Open World Scene Graph Generation using Vision Language Models

Amartya Dutta[1‡] Kazi Sajeed Mehrab[1*] Medha Sawhney[1*] Abhilash Neog[1] Mridul Khurana[1]
Sepideh Fatemi[1] Aanish Pradhan[1] M. Maruf[2] Ismini Lourentzou[3†]
Arka Daw[2†] Anuj Karpatne[1†‡]
[1]Virginia Tech [2]Amazon [3]University of Illinois Urbana-Champaign
{amartya, karpatne}@vt.edu

Abstract

Scene-Graph Generation (SGG) seeks to recognize objects in an image and distill their salient pairwise relationships. Most methods depend on dataset-specific supervision to learn the variety of interactions, restricting their usefulness in open-world settings, involving novel objects and/or relations. Even methods that leverage large Vision Language Models (VLMs) typically require benchmark-specific fine-tuning. We introduce Open-World SGG, a training-free, efficient, model-agnostic framework that taps directly into the pretrained knowledge of VLMs to produce scene graphs with zero additional learning. Casting SGG as a zero-shot structured-reasoning problem, our method combines multi-modal prompting, embedding alignment, and a lightweight pair-refinement strategy, enabling inference over unseen object vocabularies and relation sets. To assess this setting, we formalize an Open-World evaluation protocol that measures performance when no SGG-specific data have been observed either in terms of objects and relations. Experiments on Visual Genome, Open Images V6, and the Panoptic Scene Graph (PSG) dataset demonstrate the capacity of pretrained VLMs to perform relational understanding without task-level training.

1. Introduction

Scene Graph Generation (SGG) aims to convert an image into a structured graph, where nodes correspond to object entities and edges capture the semantic relationships between them. This intermediate representation enables structured reasoning over visual content and has been shown to benefit a range of downstream tasks, including image captioning, visual question answering, and referring expression generation [11, 13, 35, 42]. Achieving accurate SGG requires a strong understanding of both visual appearance and the contextual semantics that govern interactions between objects.

Traditional approaches to SGG are predominantly supervised, trained on datasets like Visual Genome that provide dense annotations of object and relationship triplets. While these models have made significant progress, they are fundamentally constrained by the scope and vocabulary of the annotated data. Annotation is labor-intensive and often biased, and the long-tail distribution of object and predicate classes further hampers generalization to complex, real-world imagery [28, 45]. To address these limitations, recent efforts have explored Open-Vocabulary SGG (OV-SGG), where the goal is to predict objects or relations that were not seen during training. This includes tasks such as Open-Vocabulary Object Detection (OVD) – detecting unseen object categories - and Open-Vocabulary Relationships (OVR) – predicting unseen predicates between known object pairs [8]. More recent formulations consider fully open-world settings where objects or predicates may be novel at inference time [3, 27]. However, even these models rely on fine-tuning or auxiliary training stages, limiting their adaptability and requiring curated data.

Given the rapid advancements of Vision-Language Models (VLMs) trained on massive image-text corpora [1, 5, 25, 29, 34], a natural question arises: *Can VLMs enable zero-shot scene graph generation without requiring task specific training?* These models demonstrate strong visual and language generalization, and recent work has proposed reformulating SGG sub-tasks – particularly predicate classification – as image-text matching problems. Yet, most of these approaches still incorporate dataset-specific modules or rely on expensive pairwise inference pipelines, therefore not evaluating VLMs on a truly open-world setting.

Despite the growing interest in leveraging VLMs for SGG, progress in evaluating these models for SGG has been hindered by several key limitations. *First*, there is a lack of standardized baselines for open-world SGG, making it difficult to assess how well models generalize to unseen categories. *Second*, there is no established methodology for prompting VLMs to generate scene graphs in a way that is

[*] Denotes equal contribution, [†] denotes equal advising and [‡] denotes the corresponding authors.

both effective and comparable with existing methods. *Third*, the open-ended nature of VLM outputs makes it nontrivial to elicit structured predictions like subject–predicate–object triplets, which would enable their outputs to be quantitatively evaluated with popular SGG datasets, like Visual Genome [16], Panoptic Scene Graph (PSG) [40] and OpenImage (OI) [17].

To bridge this gap, we present Open World SGG (OwSGG) an end-to-end, model-agnostic framework for zero-shot scene graph generation using pretrained VLMs. Our method combines multimodal prompting, embedding alignment, and a lightweight pair-refinement strategy to transform raw VLM outputs into structured scene graphs that are compatible with existing evaluation protocols, enabling quantitative benchmarking without requiring any task-specific training. Using this framework, we conduct a comprehensive evaluation of two popular VLMs, LLaVa-next [25] and Qwen2-VL [37] – across a range of settings – including closed vocabulary, open-vocabulary objects and open-vocabulary relationships. We further introduce a fully open-world case. While we do not obtain the best results in the closed-world setting, we find that VLMs, despite no access to task-specific training, can match or even surpass these models in certain open-world cases. To encourage further research, we introduce an open-world baseline that isolates the performance of VLMs on novel object AND novel relation pairs, providing a new point of comparison for future methods. Our findings highlight the potential of pretrained vision-language models for scalable scene graph understanding and underscore the need for new methods and benchmarks tailored to the open-world setting.

2. Related Works

Scene Graph Generation introduced in [16], aims to localize, classify and predict relationships between entities in images, enabling structured visual understanding. Early works in SGG [13, 38] predict pairwise relationships between objects to construct graphs and demonstrate various scene graph applications, such as Image Captioning and VQA [10, 11]. [45] improve SGG by higher-order "motifs" in the object–predicate–object statistics, yielding large gains. However, these methods rely on fully supervised training with labeled scene graph data, which is expensive to annotate, difficult to scale to the wide variety of objects and relationships that can exist in natural scenes, and can suffer from data imbalance. To mitigate these, weakly and semi-supervised approaches [15] and imbalanced learning strategies [4] have been explored, but they remain restricted to a closed vocabulary of relationships seen during training. Recent work has introduced open-vocabulary SGG (OV-SGG), in two primary settings: Open-vocabulary Detection (OvD), which predicts known predicates between unseen object categories [3, 9, 44, 47], and Open-vocabulary Relationships (OvR), which classifies unseen predicates between known object categories [3, 44]. A more recent line of work addresses the combined OvD+OvR setting, though still relying on task-specific supervision [3, 47]. Tackling a truly open-world, training-free SGG setting – where models generate scene graphs without any fine-tuning or supervision was previously infeasible due to limitations in model capabilities. However, recent progress in zero-shot object detection, language modeling, and vision-language models (VLMs) now makes it possible to evaluate whether such models can construct scene graphs without any fine-tuning, enabling a truly open-world and training-free approach to structured visual understanding.

Vision-Language Models (VLMs) for Scene Graph Generation have become a popular choice for SGG in recent times, given the large amount of pre-trained knowledge VLMs have. Early VLMs such as [19, 20, 30] have shown promising performance over image and text modalities. This encouraged the development of the more recent VLMs [1, 5, 25, 29, 34] which have shown promising performance and have become popular as foundation models capable of several Vision and Language tasks. [3, 22] have used pre-trained VLMs for predicting Open Vocabualry SG relations. While these methods do utilise the knowledge of VLMs, they still include some dataset or task-specific training [2, 39]. Besides, just using the language priors of VLMs for predicting unseen relationships, [6, 27] also use VLMs and LLMs for object pair refinement before predicting relationships between them. This is an expensive process ranging in the $\mathcal{O}(n^2)$ operations. We address these limitations by proposing a lightweight pair refinement module in our evaluation framework that enables the use of off-the-shelf VLMs in a training free and lightweight manner to generate and evaluate SGs in an Open-World setting.

3. Open-world Scene Graph Generation (Ow-SGG)

In this section, we present our proposed framework for Open-world Scene Graph Generation (SGG) using Vision-Language Models (VLMs). We begin by introducing some background of SGG and notations used throughout this work. We then introduce the taxonomy of problem setups within the Open-world SGG (Ow-SGG) paradigm, delineating the novel challenges for each one of them. Finally, we describe our proposed framework for leveraging VLMs for SGG, highlighting how such models can be used in open-world settings.

3.1. Background and Notations

The goal of Scene Graph Generation (SGG) is to construct a structured graph-based representation of an image's visual content, termed as a scene graph. Formally, a scene graph is defined as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{\mathbf{v}_i\}_{i=1}^{N}$ is the set of nodes representing objects, and $\mathcal{E} = \{\mathbf{e_{ij}}\} \subseteq \mathcal{V} \times \mathcal{V} \times \mathcal{R}$ is the set of directed edges encoding pairwise relationships between objects, $r_{ij} \in \mathcal{R}$, where $\mathcal{R}$ is a fixed set of relation (predicate) classes.

Every object $\mathbf{v_i} = (\mathbf{b_i}, o_i)$ is associated with a class label $o_i \in \mathcal{O}$, where $\mathcal{O}$ is the predefined set of object categories, and a bounding box $\mathbf{b_i} \in \mathbb{R}^4$, which specifies its spatial location within the image. Each directed edge $\mathbf{e_{ij}} = (\mathbf{v_i}, \mathbf{v_j}, r_{ij})$ represents a relationship from object $\mathbf{v_i}$ to object $\mathbf{v_j}$, labeled by $r_{ij} \in \mathcal{R}$. Based on the amount of information available for generating an image's scene graph, there are two problem formulations of SGG that we consider in our work.

Predicate Classification (PredCls). In this problem formulation, we are given an image $I \in \mathbb{R}^{H \times W \times C}$ and the set of objects, $\mathcal{V}$, present in the image. The goal of PredCls is then to predict $\mathcal{E}$, i.e., the object pairs $(\mathbf{v_i}, \mathbf{v_j})$ present in the scene graph and its predicate class r_{ij}. Note that PredCls requires ground-truth knowledge of the set of objects $\mathcal{V}$ present in the image and hence can be considered as a restricted problem setting of SGG.

Scene Graph Detection (SGDet). Given an image I, the goal of the SGDet is to detect both the set of objects, $\mathcal{V}$, and the semantic relationships, $\mathcal{E}$, between all object pairs in the scene graph. The output is a set of triplets, $\mathcal{E} = \{(\mathbf{v_i}, \mathbf{v_j}, r_{ij}) \mid \mathbf{i} \neq \mathbf{j},\ r_{ij} \in \mathcal{R},\ \mathbf{v_i}, \mathbf{v_j} \in \mathcal{V}\}$, that compactly represent the structured content of the image. SGDet explores a more general problem setting of SGG than PredCls.

3.2. Open-world Taxonomy

We consider a range of task settings in SGG to benchmark the performance of new and existing methods in open-world settings. These tasks are defined by the novelty of triplet components $(o_i, o_j, r_{ij}) \in \mathcal{E}_{\text{test}}$ observed during testing, with respect to the set of training triplets $\mathcal{E}_{\text{train}}$ in terms of novel objects, novel relations, or both.

Close Vocabulary (CS). All triplets in this setting consist of object pairs and predicate combinations that have been observed during training. Formally, for every (o_i, o_j, r_{ij}) in $\mathcal{E}_{\text{test}}$, we have seen a different instance of (o_i, o_j, r_{ij}) in $\mathcal{E}_{\text{train}}$.

