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[1] Ismini Lourentzou[†2] Arka Daw[†3] Anuj Karpatne[†1]

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 multimodal 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 PSG dataset demonstrate the capacity of pretrained VLMs to perform relational understanding without task-level training.

1. Introduction

Scene Graph Generation (SGG) converts an image into a structured graph, with nodes as object entities and edges as semantic relationships. This representation supports structured reasoning over visual content and benefits tasks like image captioning, visual question answering, and referring expression generation (Johnson et al., 2015; Teney et al., 2017; Hudson & Manning, 2019; Yang et al., 2019). Accurate SGG demands understanding both visual appearance and contextual object interactions.

Traditional SGG methods are supervised and trained on datasets like Visual Genome, which provide dense triplet annotations. While effective, these models are limited by annotation bias, vocabulary constraints, and poor generalization to rare object-predicate classes (Zellers et al., 2018; Lu et al., 2016). To overcome these challenges, Open-Vocabulary SGG (OV-SGG) has emerged, targeting prediction of unseen objects (OVD) or relations (OVR) (Gu et al., 2021), and even settings where both objects or predicates may be novel at inference (Chen et al., 2024; Liu et al., 2025). However, such models still rely on fine-tuning or auxiliary training, limiting adaptability. With the rise of Vision-Language Models (VLMs) trained on large-scale image-text corpora (Bai et al., 2023; Deitke et al., 2024; Liu et al., 2024b; OpenAI, 2024; Team et al., 2024), a key question arises: *Can VLMs enable zero-shot SGG without task-specific training?* While VLMs demonstrate strong generalization and have been applied to reformulate SGG sub-tasks as image-text matching, most approaches still depend on dataset-specific components or costly pairwise inference, limiting open-world evaluation.

Despite growing interest, evaluating VLMs for SGG faces key challenges. *First*, there is no standardized baseline for open-world SGG. *Second*, prompting strategies for structured graph generation are underdeveloped. *Third*, extracting subject–predicate–object triplets from open-ended VLM outputs remains nontrivial, hindering compatibility with benchmarks such as Visual Genome (Krishna et al., 2017), Panoptic Scene Graph (Yang et al., 2022), and OpenImage (Kuznetsova et al., 2020).

To address this, we propose **Open World SGG (OwSGG)**: a zero-shot, model-agnostic framework using pretrained VLMs. Our approach combines multimodal prompting, embedding alignment, and a lightweight pair-refinement module to convert raw VLM outputs into structured graphs compatible with existing evaluation protocols—without any task-specific training. We evaluate LLaVa-next (Liu et al., 2024b) and Qwen2-VL (Wang et al., 2024) under closed, open-vocabulary, and fully open-world settings. While not state-of-the-art in closed settings, VLMs show potential in

[*]Equal contribution [†]Equal advising [1]Department of Computer Science, Virginia Tech [2]School of Information Sciences, University of Illinois Urbana-Champaign [3]Oak Ridge National Lab. Correspondence to: Amartya Dutta <amartya@vt.edu>, Anuj Karpatne <karpatne@vt.edu>.

ICML 2025 Workshop on Assessing World Models. Copyright 2025 by the author(s).

open-world cases. To support future research, we introduce a strong open-world evaluation baseline that isolates performance on novel object–relation pairs. Our results highlight the promise of pretrained VLMs for scalable, open-world scene understanding and underscore the need for new benchmarks.

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

We introduce our *framework for Open-world Scene Graph Generation (SGG) using Vision-Language Models (VLMs)*. We first provide background and notation, then *define a taxonomy of open-world SGG tasks*, highlighting their distinct challenges. Finally, we present our VLM-based approach and its suitability for open-world scenarios.

2.1. Background and Notations

Scene Graph Generation (SGG) aims to represent an image's visual content as a structured graph, or scene graph, defined as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. Here, $\mathcal{V} = \{\mathbf{v_i}\}_{i=1}^{N}$ denotes the set of nodes (objects), and $\mathcal{E} = \{\mathbf{e_{ij}}\} \subseteq \mathcal{V} \times \mathcal{V} \times \mathcal{R}$ is the set of directed edges, with $r_{ij} \in \mathcal{R}$, representing pairwise semantic relationships.

Each object $\mathbf{v_i} = (\mathbf{b_i}, o_i)$ includes a bounding box $\mathbf{b_i} \in \mathbb{R}^4$ and a class label $o_i \in \mathcal{O}$, where $\mathcal{O}$ is a predefined set of object categories. Each edge $\mathbf{e_{ij}} = (\mathbf{v_i}, \mathbf{v_j}, r_{ij})$ denotes a relation from $\mathbf{v_i}$ to $\mathbf{v_j}$, labeled by r_{ij}. We consider two problem formulations for SGG based on the available information.

