Jin, Q. (2019). Visual relation detection with multi-level attention. In *ACM MM*, (pp 121–129).

Zhong, Y., Shi, J., Yang, J., Xu, C., & Li, Y. (2021). Learning to generate scene graph from natural language supervision. In *ICCV*, (pp 1823–1834).

Zhou, Z., Alabi, O., Wei, M., Vercauteren, T., & Shi, M. (2023). Text promptable surgical instrument segmentation with vision-language models. *NeurIPS, 36*, 28611–28623.

Zhou, Z., Shi, M., & Caesar, H. (2023b). Hilo: Exploiting high low frequency relations for unbiased panoptic scene graph generation. In *ICCV*, (pp 21637–21648).

Zhou, Z., Zhu, Z., Caesar, H., Shi, M. (2025). Openpsg: Open-set panoptic scene graph generation via large multimodal models. In *ECCV*, (pp 199–215).

Zhu, G., Zhang, L., Jiang, Y., Dang, Y., Hou, H., Shen, P., Feng, M., Zhao, X., Miao, Q., Shah, S. A. A. & others (2022). Scene graph generation: A comprehensive survey. *Neurocomputing*, *566*, Article 127052.