Zero-Shot (ZS). In this setting, the full triplet combination $(o_i, o_j, r_{ij}) \in \mathcal{E}_{\text{test}}$ has not been seen during training, but individual components are known, i.e., while $(o_i, o_j, r_{ij}) \notin \mathcal{E}_{\text{train}}$, $o_i, o_j \in \mathcal{O}_{\text{train}}$ and $r_{ij} \in \mathcal{R}_{\text{train}}$, where $\mathcal{O}_{\text{train}}$ and $\mathcal{R}_{\text{train}}$ denote the set of observed objects and relation classes.

Open-Vocabulary Relations (OVR). This open-world task setting of SGG explores the scenario where object classes are known but we are interested in detecting predicate classes that we have never seen before during training. Formally, for every $(o_i, o_j, r_{ij}) \in \mathcal{E}_{\text{test}}$, while $o_i, o_j \in \mathcal{O}_{\text{train}}$, $r_{ij} \notin \mathcal{R}_{\text{train}}$.

Open-Vocabulary Detections (OVD). Here we consider the scenario where object classes are novel but the relation classes are known, i.e., for every $(o_i, o_j, r_{ij}) \in \mathcal{E}_{\text{test}}$, while $r_{ij} \in \mathcal{R}_{\text{train}}$, $o_i, o_j \notin \mathcal{O}_{\text{train}}$.

Open-Vocabulary Detections + Relations (OVD+R). This open-world task setting considers a union of OVD and OVR where either the object classes are novel OR the predicate classes are novel. Formally, for every $(o_i, o_j, r_{ij}) \in \mathcal{E}_{\text{test}}$, we have $(o_i, o_j \notin \mathcal{O}_{\text{train}}) \vee (r_{ij} \notin \mathcal{R}_{\text{train}})$.

Open World (OW). A special case of OVD+R setting is when both object classes AND predicate classes are novel. This represents the most challenging setup that we refer to as the strictly Open World (OW) setting, formally defined as $\forall\ (o_i, o_j, r_{ij}) \in \mathcal{E}_{\text{test}},\ (o_i, o_j \notin \mathcal{O}_{\text{train}}) \wedge (r_{ij} \notin \mathcal{R}_{\text{train}})$.

3.3. Proposed Framework for Using VLMs for Ow-SGG

Figure 1 shows our framework for Ow-SGG using VLMs, comprising of the following five steps.

3.3.1. Entity Generation

The goal of this step is to enumerate a diverse set of potential entities present in an image by prompting vision-language models (VLMs). We simply prompt a VLM to generate candidate object classes (or entities) present in an image.

3.3.2. Entity Mapping

In this step, we consider the task of mapping entities generated by VLMs to known object categories. Note that the entities predicted by a VLM may not directly correspond to the predefined or canonical class labels of objects in the dataset, and may occasionally include paraphrased or hallucinated concepts. Consequently, further processing is required to semantically align these free-form predictions of entity names with the predefined dataset object names. To achieve this, the entity mapping module uses an embedding-based matching mechanism to associate predicted entities to the dataset object categories. To identify semantically similar categories for every predicted entity, we compute similarity scores using a contrastive text encoder (SimCSE [7]) and rank the candidates using a softmax-based scoring strategy. In practice, we retain up to k dataset object categories whose similarity scores fall within a δ-neighborhood of the top match with respect to any predicted entity. This strategy enhances robustness to synonyms or ambiguous naming (e.g., "man" vs. "boy") while maintaining high semantic alignment. At the end of this process, every predicted entity is associated with one or more dataset object categories, enabling consistent downstream reasoning. The complete procedure is detailed in the `Semantic Pair Scoring for Entity Mapping` box below.

3.3.3. Entity Detection

Once the VLM-generated entities have been mapped to a list of candidate objects, Grounding DINO [26] is used to

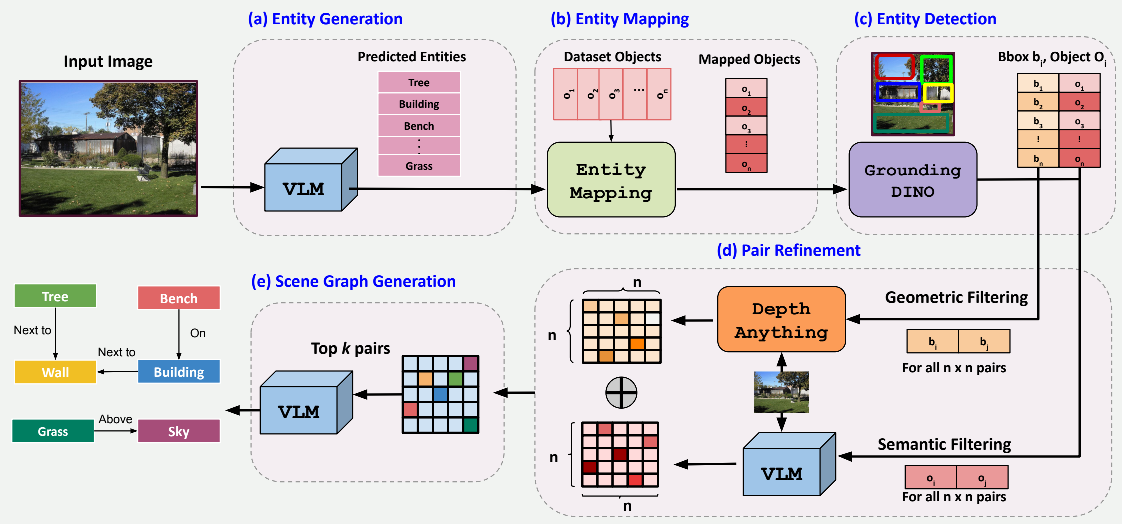


Figure 1. Overview of our proposed framework for open-world SGG using VLMs.

localize different instances of every object by predicting its corresponding bounding box inside the image. This step also serves as another layer of refinement by ignoring entities predicted but otherwise not present in the image.

3.3.4. Pair Refinement

Given the set of object proposals $\{(\mathbf{b_i}, o_i)\}_{i=1}^{N}$ produced by the Object Detection module, the next step in the pipeline involves constructing a list of meaningful object pairs $(\mathbf{v_i}, \mathbf{v_j})$ for relational inference. A naïve approach is to select only overlapping or spatially adjacent objects. However, this has the risk of missing crucial object interactions that are semantically meaningful but show small geometric overlap (e.g., object pairs with relationships such as `looking at`). Conversely, enumerating all $N(N-1)$ possible directed object pairs is computationally expensive and introduces a high degree of noise, especially when, 1) the resulting sequence becomes too long to be processed by most VLMs, and 2) many object pairs are semantically or spatially irrelevant.

We propose a *pair refinement module* that combines semantic and geometric cues to generate a concise yet informative set of candidate object pairs for relationship prediction before directly pruning out the object pairs. The module comprises two branches – semantic pair refinement and geometric pair refinement – which are then fused to produce a final refinement map.

Semantic pair refinement utilizes a VLM to estimate the semantic closeness of interactions between every object pair. Given a pair (o_i, o_j) of object category labels, the VLM assigns a semantic compatibility score:

$$\mathbf{P^S_{ij}} = \texttt{VLM}(o_i, o_j) \in [0, 1], \tag{1}$$

where $\mathbf{P^S_{ij}}$ is the semantic compatibility between objects i and j. Since the VLM score is conditioned only on the object types (not their specific locations), we assign the same $\mathbf{P^S_{ij}}$ to all instances of an object class pair (o_i, o_j). This results in a semantic pair matrix $\mathbf{P^S} \in \mathbb{R}^{N \times N}$.

Semantic Pair Scoring for Entity Mapping

Let p be a predicted entity, $D = \{d_1, \ldots, d_N\}$ the dataset object categories, $h(\cdot)$ a SimCSE encoder, $\tau = 0.2$ the temperature, $\delta = 0.05$ the threshold, and $k = 2$ the maximum number of retained dataset object categories per entity.

1. Compute similarity scores:

$$s_i = \cos\big(h(\text{"There is a } p \text{ in the image."}), \\ h(\text{"There is a } d_i \text{ in the image."})\big)$$

2. Apply temperature-scaled softmax:

$$\hat{s}_i = \frac{\exp(s_i/\tau)}{\sum_{j=1}^{N} \exp(s_j/\tau)}$$

3. Define near-maximum set:

$$S_{\max} = \max_i \hat{s}_i, \quad C = \{i : S_{\max} - \hat{s}_i < \delta\}$$

4. Return top-k object labels from C:

$$M(p) = \{d_i \mid i \in C \text{ and } \hat{s}_i \text{ in top-}k \text{ of } C\}$$

5. Combine mappings across all predicted entities:

$\mathcal{M} = \bigcup_{p \in \mathcal{P}} M(p)$, where $\mathcal{P}$ is the set of predicted entities.

The final output $\mathcal{M}$ contains the set of matched dataset object categories aligned to the free-form predicted entities from the VLM.

Geometric pair refinement filters out spatially implausible pairs, by estimating spatial distances between objects using their 2D bounding boxes and an estimated depth map D. Following [6], we compute the 2D center of every bounding box $\mathbf{b_i}$ as $\mathbf{c}_i^{2D} = (x_i, y_i)$, and extract its normalized median depth value $\mathbf{d_i} \in [0, 1]$ from a monocular depth map D that we generate using [41]. Given two objects o_i and o_j, we first compute the Euclidean distance between their 2D centers, $\mathbf{x_{ij}}$ $(= \|\mathbf{c}_i^{2D} - \mathbf{c}_j^{2D}\|_2)$, and normalize it by the image diagonal length $y = \sqrt{H^2 + W^2}$. The total distance between the objects is defined as a weighted combination of normalized 2D distance and absolute depth difference,

$$\mathbf{d_{ij}} = \lambda_1 \left(\frac{\mathbf{x_{ij}}}{y} \right) + \lambda_2 ||\mathbf{d_i} - \mathbf{d_j}||_2, \tag{2}$$

where $\lambda_1, \lambda_2 > 0$ are hyperparameters that control the relative contributions of the 2D and 3D components. A pair is retained if $\mathbf{d_{ij}} < \tau$, for a chosen threshold $\tau > 0$. To enable soft filtering, we convert this into a compatibility score using a sigmoid function,

$$\mathbf{P}_{\mathbf{ij}}^{\mathbf{G}} = \sigma\left(-\beta(\mathbf{d_{ij}} - \tau)\right), \tag{3}$$

where β is an inverse-temperature parameter that controls the sharpness of the score, with higher values making the sigmoid more sensitive to whether the distance d_{ij} is above or below the threshold τ. The resulting matrix $\mathbf{P}^{\mathbf{G}} \in \mathbb{R}^{N \times N}$ softly encodes spatial plausibility of each object pair.