Predicate Classification (PredCls). Given an image $I \in \mathbb{R}^{H \times W \times C}$ and the set of ground-truth objects $\mathcal{V}$, the task is to predict the relationships $\mathcal{E}$, i.e., identify object pairs $(\mathbf{v_i}, \mathbf{v_j})$ and assign predicate labels r_{ij}. As objects are provided, PredCls is a constrained variant of SGG.

Scene Graph Detection (SGDet). Given only the image I, the goal is to detect both objects $\mathcal{V}$ and their relationships $\mathcal{E}$. The output is a set of triplets $\mathcal{E} = \{(\mathbf{v_i}, \mathbf{v_j}, r_{ij}) \mid i \neq j,\ r_{ij} \in \mathcal{R}\}$, representing the scene's structured content. SGDet reflects a more general and realistic setting of the SGG problem.

2.2. Open-world Taxonomy

To benchmark SGG in open-world settings, we define a taxonomy of task setups based on the novelty of triplet components $(o_i, o_j, r_{ij}) \in \mathcal{E}_{\text{test}}$ with respect to the training set $\mathcal{E}_{\text{train}}$. These categories capture varying degrees of novelty in objects, predicates, or both.

Closed Vocabulary (CS). All triplets in $\mathcal{E}_{\text{test}}$ are seen during training, i.e., $(o_i, o_j, r_{ij}) \in \mathcal{E}_{\text{train}}$.

Zero-Shot (ZS). Full triplets are unseen, but components are known: $(o_i, o_j, r_{ij}) \notin \mathcal{E}_{\text{train}}$, while $o_i, o_j \in \mathcal{O}_{\text{train}}$ and $r_{ij} \in \mathcal{R}_{\text{train}}$.

Open-Vocabulary Relations (OVR). Predicate is novel, objects are seen: $o_i, o_j \in \mathcal{O}_{\text{train}}$, $r_{ij} \notin \mathcal{R}_{\text{train}}$.

Open-Vocabulary Detections (OVD). Objects are novel, predicate is known: $r_{ij} \in \mathcal{R}_{\text{train}}$, $o_i, o_j \notin \mathcal{O}_{\text{train}}$.

Open-Vocabulary Detections + Relations (OVD+R). Either the object or predicate is novel:

$$(o_i, o_j \notin \mathcal{O}_{\text{train}}) \vee (r_{ij} \notin \mathcal{R}_{\text{train}}).$$

Open World (OW). Both object and predicate are novel, the most challenging setup:

$$(o_i, o_j \notin \mathcal{O}_{\text{train}}) \wedge (r_{ij} \notin \mathcal{R}_{\text{train}}).$$

This taxonomy supports systematic evaluation of generalization to increasingly open and challenging scenarios in scene graph generation.

2.3. Proposed Framework for Using VLMs for Ow-SGG

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

2.3.1. 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. Our Entity Generation Prompts are shown in Appendix C.2.

2.3.2. Entity Mapping

We map VLM-predicted entities to known object categories, as their outputs may include paraphrases or hallucinations not aligned with dataset labels. To ensure semantic consistency, we use an embedding-based module that matches predicted entities to dataset categories via similarity scores computed with a contrastive encoder (SimCSE (Gao et al., 2021)). A softmax-based strategy ranks candidates, and up to k categories within a δ-neighborhood of the top match are retained per entity. This approach improves robustness to synonyms (e.g., "man" vs. "boy") and supports consistent downstream reasoning. Details are shown in the `Semantic Pair Scoring for Entity Mapping` box below. Additionally, we detail the steps and also compare our method against other approaches in Appendix C.3.

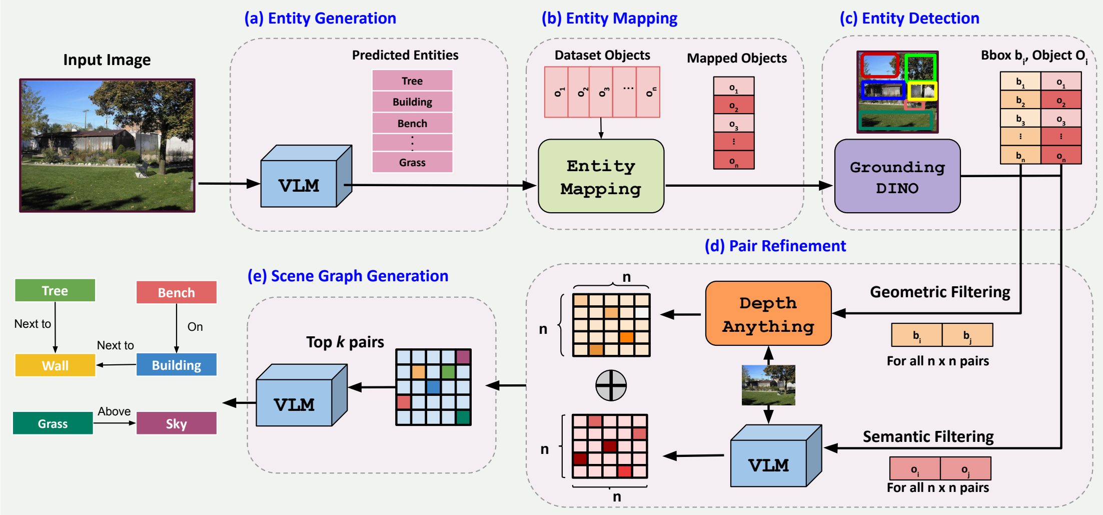

Figure 1. Overview of our proposed framework for open-world SGG using VLMs.