Fusion of Semantic and Geometric Maps. Finally, we generate a unified refinement map $\mathbf{P}^{\text{final}} \in \mathbb{R}^{N \times N}$ by taking a weighted sum of the semantic and geometric scores:

$$\mathbf{P}_{\mathbf{ij}}^{\text{combined}} = \alpha \log \mathbf{P}_{\mathbf{ij}}^{\mathbf{S}} + (1 - \alpha) \log \mathbf{P}_{\mathbf{ij}}^{\mathbf{G}}, \tag{4}$$

where $\alpha \in [0, 1]$ is a tunable coefficient that controls the relative importance of semantic versus geometric refinement. Top-k pairs with the highest $\mathbf{P}_{\mathbf{ij}}^{\text{combined}}$ scores are retained for downstream relation prediction, resulting in a cleaner, more meaningful set of candidate triplets.

3.3.5. Scene Graph Generation

We pass the set of refined pairs obtained from the previous module to a Vision-Language Model (VLM), which is prompted to predict the corresponding relationship between the two entities given the input image. This results in a set of relational triplets that form the final scene graph.

4. Experimental Results

Evaluation Metrics. Having obtained the relationships R and confidence scores S for each object pair across the dataset, we measure performance against existing methods using two standard SG metrics: *Recall@K (R@K)* and *mean Recall@K (mR@K)*. These metrics quantify the proportion of ground-truth relationships a model is able to retrieve in its top-K predictions, either considering all predicate categories as a whole ($R@K$) or on a category-by-category basis ($mR@K$).

Datasets: We perform extensive evaluations on the VG150 Dataset [16], Open Image v6 (OIV6) [17], and the Panoptic Scene Graph Generation Dataset (PSG) [40]. The VG150 has 150 objects and 50 relationship categories. The OIV6 has 601 objects and 30 relationship categories while the PSG dataset has 133 object and 56 relationship categories.

Backbone and Baselines: We consider the following VLM backbones for implementing the proposed approach - LlaVa-next 7b [24], Qwen2-vl 7b and Qwen2-vl 72b [37]. We choose these VLMs as they are general foundational models and we evaluate them to exhibit the potential of our framework in being a model-agnostic SG Evaluation framework. We compare our evaluations against some well-known SOTA baselines such as - PGSG [22], SGTR [21], RAHP [27], OvSGTR [3].

4.1. Open Vocabulary Relationship (OVR) Results

We evaluate our framework on the Open-Vocabulary Relationship Prediction (OVR) task, which assesses a model's ability to correctly identify predicates that are absent from the training set. This task measures a model's capacity to generalize to rare or unseen relationships. Since our framework operates without any task-specific training, it is particularly well-suited for open-vocabulary settings. In contrast to conventional scene graph generation (SGG) models—which often struggle with unseen predicates due to their dependence on fixed label spaces—our approach leverages the semantic priors of vision-language models to reason over a broader predicate space. As shown in Table 2, our Qwen2-72B-based framework outperforms baseline methods on the OVR task for the PSG dataset. On the Visual Genome (VG) dataset under the PredCls setting, the OwSGG Qwen2-72B model achieves performance comparable to the best existing model. However, in most other settings—particularly on VG—our models underperform relative to baselines. These results suggest that vision-language models can generalize well in simpler but face challenges when applied to more complex or varied data.

4.2. Close Vocabulary and Zero-Shot Results

We also evaluate our open-world framework in the standard closed-vocabulary scene graph generation (SGG) setting, and additionally report zero-shot performance within this setup, as shown in Table 1. While our models are not expected to outperform baselines trained explicitly on

Table 1. **Close Vocabulary SGG Performance on VG150, OIV6, and PSG**: We show Zero-Shot and Close Vocabulary results on the VG150, OIV6 and the PSG Dataset. We compare our results on both SgDet and PredCls for VG150 and OIV6 and only SgDet for PSG.

		Method Name	Close Vocabulary		Zero-Shot
			mR @ 20 / 50 / 100	R @ 20 / 50 / 100	R @ 20 / 50 / 100
VG	PredCls	IMP [38]	11.7 / 14.8 / 16.1	– / 44.8 / 53.1	–
		MOTIFS[45]	11.7 / 14.8 / 16.1	58.5 / 65.2 / 67.1	– / 10.9 /14.5
		VCTree+HIERCOM [12]	17.6 / 26.3 / 31.8	55.9 / 69.8 / 75.8	– / 17.8 / 24.8
		CooK [14]	– / 33.7 / 35.8	– / 62.1 / 64.2	–
		CaCao [43]	36.2 / 31.7 / 43.7	–	–
		OwSGG (llava-next)	9.74 / 14.96 / 19.26	9.72 / 14.87 / 19.08	3.99 / 6.74 / 10.02
		OwSGG (Qwen7b)	4.82 / 8.73 / 12.64	4.9 / 8.9 /12.87	2.67 / 5.77 / 8.82
		OwSGG (Qwen72b)	7.63 / 13.54 / 19.76	7.53 / 13.44 / 19.64	3.3 / 6.52 / 9.7
	SGDet	SSRCNN [36]	– / 18.6 / 22.5	– / 23.7 / 27.3	– / 3.1 / 4.5
		SGTR [21]	– / 12.0 / 15.2	–/ 24.6 / 28.4	– / 2.5 / 5.8
		PGSG [22]	– / 8.9 / 11.5	– / 16.7 / 21.2	– / 6.2 / 8.5
		OwSGG (llava-next)	1.88 / 2.89 / 3.7	1.7 / 2.61 / 3.36	0.98 / 1.71 / 2.36
		OwSGG (Qwen7b)	0.67 / 1.15 / 1.71	0.64 / 1.09 / 1.61	0.48 / 0.91 / 1.32
		OwSGG (Qwen72b)	1.38/ 2.43 / 3.4	1.3 / 2.28 / 3.18	0.73 / 1.24 /1.95
OI6	PredCls	SGTR [21]	–	– / 59.9 / –	–
		ReIDN [46]	–	– / 72.8 / –	–
		GPS-Net [23]	–	– / 74.7 / –	–
		HEIRCOM [12]	–	– / 85.4 / –	–
		OwSGG (llava-next)	59.92 / 66.82 / 70.24	59.88 / 66.81 / 70.22	30.08 / 35.03 / 36.59
		OwSGG (Qwen7b)	56.91 / 67.59 / 73.51	56.88 / 67.6 / 73.47	26.82 / 33.46 / 36.59
		OwSGG (Qwen72b)	71.54 / 79.83 / 83.76	71.56 / 79.86 / 83.78	40.1 / 47.14 / 47.14
	SGDet	SGTR [21]	– /38.6 / –	– / 59.1 / –	– / 19.4 / 31.6
		PGSG [22]	– / 8.9 / 11.5	– / 16.7 / 21.2	– / 23.1 / 38.6
		OwSGG (llava-next)	7.93 / 9.6 / 11.21	7.9 / 9.55 / 11.16	2.34 / 4.04 / 6.77
		OwSGG (Qwen7b)	2.7 / 4.61 / 6.74	2.68 / 4.59 / 6.71	2.47 / 2.99 / 2.99
		OwSGG (Qwen72b)	6.68 / 8.8 / 10.82	6.65 / 8.75 / 10.75	3.52 / 3.52 / 4.3
PSG	SGDet	PSGTR [40]	– / 20.3 / 21.5	– / 32.1 / 35.3	– / 3.1 / 6.4
		SGTR [21]	– / 24.3 / 27.2	– / 33.1 / 36.3	– / 4.1 / 5.8
		PGSG [22]	– / 20.9 / 22.1	– / 32.7 / 33.4	– / 6.8 / 8.9
		OwSGG (llava-next)	5.59 / 8.12 / 10.02	5.63 / 8.12 / 10.08	1.84 / 2.51 / 4.68
		OwSGG (Qwen7b)	3.71 / 6.13 / 8.47	3.93 / 6.34 / 8.59	1.84 / 3.34 / 5.3
		OwSGG (Qwen72b)	7.22 / 10.49 / 13.67	7.36 / 10.68 / 13.98	2.84 / 4.68 / 6.44

Table 2. **Open Vocabulary Relation SGG Performance on VG150 and PSG**: We show OVR results on the VG150 and the PSG Dataset. We compare our results on both SgDet & PredCls.

		VG		PSG	
	Method Name	novel (Relation)		novel (Relation)	
		mR @ 50 / 100	R @ 50 / 100	mR @ 50 / 100	R @ 50 / 100
SGDet	VS3+RAHP [27]	–	3.75 / 5.12	–	–
	OvSGTR [3]	1.82 / 2.32	13.45 / 16.19	–	–
	OvSGTR+RAHP [3]	3.01 / 4.04	15.59 / 19.92	–	–
	PGSG [22]	3.7 / 5.2	–	7.4 / 11.3	–
	SGTR [21]	0.0 / 0.0	–	0.0 / 0.0	0.0 / 0.0
	OwSGG (LLaVA-next)	2.34 / 3.04	2.33 / 3.04	8.27 / 10.4	8.31 / 10.49
	OwSGG (Qwen7b)	1.14 / 1.67	1.15 / 1.67	5.77 / 7.51	5.93 / 7.6
	OwSGG (Qwen72b)	2.19 / 3.07	2.19 / 3.06	10.25 / 13.35	10.42 / 13.54
PredCls	CaCao [43]	–	7.4 / 9.7	–	–
	PGSG [22]	5.2 / 7.7	–	–	–
	SGTR+RAHP [27]	11.82 / 15.46	15.46 / 20.37	–	–
	OwSGG (LLaVA-next)	0.75 / 1.36 / 1.5	0.74 / 1.36 / 1.5	4.82 / 5.77	4.89 / 5.81
	OwSGG (Qwen7b)	0.44 / 1.2 / 2.12	0.44 / 1.19 / 2.11	4.02 / 5.29	4.03 / 5.31
	OwSGG (Qwen72b)	7.64 / 11.04	7.62 / 11.02	6.36 / 7.68	6.34 / 7.69

dataset-specific labels, they yield several notable results. On the OIV6 dataset under the PredCls setting, the OwSGG (Qwen2-72B) model achieves higher recall at R@50 than all baselines except HEIRCOM [12]. As anticipated, our framework shows a relative advantage in the zero-shot scenario, where conventional models often fail to generalize beyond their training vocabulary. This is evident in the SgDet setting, where OwSGG (Qwen2-72B) surpasses all models except PGSG at R@100. However, on the more complex VG dataset, our models struggle to match baseline performance under both PredCls and SgDet settings.

Table 3. **Open Vocabulary Detection and Open World SGG Performance on VG150**: We show results for the SgDet task on the VG150 Dataset. † indicates that the results were generated for this work.

Method Name	OVD + R		OW
	novel (Object)	novel (Relation)	novel (Object & Relation)
	R@50 / R@100	R@50 / R@100	R@50 / R@100
IMP [38]	0.00 / 0.00	0.00 / 0.00	0.00 / 0.00†
MOTIFS [45]	0.00 / 0.00	0.00 / 0.00	0.00 / 0.00†
VCTREE [32]	0.00 / 0.00	0.00 / 0.00	0.00 / 0.00†
TDE [33]	0.00 / 0.00	0.00 / 0.00	0.00 / 0.00†
VS3 [48]	6.00 / 7.51	0.00 / 0.00	–
OvSGTR (Swin-B) [3]	17.58 / 21.72	14.56 / 18.20	5.97 / 10.06†
VS3+RAHP [27]	13.01 / 14.82	3.75 / 5.12	–
OvSGTR+RAHP (Swin-T) [27]	12.45 / 15.38	13.31 / 16.46	–
OwSGG (LLaVA-next)	2.37 / 3.07	2.33 / 3.04	1.92 / 2.56
OwSGG (Qwen7b)	0.87 / 1.28	1.15 / 1.67	0.86 / 1.18
OwSGG (Qwen72b)	1.88 / 2.73	2.19 / 3.06	1.61 / 2.41

4.3. Open Vocabulary Detection + Relation based SGG (OvD+R) and Open World Results

The **OvD+R** setting evaluates models trained exclusively on base classes of objects and relationships, but tested on either novel objects or novel relationships—never both simultaneously. This setup measures partial generalization, where some components of the scene graph remain within the training distribution. In contrast, we introduce a more stringent **Open-World (OW)** evaluation setting, in which models must reason about both unseen objects and unseen relationships at test time, without any task-specific fine-tuning. The corresponding results are presented in Tab. 3. This scenario more closely reflects real-world conditions and provides a rigorous assessment of a model's compositional generalization and robustness. Our proposed framework is explicitly designed for this open-world regime, leveraging the semantic flexibility of vision-language models without relying on supervision from predefined label sets. While the OwSGG results are still lower than those of baseline models trained with access to closed-world labels, they demonstrate the potential of this approach. The goal of establishing the OW baseline is to motivate future work toward models capable of operating effectively in fully unknown environments.

4.4. Ablation Results

Fig. 2 (a) illustrates how varying the hyperparameter α, with a fixed `top_k` of 25, influences model performance. As expected, increasing `top_k` generally improves recall by increasing the likelihood of retrieving Ground Truth object pairs. The F1 score here reflects the quality of pair refinement—Recall captures how well Ground Truth pairs are preserved, while Precision indicates the degree to which noisy

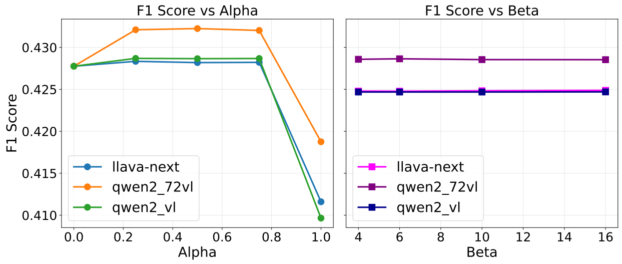


Figure 2. Ablation Study: F1 scores across different (a) α and (b) β values for the Qwen-72B model.

Table 4. Effect of depth and semantic filtering on PSG dataset (PredCls task, Qwen72b model). Bold = best, underline = 2^{nd} best.

Setup	Depth	Semantic	R@20/50/100	mR@20/50/100
CS	✓	✗	**4.8 / 6.94 / 8.32**	**4.76 / 6.86 / 8.22**
	✗	✓	2.67 / 4.03 / 5.09	2.61 / 4.0 / 5.05
	✓	✓	4.76 / 6.02 / 6.91	**4.76**/ 6.02 / 6.91
ZS	✓	✗	2.06 / **2.9** / **4.35**	2.09 /**2.9** / **4.35**
	✗	✓	1.34 / 2.17 / 2.17	1.34 / 2.17 / 2.17
	✓	✓	**2.34** / 2.68 / 3.51	**2.34** / 2.68 / 3.51
OVR	✓	✗	4.36 / 6.09 / 7.47	4.4 / 6.17 / 7.55
	✗	✓	2.03 / 3.56 / 4.52	2.12 / 3.65 / 4.61
	✓	✓	**4.88 / 6.34 / 7.69**	**4.88 / 6.36 / 7.68**

pairs are filtered out. An $\alpha = 0$ corresponds to pure depth-based refinement, whereas $\alpha = 1$ denotes purely semantic refinement. The results suggest that a balanced combination of both semantic and depth cues leads to more effective pair refinement for all the models. Fig. 2 (b) presents further ablation results, showing how the F1 score varies with different values of the β parameter used during depth-based refinement. These results highlight the model's ability to retain meaningful pairs under a fixed `top_k` of 25. Tab. 4 complements these findings by reporting Triplet Recall values across various pair refinement strategies. We observe that while depth-only refinement performs well in a closed-vocabulary setting, combining semantic and depth-based filtering yields consistently better performance as the evaluation setting becomes more open and data-limited.

5. Limitations and Conclusions

The OwSGG framework leverages several pre-trained components—such as Grounding-DINO for object detection and SimCSE for embedding similarity—which can introduce error at various stages of the scene graph generation (SGG) pipeline. Understanding the contribution of these sources of error is an important direction for future work. Additionally, our method constructs textual prompts by pairing detected objects and feeding them into vision-language models (VLMs), which inherently limits scalability due to the context length constraints of these models. Future research can explore more efficient pair refinement strategies to reduce the number of candidate pairs that require evaluation by the VLM. Despite these limitations, our results demonstrate that VLMs, when guided by prompts and supplemented with object detection and embedding-based modules, are capable of predicting scene graph relationships without any task-specific training. This underscores the potential of zero-shot methods for structured vision-language tasks and paves the way toward more general, flexible, and interpretable visual reasoning systems.

References

[1] Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-vl: A frontier large vision-language model with versatile abilities. *arXiv preprint arXiv:2308.12966*, 2023.

[2] Zuyao Chen, Jinlin Wu, Zhen Lei, Zhaoxiang Zhang, and Changwen Chen. Gpt4sgg: Synthesizing scene graphs from holistic and region-specific narratives. *arXiv preprint arXiv:2312.04314*, 2023.

[3] Zuyao Chen, Jinlin Wu, Zhen Lei, Zhaoxiang Zhang, and Chang Wen Chen. Expanding scene graph boundaries: fully open-vocabulary scene graph generation via visual-concept alignment and retention. In *European Conference on Computer Vision*, pages 108–124. Springer, 2024.

[4] Yuren Cong, Michael Ying Yang, and Bodo Rosenhahn. Reltr: Relation transformer for scene graph generation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(9): 11169–11183, 2023.

[5] Matt Deitke, Christopher Clark, Sangho Lee, Rohun Tripathi, Yue Yang, Jae Sung Park, Mohammadreza Salehi, Niklas Muennighoff, Kyle Lo, Luca Soldaini, et al. Molmo and pixmo: Open weights and open data for state-of-the-art multimodal models. *arXiv preprint arXiv:2409.17146*, 2024.

[6] Abdelrahman Elskhawy, Mengze Li, Nassir Navab, and Benjamin Busam. Prism-0: A predicate-rich scene graph generation framework for zero-shot open-vocabulary tasks. *arXiv preprint arXiv:2504.00844*, 2025.

[7] Tianyu Gao, Xingcheng Yao, and Danqi Chen. Simcse: Simple contrastive learning of sentence embeddings. *arXiv preprint arXiv:2104.08821*, 2021.

[8] Xiuye Gu, Tsung-Yi Lin, Weicheng Kuo, and Yin Cui. Open-vocabulary object detection via vision and language knowledge distillation. *arXiv preprint arXiv:2104.13921*, 2021.

[9] Tao He, Lianli Gao, Jingkuan Song, and Yuan-Fang Li. Towards open-vocabulary scene graph generation with prompt-based finetuning. In *European Conference on Computer Vision*, pages 56–73. Springer, 2022.

[10] Marcel Hildebrandt, Hang Li, Rajat Koner, Volker Tresp, and Stephan Günnemann. Scene graph reasoning for visual question answering. *arXiv preprint arXiv:2007.01072*, 2020.

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

[12] Bowen Jiang, Zhijun Zhuang, Shreyas S Shivakumar, and Camillo J Taylor. Enhancing scene graph generation with hierarchical relationships and commonsense knowledge. In

2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), pages 8883–8894. IEEE, 2025.

[13] Justin Johnson, Ranjay Krishna, Larry Stark, Jia Li, David A Shamma, Michael Bernstein, and Li Fei-Fei. Image retrieval using scene graphs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3668–3678, 2015.

[14] Hyeongjin Kim, Sangwon Kim, Dasom Ahn, Jong Taek Lee, and Byoung Chul Ko. Scene graph generation strategy with co-occurrence knowledge and learnable term frequency. *arXiv preprint arXiv:2405.12648*, 2024.

[15] Kibum Kim, Kanghoon Yoon, Jaehyeong Jeon, Yeonjun In, Jinyoung Moon, Donghyun Kim, and Chanyoung Park. Llm4sgg: large language models for weakly supervised scene graph generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 28306–28316, 2024.

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

[17] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, et al. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *International journal of computer vision*, 128(7):1956–1981, 2020.

[18] Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*, 2023.

[19] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International conference on machine learning*, pages 12888–12900. PMLR, 2022.

[20] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pages 19730–19742. PMLR, 2023.

[21] Rongjie Li, Songyang Zhang, and Xuming He. Sgtr: End-to-end scene graph generation with transformer. In *proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 19486–19496, 2022.

[22] Rongjie Li, Songyang Zhang, Dahua Lin, Kai Chen, and Xuming He. From pixels to graphs: Open-vocabulary scene graph generation with vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 28076–28086, 2024.

[23] Xin Lin, Changxing Ding, Jinquan Zeng, and Dacheng Tao. Gps-net: Graph property sensing network for scene graph generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3746–3753, 2020.

[24] Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. Llava-next: Improved reasoning, ocr, and world knowledge, 2024.

[25] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36, 2024.

[26] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In *European Conference on Computer Vision*, pages 38–55. Springer, 2024.

[27] Tao Liu, Rongjie Li, Chongyu Wang, and Xuming He. Relation-aware hierarchical prompt for open-vocabulary scene graph generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 5576–5584, 2025.

[28] Cewu Lu, Ranjay Krishna, Michael Bernstein, and Li Fei-Fei. Visual relationship detection with language priors. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*, pages 852–869. Springer, 2016.

[29] OpenAI. "hello gpt-4", 2024.

[30] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[31] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*, 2019.

[32] Kaihua Tang, Hanwang Zhang, Baoyuan Wu, Wenhan Luo, and Wei Liu. Learning to compose dynamic tree structures for visual contexts. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6619–6628, 2019.

[33] Kaihua Tang, Yulei Niu, Jianqiang Huang, Jiaxin Shi, and Hanwang Zhang. Unbiased scene graph generation from biased training. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3716–3725, 2020.

[34] Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*, 2024.

[35] Damien Teney, Lingqiao Liu, and Anton van Den Hengel. Graph-structured representations for visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2017.

[36] Yao Teng and Limin Wang. Structured sparse r-cnn for direct scene graph generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19437–19446, 2022.

[37] Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin

Ge, et al. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024.

[38] Danfei Xu, Yuke Zhu, Christopher B Choy, and Li Fei-Fei. Scene graph generation by iterative message passing. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5410–5419, 2017.

[39] Mingjie Xu, Mengyang Wu, Yuzhi Zhao, Jason Chun Lok Li, and Weifeng Ou. Llava-spacesgg: Visual instruct tuning for open-vocabulary scene graph generation with enhanced spatial relations. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 6362–6372. IEEE, 2025.

[40] Jingkang Yang, Yi Zhe Ang, Zujin Guo, Kaiyang Zhou, Wayne Zhang, and Ziwei Liu. Panoptic scene graph generation. In *European Conference on Computer Vision*, pages 178–196. Springer, 2022.

[41] Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2. *Advances in Neural Information Processing Systems*, 37: 21875–21911, 2024.

[42] Sibei Yang, Guanbin Li, and Yizhou Yu. Cross-modal relationship inference for grounding referring expressions. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4145–4154, 2019.

[43] Qifan Yu, Juncheng Li, Yu Wu, Siliang Tang, Wei Ji, and Yueting Zhuang. Visually-prompted language model for fine-grained scene graph generation in an open world. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 21560–21571, 2023.

[44] Alireza Zareian, Kevin Dela Rosa, Derek Hao Hu, and Shih-Fu Chang. Open-vocabulary object detection using captions. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14393–14402, 2021.

[45] Rowan Zellers, Mark Yatskar, Sam Thomson, and Yejin Choi. Neural motifs: Scene graph parsing with global context. arXiv preprint arXiv:1711.06640, 2018.

[46] Ji Zhang, Kevin J Shih, Ahmed Elgammal, Andrew Tao, and Bryan Catanzaro. Graphical contrastive losses for scene graph parsing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11535–11543, 2019.

[47] Yong Zhang, Yingwei Pan, Ting Yao, Rui Huang, Tao Mei, and Chang-Wen Chen. Learning to generate language-supervised and open-vocabulary scene graph using pre-trained visual-semantic space. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2915–2924, 2023.

[48] Yong Zhang, Yingwei Pan, Ting Yao, Rui Huang, Tao Mei, and Chang-Wen Chen. 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*, pages 2915–2924, 2023.

Open World Scene Graph Generation using Vision Language Models

Supplementary Material

A. Dataset Descriptions and Evaluation Splits

Datasets We evaluate our framework on two categories of SGG: *SgDet* and *PredCls*. We evaluate on three datasets: Visual Genome (VG) [16], Open Images V6 (OIV6) [17], and Panoptic Scene Graph (PSG) [40], using the standard splits from prior work [38, 40, 45]. Since our method requires no training, we evaluate only on the test data. For VG [16], we follow the cleaning protocol of [38, 45], removing images with insufficient annotations. This yields 26,446 test images (from 32,422), covering 150 object and 50 relation classes. For OIV6 [17], we use the test split with 5,322 images, 601 object classes, and 30 relations. For PSG [40], we evaluate on the validation split, which contains 1,000 images, 133 objects, and 56 relations.

We also leverage publicly available scripts and ID lists for split generation and novelty definitions:

- **Zero-Shot Triplets** are generated using the T-CAR repository's notebook[1], which filters unseen triplets from the combined val+test pool.
- **VG Novel Predicates** (VG150) come from the OvSGTR codebase[2], and the base predicate set follows [9].
- **OIV6 Novel Objects** are defined in the Pix2Grp CVPR2024 script[3], and similarly for **PSG Novel Predicates**[4].
- For VG and OIV6, we adopt the train/val/test splits from previous works [38, 45]. For PSG, we follow the official code and splits distributed at[5].

B. Implementation Details

B.1. Vision Language Models

All VLMs used are instruction-tuned to interpret structured prompts better. For inference, we leverage the vLLM framework [18], which enables efficient execution of large-scale language models through a paged attention mechanism. Unlike traditional approaches that allocate contiguous memory, paged attention uses fixed-size pages, reducing fragmentation and improving memory reuse—allowing larger models to run with lower overhead. vLLM also features an optimized key-value (KV) cache that eliminates redundant computations by reusing previously computed attention values, significantly accelerating autoregressive generation. These optimizations make vLLM highly scalable and well-suited for low-latency inference with large VLMs. Due to hardware constraints, we quantize all models: 7B models from `float32` to `bfloat16`, and Qwen2-vl-72B using `AWQ`. This substantially reduces memory usage while maintaining performance.

B.2. Entity Generation

In the Entity Generation module, we prompt a VLM with the task of generating a comprehensive list of entities present in the input image. The module is configured using the following hyper-parameters:

1. `num_outputs=1`: We request a single generation output per image.
2. `temperature=0.1`: A low temperature ensures deterministic outputs, reducing randomness and encouraging factual extraction.
3. `max_tokens=512`
4. `top_p=1.0`: This enables nucleus sampling with a large cutoff to avoid premature truncation of less frequent but relevant entities.
5. `presence_penalty=0.4`: Penalizes repetitions to encourage novel mentions without being too aggressive.
6. `repetition_penalty=1.1`: Mildly discourages duplicate tokens during generation.

Prompt examples. We use dataset-specific prompts tailored to encourage comprehensive object enumeration. The prompt examples used for three datasets are shown by `PSG Dataset Prompt`, `Open Images (OI) Prompt` and `Visual Genome (VG) Prompt`.

[1] https://github.com/jkli1998/T-CAR/blob/main/zs_check.ipynb

[2] https://github.com/gpt4vision/OvSGTR/blob/018453e07cf04be416ac42d13e1bf27d1611678d/datasets/vg.py#L37

[3] https://github.com/SHTUPLUS/Pix2Grp_CVPR2024/blob/main/lavis/datasets/datasets/oiv6_rel_detection.py

[4] https://github.com/SHTUPLUS/Pix2Grp_CVPR2024/blob/main/lavis/datasets/datasets/psg_rel_detection.py

[5] https://github.com/franciszzj/OpenPSG

PSG Dataset Prompt

```
### Task Start You are an expert at
detecting objects in images.  You
are given an image.  Your task is
to list all objects visible in the
image, including both foreground and
background.  The objects may include
natural elements, human-made structures,
or any other discernible entities.
### Output Format Instructions
- Do not repeat object names.
- Do not describe attributes, adjectives,
or relationships.
- Return the result as a comma-separated
list.
- If unsure, include it.
### Prompt
List all the objects visible in the
image, including foreground and
background.  Return the objects as a
comma-separated list.
```

Open Images (OI) Prompt

```
### Task Start
You are an expert at detecting objects
in images.  You are given an image.
Your task is to identify and list all
visible objects in the image, including
both foreground and background.  Include
a wide range of recognizable categories,
whether specific or general, as long as
they are visibly present in the scene.
### Output Format Instructions
- Do not repeat object names.
- Do not describe attributes, adjectives,
or relationships.
- Return the result as a comma-separated
list.
- If unsure, include it.
### Prompt
List all the objects visible in the
image, including foreground and
background.  Return the objects as a
comma-separated list.
```

Visual Genome (VG) Prompt

```
### Task Start
You are an expert at detecting objects
in images.  You are given an image.
Your task is to list all identifiable
objects visible in the image, including
those in the foreground and background.
Include both whole objects and
meaningful parts or components that
are visually discernible.
### Output Format Instructions
- Do not repeat object names.
- Do not describe attributes, adjectives,
or relationships.
- Return the result as a comma-separated
list.
- If unsure, include it.
### Prompt
List all the objects visible in the
image, including foreground and
background.  Return the objects as a
comma-separated list.
```

B.3. Entity Mapping

Our entity-mapping pipeline aligns VLM-predicted object labels to a fixed ground-truth vocabulary via a three-stage cascade. First, each label is normalized (converted to lowercase, trimmed of whitespace, and stripped of all punctuation). Second, we compare the normalized prediction directly against a cache of normalized ground-truth entries; any exact hits are accepted with confidence 1.0. Third, any remaining labels are resolved via semantic matching with a contrastively pretrained SimCSE [7] encoder.

In the semantic stage, we convert each candidate label *X* into a full sentence of the form

> "There is a *X* in the image."

and embed it with SimCSE. We compare that embedding—via cosine similarity—to a cache of precomputed embeddings for every normalized ground-truth entry. To sharpen the score distribution, we apply temperature scaling with $\tau = 0.2$. We then filter out any ground-truth entries whose cosine score falls more than $\Delta = 0.05$ below the maximum observed score, and finally select the top $k = 2$ remaining candidates as our matches.

Illustrative Mapping Cases We present examples to illustrate both positive and negative mapping outcomes from our entity alignment module. A mapping is considered **positive** if one or more of the matched categories appear in the ground truth, and **negative** if all matches are semantically reasonable but absent from the GT labels.

Positive Mapping Cases

- *GT objects:* `person`, `tree`, `car`

- *VLM prediction:* man
- *SimCSE top-2 matches:*
  - gentleman (cos = 0.92) [not in GT]

  - person (cos = 0.89) **[in GT]**
- *VLM prediction:* woman
- *SimCSE top-2 matches:*
  - lady (cos = 0.90) [not in GT]

  - person (cos = 0.87) **[in GT]**
- *GT objects:* dog, grass
- *VLM prediction:* puppy
- *SimCSE top-2 matches:*
  - canine (cos = 0.82) [not in GT]

  - dog (cos = 0.79) **[in GT]**

Negative Mapping Cases

- *GT objects:* person, car, tree
- *VLM prediction:* skateboarder
- *SimCSE top-2 matches:*
  - skateboard (cos = 0.76) [not in GT]

  - rider (cos = 0.73) [not in GT]
- *GT objects:* tennis racket, person
- *VLM prediction:* tennis player
- *SimCSE top-2 matches:*
  - athlete (cos = 0.81) [not in GT]

  - player (cos = 0.78) [not in GT]

In Sec. B.4 we show how the negative mapping cases are handled by using Grounding DINO [26] as our object detection module.

B.3.1. Entity Mapping Ablation

To quantify the benefit of SimCSE's contrastive training, we ran an ablation comparing it against a standard Sentence-BERT (SBERT) [31] encoder—while keeping the same normalization and synonym steps across three datasets (PSG, OI, VG) and three VLMs: LLava Next, Qwen2-vl 7b *Qwen7)* and Qwen2-vl 72b(Qwen72). The grouped bar chart above shows recall for each model–method pairing. Overall, SimCSE (gold, crimson, sky-blue bars) yields up to a 5% recall boost over SBERT (orange, pink, teal bars) on the PSG and OI sets, particularly for Qwen7, highlighting its stronger discrimination of fine-grained object labels. On the more challenging VG data, both methods converge to lower recall, although SBERT slightly outperforms SimCSE for Qwen72 on PSG. These results suggest that contrastive supervision in SimCSE enhances generalization in complex scenes, while SBERT can sometimes better capture subtle category nuances in smaller models as shown in Fig. 3.

B.4. Entity Detection

We utilize Grounding-DINO [26] for zero-shot entity detection, specifically employing the *groundingdino_swinb_cogcoor* variant. We set the box_threshold to 35% and the text_threshold to 25%, following default values recommended by the authors. A single object name serve as a single text prompt for Grounding-DINO.

It is worth noting that the original Grounding-DINO paper highlights its capability to ground multiple objects in the text by separating their names with dots (e.g., 'person.cat.dog'). However, in our practical experience, while combining multiple objects in a single prompt speeds up entity detection, it compromises the quality of detected boxes. Grounding-DINO demonstrates superior performance when tasked with detecting a single object per text prompt. Therefore, we adopt a strategy of providing individual object names to maximize detection quality.

Object Filtering As discussed in Section B.3, the entity mapping stage may generate spurious or semantically irrelevant object labels. Here, we show how the pair refinement and filtering stages effectively remove such cases before the final triplet prediction. Figures 4 and 5 illustrate two examples where several incorrect or irrelevant mapped entities are successfully discarded.

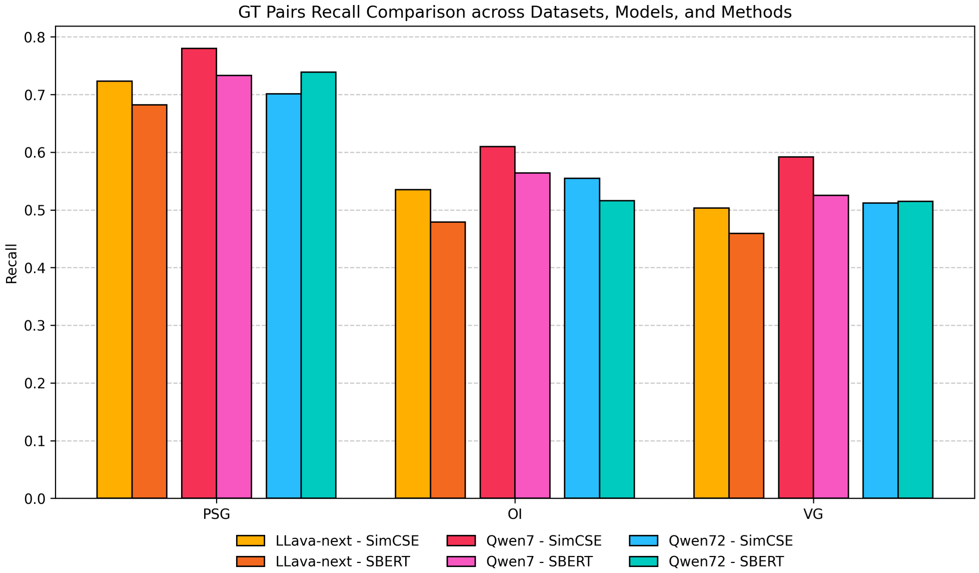


Figure 3. Recall comparison across datasets, models, and methods.

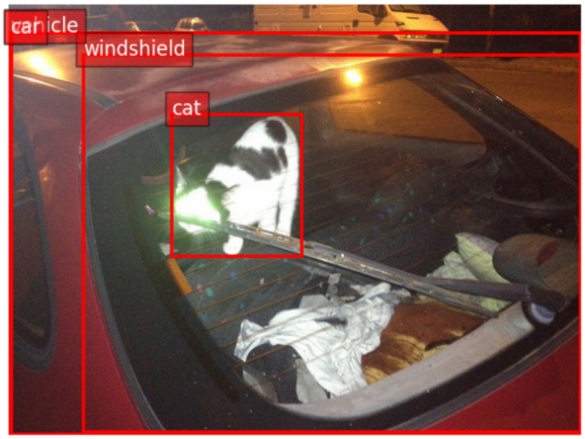


Figure 4. *Example 1* — Initial mapped entities: `[windshield, vehicle, light, building, car, street, cat, bag]`. Irrelevant objects such as `light`, `building`, `street`, and `bag` are successfully filtered out.

B.5. Pair Refinement

We present the prompt formulation and hyperparameter values used in the two stages of pair refinement in our framework.

B.5.1. Semantic Pair Refinement

To perform semantic filtering, we present the VLM with the full set of candidate entity pairs and ask it to rank them by their semantic relevance. Fig. 11 shows the exact prompt used in the example of Fig. 6, illustrating how the model refines the image's relationships.

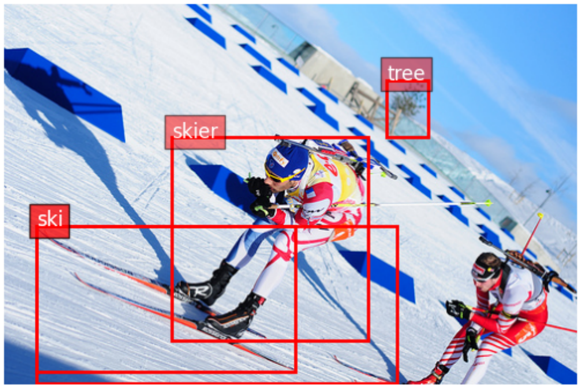


Figure 5. *Example 2* — Initial mapped entities: `[ski, light, tree, snow, skier, number, snow, roof]`. Irrelevant objects such as `light`, `number`, and `roof` are removed during filtering.

B.5.2. Geometric Pair Refinement

Following prior work [6], we adopt the same geometric distance formulation: $\lambda_1 \left(\frac{\mathbf{x_{ij}}}{y}\right) + \lambda_2\|\mathbf{d_i} - \mathbf{d_j}\|_2 < \tau$, where $\lambda_1 = 1.0$, $\lambda_2 = 1.5$, and $\tau = 0.5$. Unlike [6], which directly prunes pairs exceeding this threshold, we convert the distance into a soft compatibility score using a sigmoid function (Eq. 3).

We introduce an additional hyperparameter β, which controls the sharpness of this score. We use $\beta = 16$ for 7B models (LLaVA-next and Qwen2-vl 7B), and $\beta = 10$ for Qwen2-vl 72B. For the final fusion of semantic and geometric scores, we set the weighting factor $\alpha = 0.25$.

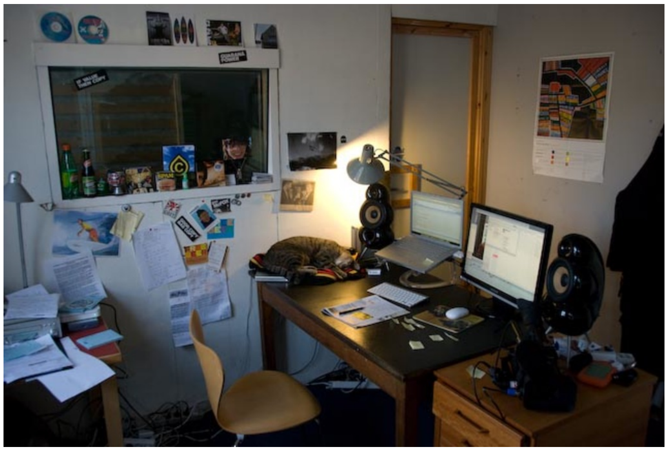


Figure 6. Example of semantic pair refinement. Given an image and a list of object pairs, the VLM is prompted to assign interaction likelihood scores, helping filter out semantically implausible relationships.

B.6. Scene Graph Generation

In the final scene-graph generation stage, we feed the VLM the semantically refined object pairs and ask it to infer their relationships. Because Qwen2-VL was instruction-tuned on bounding-box annotations—unlike the LLaVA models—the precise prompt templates differ: see `Relation Generation Prompt for LLaVA` and `Relation Generation Prompt for Qwen2-VL` for the exact prompts used for Fig. 7. The final list of correct and incorrect outputs is presented in `Sample VLM Outputs (Correct and Incorrect)`.

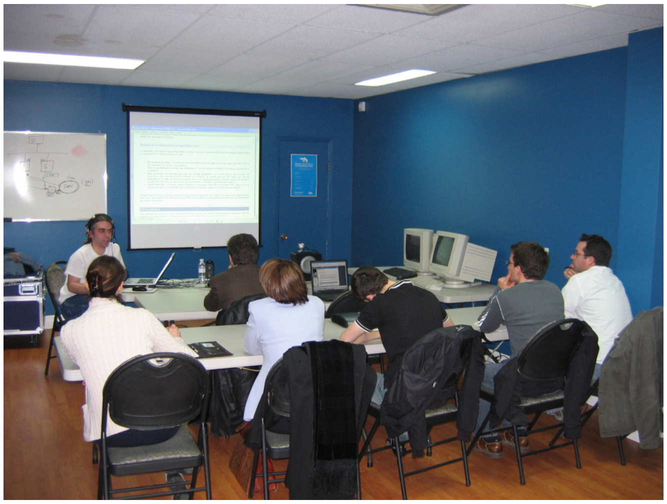


Figure 7. Final scene graph generation setup. Refined object pairs, along with their bounding box coordinates, are passed to the VLM to predict relationships.

Sample VLM Outputs (Correct and Incorrect)

Pair 1:
Sentence1: The woman is sitting on the chair. | Sentence2: The chair is being used by the woman.

Pair 2:
Sentence1: The woman is next to the chair. | Sentence2: The chair is beside the woman.

Pair 3:
Sentence1: The woman is located on the table. | Sentence2: The table is behind the woman.

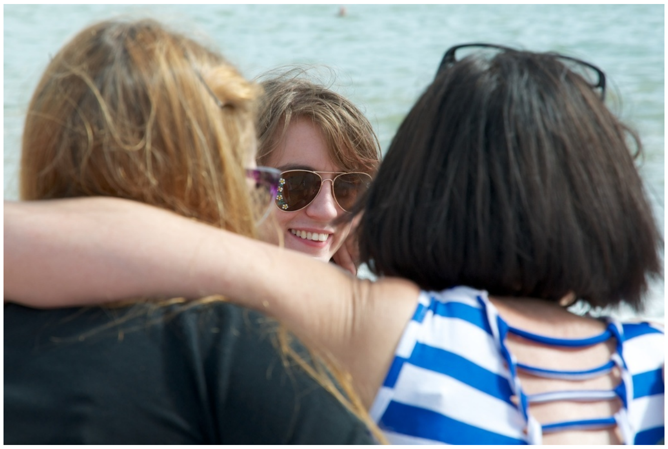
Pair 4:
Sentence1: The woman is resting her arm on the table. | Sentence2: The table is supporting the woman's arm.

Pair 5:
Sentence1: The chair is on top of the table. | Sentence2: The table is on the chair.

Pair 6:
Sentence1: The man is seated at the table. | Sentence2: The table is in front of the man.

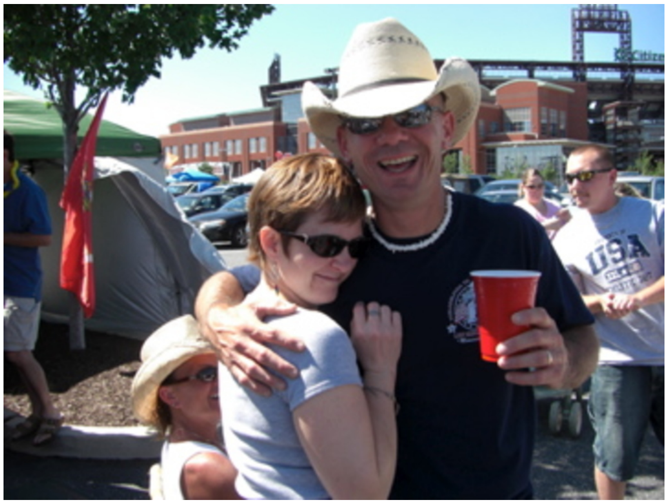
C. Qualitative Results for Pair Refinement

To better understand the impact of our pair refinement module, we visualize object pairs selected by each refinement strategy: semantic-only, depth-only, and the fused combination of both. For each image, we also list the ground-truth object pairs from the dataset. This comparison highlights how semantic and spatial cues contribute differently to filtering, and how their combination improves the selection of meaningful object pairs for relation prediction. Fig. 8, 9 and 10 show examples of these.



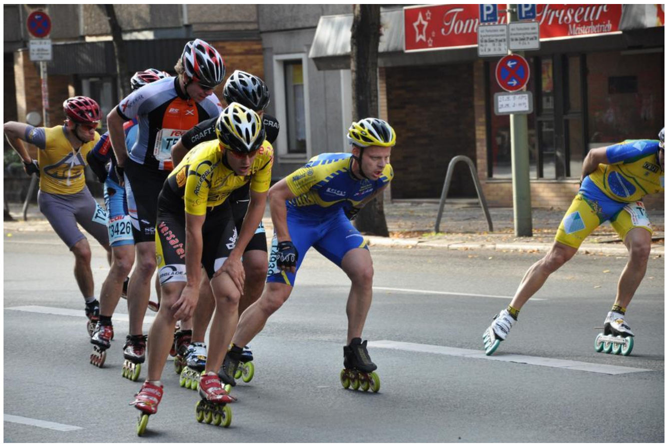
Semantic	Depth	Fused	GT Pairs
girl[385,79,587,399] glasses[660,57,936,146]	sunglasses[416,256,572,324] goggles[413,254,575,325]	girl[385,79,587,399] glasses[418,256,572,324]	girl[385,79,587,399] glasses[418,256,572,324]
sunglasses[416,256,572,324] girl[385,79,587,399]	glasses[418,256,572,325] girl[385,79,587,399]	sunglasses[416,256,572,324] girl[385,79,587,399]	sunglasses[416,256,572,324] girl[385,79,587,399]
girl[0,10,595,682] glasses[660,57,936,146]	woman[380,90,587,417] girl[385,79,587,399]	woman[380,90,587,417] girl[385,79,587,399]	woman[380,90,587,417] girl[385,79,587,399]

Figure 8. Example 1: Qualitative comparison of top object pairs from different methods versus the Ground Truth (GT) pairs for the given image (top). The table (bottom) details these pairs. Green indicates a correct pair, while red indicates an incorrect one.



Semantic Pairs	Depth Pairs	Fused Pairs	GT Pairs
girl[329,219,620,768] glasses[861,253,944,281]	woman[329,219,620,765] girl[329,219,620,768]	woman[329,219,620,765] girl[329,219,620,768]	woman[329,219,620,765] girl[329,219,620,768]
girl[329,219,620,768] sunglasses[520,153,662,204]	glasses[523,159,685,201] girl[329,219,620,768]	sunglasses[520,153,662,204] girl[329,219,620,768]	sunglasses[520,153,662,204] girl[329,219,620,768]
girl[329,219,620,768] sun hat[460,22,736,238]	glasses[423,355,566,395] man[295,19,924,768]	glasses[423,355,566,395] man[295,19,924,768]	glasses[423,355,566,395] man[295,19,924,768]

Figure 9. Example 2: Qualitative comparison of top object pairs from different methods versus the Ground Truth (GT) pairs for the given image (top). The table (bottom) details these pairs. Green indicates a correct pair, while red indicates an incorrect one.



Semantic Pairs	Depth Pairs	Fused Pairs	GT Pairs
bicycle helmet[90,138,156,21 man[751,189,1022,520]	man[197,150,420,633] roller skates[272,523,317,59	man[197,150,420,633] roller skates[272,523,317,59	man[197,150,420,633] roller skates[272,523,317,595]
bicycle helmet[328,149,409,2 man[259,98,429,563]	roller skates[361,524,390,58 man[331,174,576,589]	roller skates[361,524,390,58 man[331,174,576,589]	roller skates[361,524,390,585] man[331,174,576,589]
bicycle helmet[90,138,156,21 man[0,141,195,491]	roller skates[262,503,280,57 man[331,174,576,589]	man[131,98,262,530] bicycle helmet[90,138,156,21	man[131,98,262,530] bicycle helmet[90,138,156,218]

Figure 10. Example 3: Qualitative comparison of top object pairs from different methods versus the Ground Truth (GT) pairs for the given image (top). The table (bottom) details these pairs. Green indicates a correct pair, while red indicates an incorrect one.

Semantic Pair Scoring Prompt

```
You are a world-class vision-language analyst, highly specialized in understanding
spatial and functional relationships between objects in visual scenes.  Your role is to
evaluate how likely it is that specific object pairs are engaged in meaningful physical
interactions in the given image.
### Object Pair List:


Pair 1:  book and bookcase
Pair 2:  book and bottle
Pair 3:  book and cat
Pair 4:  book and chair
Pair 5:  book and chest of drawers
Pair 6:  book and computer monitor
Pair 7:  book and desk
Pair 8:  book and drawer
Pair 9:  book and lamp
Pair 10:  book and laptop
Pair 11:  book and mouse
Pair 12:  book and musical keyboard
Pair 13:  book and poster
Pair 14:  book and window
Pair 15:  bookcase and bottle
Pair 16:  bookcase and cat
Pair 17:  bookcase and chair
Pair 18:  bookcase and chest of drawers
Pair 19:  bookcase and computer monitor
Pair 20:  bookcase and desk
Pair 21:  bookcase and drawer
Pair 22:  bookcase and lamp
Pair 23:  bookcase and laptop
Pair 24:  bookcase and mouse
Pair 25:  bookcase and musical keyboard
Pair 26:  bookcase and poster
Pair 27:  bookcase and window
Pair 28:  bottle and cat
Pair 29:  bottle and chair
Pair 30:  bottle and chest of drawers
Pair 31:  bottle and computer monitor
Pair 32:  bottle and desk
Pair 33:  bottle and drawer
Pair 34:  bottle and lamp
Pair 35:  bottle and laptop
Pair 36:  bottle and mouse
Pair 37:  bottle and musical keyboard
Pair 38:  bottle and poster
Pair 39:  bottle and window
Pair 40:  cat and chair
Pair 41:  cat and chest of drawers
Pair 42:  cat and computer monitor
Pair 43:  cat and desk
Pair 44:  cat and drawer
Pair 45:  cat and lamp
Pair 46:  cat and laptop
Pair 47:  cat and mouse
Pair 48:  cat and musical keyboard
Pair 49:  cat and poster
Pair 50:  cat and window

### Task:
Carefully assess each object pair listed above and determine the likelihood that they
participate in a meaningful interaction within the scene.  Base your assessment on how
objects of those categories typically relate in physical or functional terms within
real-world images.
Provide a single integer confidence score from 1 to 5 for each pair, where:
- 1 = Very Unlikely
- 2 = Unlikely
- 3 = Uncertain
- 4 = Likely
- 5 = Very Likely

### Output Format:
- Do not include any object names, explanations, or extra text.
- Stop after the final pair.
- You must return exactly one line per pair listed above.
- Use the format:  Pair [index]:  [score]


### Begin:
```

Figure 11. The full text of the Semantic Pair Scoring Prompt.

Relation Generation Prompt for LLaVA

You are a vision-language expert. Given an image with pairs of objects along with their bounding box coordinates. The bounding box coordinates are defined by (X_top_left, Y_top_left, X_bottom_right, Y_bottom_right) and are normalized between 0 and 1.
Object Pair List
Pair 1: First object: ’woman’ [0.36, 0.51, 0.49, 0.87], Second object:’chair’ [0.37, 0.67, 0.57, 1.0]
Pair 2: First object: ’woman’ [0.36, 0.51, 0.49, 0.87], Second object:’chair’ [0.54, 0.64, 0.71, 1.0]
Pair 3: First object: ’woman’ [0.36, 0.51, 0.49, 0.87], Second object:’chair’ [0.7, 0.63, 0.9, 0.97]
Pair 4: First object: ’woman’ [0.36, 0.51, 0.49, 0.87], Second object:’chair’ [0.82, 0.61, 1.0, 0.93]
Pair 5: First object: ’woman’ [0.36, 0.51, 0.49, 0.87], Second object:’chair’ [0.14, 0.7, 0.32, 1.0]
Pair 6: First object: ’woman’ [0.36, 0.51, 0.49, 0.87], Second object:’table’ [0.07, 0.64, 0.54, 0.74]
Pair 7: First object: ’woman’ [0.36, 0.51, 0.49, 0.87], Second object:’table’ [0.51, 0.6, 0.77, 0.71]
Pair 8: First object: ’woman’ [0.36, 0.51, 0.49, 0.87], Second object:’chair’ [0.48, 0.57, 0.55, 0.63]
Pair 9: First object: ’woman’ [0.36, 0.51, 0.49, 0.87], Second object:’table’ [0.18, 0.55, 0.53, 0.65]
Pair 10: First object: ’chair’ [0.37, 0.67, 0.57, 1.0], Second object:’man’ [0.51, 0.52, 0.69, 0.91]
Pair 11: First object: ’chair’ [0.37, 0.67, 0.57, 1.0], Second object:’table’ [0.07, 0.64, 0.54, 0.74]
Pair 12: First object: ’chair’ [0.37, 0.67, 0.57, 1.0], Second object:’table’ [0.51, 0.6, 0.77, 0.71]
Pair 13: First object: ’man’ [0.08, 0.42, 0.19, 0.64], Second object:’chair’ [0.06, 0.51, 0.1, 0.75]
Pair 14: First object: ’man’ [0.08, 0.42, 0.19, 0.64], Second object:’chair’ [0.48, 0.57, 0.55, 0.63]
Pair 15: First object: ’man’ [0.08, 0.42, 0.19, 0.64], Second object:’table’ [0.18, 0.55, 0.53, 0.65]
Pair 16: First object: ’man’ [0.08, 0.42, 0.19, 0.64], Second object:’table’ [0.52, 0.52, 0.75, 0.62]
Pair 17: First object: ’chair’ [0.06, 0.51, 0.1, 0.75], Second object:’table’ [0.18, 0.55, 0.53, 0.65]
Pair 18: First object: ’man’ [0.51, 0.52, 0.69, 0.91], Second object:’chair’ [0.54, 0.64, 0.71, 1.0]
Pair 19: First object: ’man’ [0.51, 0.52, 0.69, 0.91], Second object:’chair’ [0.7, 0.63, 0.9, 0.97]
Pair 20: First object: ’man’ [0.51, 0.52, 0.69, 0.91], Second object:’chair’ [0.82, 0.61, 1.0, 0.93]
Pair 21: First object: ’man’ [0.51, 0.52, 0.69, 0.91], Second object:’table’ [0.07, 0.64, 0.54, 0.74]
Pair 22: First object: ’man’ [0.51, 0.52, 0.69, 0.91], Second object:’table’ [0.51, 0.6, 0.77, 0.71]
Pair 23: First object: ’man’ [0.51, 0.52, 0.69, 0.91], Second object:’chair’ [0.48, 0.57, 0.55, 0.63]
Pair 24: First object: ’man’ [0.51, 0.52, 0.69, 0.91], Second object:’table’ [0.18, 0.55, 0.53, 0.65]
Pair 25: First object: ’man’ [0.51, 0.52, 0.69, 0.91], Second object:’table’ [0.52, 0.52, 0.75, 0.62]
Pair 26: First object: ’chair’ [0.54, 0.64, 0.71, 1.0], Second object:’man’ [0.71, 0.47, 0.85, 0.92]

```
 Pair 27:  First object:  'chair' [0.54, 0.64, 0.71, 1.0], Second object:'man' [0.84,
 0.46, 0.96, 0.78]
 Pair 28:  First object:  'chair' [0.54, 0.64, 0.71, 1.0], Second object:'table' [0.07,
 0.64, 0.54, 0.74]
 Pair 29:  First object:  'chair' [0.54, 0.64, 0.71, 1.0], Second object:'table' [0.51,
 0.6, 0.77, 0.71]
 Pair 30:  First object:  'chair' [0.54, 0.64, 0.71, 1.0], Second object:'table' [0.18,
 0.55, 0.53, 0.65]
 Pair 31:  First object:  'man' [0.71, 0.47, 0.85, 0.92], Second object:'chair' [0.7,
 0.63, 0.9, 0.97]
 Pair 32:  First object:  'man' [0.71, 0.47, 0.85, 0.92], Second object:'chair' [0.82,
 0.61, 1.0, 0.93]
 Pair 33:  First object:  'man' [0.71, 0.47, 0.85, 0.92], Second object:'table' [0.51,
 0.6, 0.77, 0.71]
 Pair 34:  First object:  'man' [0.71, 0.47, 0.85, 0.92], Second object:'chair' [0.48,
 0.57, 0.55, 0.63]
 Pair 35:  First object:  'man' [0.71, 0.47, 0.85, 0.92], Second object:'table' [0.18,
 0.55, 0.53, 0.65]
 Pair 36:  First object:  'man' [0.71, 0.47, 0.85, 0.92], Second object:'table' [0.52,
 0.52, 0.75, 0.62]
 Pair 37:  First object:  'chair' [0.7, 0.63, 0.9, 0.97], Second object:'man' [0.84, 0.46,
 0.96, 0.78]
 Pair 38:  First object:  'chair' [0.7, 0.63, 0.9, 0.97], Second object:'table' [0.07,
 0.64, 0.54, 0.74]
 Pair 39:  First object:  'chair' [0.7, 0.63, 0.9, 0.97], Second object:'table' [0.51,
 0.6, 0.77, 0.71]
 Pair 40:  First object:  'chair' [0.7, 0.63, 0.9, 0.97], Second object:'table' [0.52,
 0.52, 0.75, 0.62]
 Pair 41:  First object:  'man' [0.84, 0.46, 0.96, 0.78], Second object:'chair' [0.82,
 0.61, 1.0, 0.93]
 Pair 42:  First object:  'man' [0.84, 0.46, 0.96, 0.78], Second object:'table' [0.51,
 0.6, 0.77, 0.71]
 Pair 43:  First object:  'man' [0.84, 0.46, 0.96, 0.78], Second object:'chair' [0.48,
 0.57, 0.55, 0.63]
 Pair 44:  First object:  'man' [0.84, 0.46, 0.96, 0.78], Second object:'table' [0.52,
 0.52, 0.75, 0.62]
 Pair 45:  First object:  'chair' [0.82, 0.61, 1.0, 0.93], Second object:'table' [0.51,
 0.6, 0.77, 0.71]
 Pair 46:  First object:  'chair' [0.14, 0.7, 0.32, 1.0], Second object:'table' [0.07,
 0.64, 0.54, 0.74]
 Pair 47:  First object:  'table' [0.07, 0.64, 0.54, 0.74], Second object:'chair' [0.48,
 0.57, 0.55, 0.63]
 Pair 48:  First object:  'table' [0.51, 0.6, 0.77, 0.71], Second object:'chair' [0.48,
 0.57, 0.55, 0.63]
 Pair 49:  First object:  'chair' [0.48, 0.57, 0.55, 0.63], Second object:'table' [0.18,
 0.55, 0.53, 0.65]
 Pair 50:  First object:  'chair' [0.48, 0.57, 0.55, 0.63], Second object:'table' [0.52,
 0.52, 0.75, 0.62]
 ### Output Format Instructions
- Write two sentences describing their spatial relationship.
- Sentence one describes how the first object is related to the second object.
- Sentence two describes how the second object is related to the first object.
- Use natural but concise relationships.
- Do not describe properties of a single object.
- Format your answer in the following manner:
 Pair [idx]:
 Sentence1:|Sentence2:
```

```
### Begin:
```

Relation Generation Prompt for Qwen2-VL

```
You are a vision-language expert. Given an image with pairs of objects along with their bounding box coordinates. The bounding box coordinates are defined by (X_top_left, Y_top_left, X_bottom_right, Y_bottom_right) and are scaled between 1 and 1000.
### Object Pair List
Pair 1: First object: ’woman’ [360, 510, 490, 870], Second object: ’chair’ [370, 670, 570, 1000]
Pair 2: First object: ’woman’ [360, 510, 490, 870], Second object: ’chair’ [540, 640, 710, 1000]
Pair 3: First object: ’woman’ [360, 510, 490, 870], Second object: ’chair’ [700, 630, 900, 970]
Pair 4: First object: ’woman’ [360, 510, 490, 870], Second object: ’chair’ [820, 610, 1000, 930]
Pair 5: First object: ’woman’ [360, 510, 490, 870], Second object: ’chair’ [140, 700, 320, 1000]
Pair 6: First object: ’woman’ [360, 510, 490, 870], Second object: ’table’ [70, 640, 540, 740]
Pair 7: First object: ’woman’ [360, 510, 490, 870], Second object: ’table’ [510, 600, 770, 710]
Pair 8: First object: ’woman’ [360, 510, 490, 870], Second object: ’chair’ [480, 570, 550, 630]
Pair 9: First object: ’woman’ [360, 510, 490, 870], Second object: ’table’ [180, 550, 530, 650]
Pair 10: First object: ’chair’ [370, 670, 570, 1000], Second object: ’man’ [510, 520, 690, 910]
Pair 11: First object: ’chair’ [370, 670, 570, 1000], Second object: ’table’ [70, 640, 540, 740]
Pair 12: First object: ’chair’ [370, 670, 570, 1000], Second object: ’table’ [510, 600, 770, 710]
Pair 13: First object: ’man’ [80, 420, 190, 640], Second object: ’chair’ [60, 510, 100, 750]
Pair 14: First object: ’man’ [80, 420, 190, 640], Second object: ’chair’ [480, 570, 550, 630]
Pair 15: First object: ’man’ [80, 420, 190, 640], Second object: ’table’ [180, 550, 530, 650]
Pair 16: First object: ’man’ [80, 420, 190, 640], Second object: ’table’ [520, 520, 750, 620]
Pair 17: First object: ’chair’ [60, 510, 100, 750], Second object: ’table’ [180, 550, 530, 650]
Pair 18: First object: ’man’ [510, 520, 690, 910], Second object: ’chair’ [540, 640, 710, 1000]
Pair 19: First object: ’man’ [510, 520, 690, 910], Second object: ’chair’ [700, 630, 900, 970]
Pair 20: First object: ’man’ [510, 520, 690, 910], Second object: ’chair’ [820, 610, 1000, 930]
Pair 21: First object: ’man’ [510, 520, 690, 910], Second object: ’table’ [70, 640, 540, 740] Pair 22: First object: ’man’ [510, 520, 690, 910], Second object: ’table’ [510, 600, 770, 710]
Pair 23: First object: ’man’ [510, 520, 690, 910], Second object: ’chair’ [480, 570, 550, 630]
Pair 24: First object: ’man’ [510, 520, 690, 910], Second object: ’table’ [180, 550, 530, 650]
Pair 25: First object: ’man’ [510, 520, 690, 910], Second object: ’table’ [520, 520, 750, 620]
Pair 26:  First object:  ’chair’ [540, 640, 710, 1000], Second object:  ’man’ [710, 470,
850, 920]
Pair 27:  First object:  ’chair’ [540, 640, 710, 1000], Second object:  ’man’ [840, 460,
960, 780]
Pair 28:  First object:  ’chair’ [540, 640, 710, 1000], Second object:  ’table’ [70, 640,
540, 740]
Pair 29:  First object:  ’chair’ [540, 640, 710, 1000], Second object:  ’table’ [510,
600, 770, 710]
Pair 30:  First object:  ’chair’ [540, 640, 710, 1000], Second object:  ’table’ [180,
550, 530, 650]
Pair 31:  First object:  ’man’ [710, 470, 850, 920], Second object:  ’chair’ [700, 630,
900, 970]
Pair 32:  First object:  ’man’ [710, 470, 850, 920], Second object:  ’chair’ [820, 610,
1000, 930]
Pair 33:  First object:  ’man’ [710, 470, 850, 920], Second object:  ’table’ [510, 600,
770, 710]
Pair 34:  First object:  ’man’ [710, 470, 850, 920], Second object:  ’chair’ [480, 570,
550, 630]
Pair 35:  First object:  ’man’ [710, 470, 850, 920], Second object:  ’table’ [180, 550,
530, 650]
Pair 36:  First object:  ’man’ [710, 470, 850, 920], Second object:  ’table’ [520, 520,
750, 620]
Pair 37:  First object:  ’chair’ [700, 630, 900, 970], Second object:  ’man’ [840, 460,
960, 780]
Pair 38:  First object:  ’chair’ [700, 630, 900, 970], Second object:  ’table’ [70, 640,
540, 740]
Pair 39:  First object:  ’chair’ [700, 630, 900, 970], Second object:  ’table’ [510, 600,
770, 710]
Pair 40:  First object:  ’chair’ [700, 630, 900, 970], Second object:  ’table’ [520, 520,
750, 620]
Pair 41:  First object:  ’man’ [840, 460, 960, 780], Second object:  ’chair’ [820, 610,
1000, 930]
Pair 42:  First object:  ’man’ [840, 460, 960, 780], Second object:  ’table’ [510, 600,
770, 710]
Pair 43:  First object:  ’man’ [840, 460, 960, 780], Second object:  ’chair’ [480, 570,
550, 630]
Pair 44:  First object:  ’man’ [840, 460, 960, 780], Second object:  ’table’ [520, 520,
750, 620]
Pair 45:  First object:  ’chair’ [820, 610, 1000, 930], Second object:  ’table’ [510,
600, 770, 710]
Pair 46:  First object:  ’chair’ [140, 700, 320, 1000], Second object:  ’table’ [70, 640,
540, 740]
Pair 47:  First object:  ’table’ [70, 640, 540, 740], Second object:  ’chair’ [480, 570,
550, 630]
Pair 48:  First object:  ’table’ [510, 600, 770, 710], Second object:  ’chair’ [480, 570,
550, 630]
Pair 49:  First object:  ’chair’ [480, 570, 550, 630], Second object:  ’table’ [180, 550,
530, 650]
Pair 50:  First object:  ’chair’ [480, 570, 550, 630], Second object:  ’table’ [520, 520,
750, 620]

### Output Instructions

- For each pair, write two short sentences:
- Sentence 1:  how the first object relates to the second.
- Sentence 2:  how the second object relates to the first.
- Focus on spatial or functional interactions.
- Use this format:  Pair [index]:  Sentence1:  | Sentence2:
 ### Begin:
```