2.3.3. Entity Detection

Once the VLM-generated entities have been mapped to a list of candidate objects, Grounding DINO (Liu et al., 2024c) is used to 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. Appendix C.4 further highlights the strengths of our Detection module.

2.3.4. Pair Refinement

Given object proposals $\{(\mathbf{b_i}, o_i)\}_{i=1}^{N}$, we construct candidate pairs $(\mathbf{v_i}, \mathbf{v_j})$ for relation prediction. Exhaustively enumerating all $N(N-1)$ pairs is costly and noisy, while spatial filtering alone risks missing meaningful but non-adjacent interactions (e.g., `looking at`). We address this with a *pair refinement module* that combines semantic and geometric cues to retain only informative object pairs.

Semantic Pair Refinement. A VLM estimates semantic compatibility for each category pair (o_i, o_j), shared across instances:

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

producing a semantic matrix $\mathbf{P^S} \in \mathbb{R}^{N\times N}$.

Geometric Pair Refinement. To assess spatial plausibility, we compute each object's 2D center $\mathbf{c}_i^{2D} = (x_i, y_i)$ and its normalized median depth $\mathbf{d}_i \in [0, 1]$ from a monocular depth map (Yang et al., 2024). The pairwise distance between objects o_i and o_j combines normalized 2D and depth differences:

$$\mathbf{d_{ij}} = \lambda_1 \left(\frac{\|\mathbf{c}_i^{2D} - \mathbf{c}_j^{2D}\|_2}{\sqrt{H^2 + W^2}} \right) + \lambda_2 \|\mathbf{d}_i - \mathbf{d}_j\|_2, \tag{2}$$

where $\lambda_1, \lambda_2 > 0$ weight the 2D and depth contributions, and H, W are the image height and width.

To allow soft filtering, this distance is converted into a spatial compatibility score via a sigmoid:

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

where $\tau > 0$ is a distance threshold, and β controls the sharpness of the transition. The resulting matrix $\mathbf{P^G} \in \mathbb{R}^{N\times N}$ encodes the spatial plausibility of each object pair.

Fusion. We combine semantic and geometric scores:

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

where $\alpha \in [0, 1]$ balances modalities. Top-k pairs with highest $\mathbf{P^{\text{combined}}_{ij}}$ scores are retained for downstream relationship prediction. The effectiveness of our pair refinement strategy is further discussed in Appendix A. Additional implementation information and some qualitiatives examples are also highlighted in Appendix C.5 and D respectively.

2.3.5. 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. We show the prompts as well as some qualitative results in Appendix C.6.

3. Experimental Results

Evaluation Metrics. We evaluate relationship predictions R with confidence scores S using standard SGG metrics: *Recall@K (R@K)* and *mean Recall@K (mR@K)*.

Datasets. We evaluate on VG150 (Krishna et al., 2017), OIV6 (Kuznetsova et al., 2020), and PSG (Yang et al., 2022). VG150 contains 150 object and 50 relation classes; OIV6 includes 601 objects and 30 relations; PSG has 133 object and 56 relation classes.

Backbones and Baselines. We use LLaVa-next (Liu et al., 2024a), Qwen2-VL 7B, and Qwen2-VL 72B (Wang et al., 2024) as vision-language model (VLM) backbones to demonstrate our framework's model-agnostic nature. We compare against SOTA baselines including PGSG (Li et al., 2024), SGTR (Li et al., 2022), RAHP (Liu et al., 2025), and OvSGTR (Chen et al., 2024).

Semantic Pair Scoring for Entity Mapping

Let p be a predicted entity, $D = \{d_1, \dots, d_N\}$ the dataset categories, $h(\cdot)$ a SimCSE encoder, with temperature $\tau = 0.2$, threshold $\delta = 0.05$, and max $k = 2$ matches per entity.

1. Similarity scoring:

$$s_i = \cos\left(h(\text{"There is a } p\text{."}),\ h(\text{"There is a } d_i\text{."})\right)$$

2. Temperature-scaled softmax:

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

3. Near-maximum filter:

$$C = \left\{ i \;\middle|\; \max_j \hat{s}_j - \hat{s}_i < \delta \right\}$$

4. Retain top-k matches:

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

5. Merge mappings:

$$\mathcal{M} = \bigcup_{p \in \mathcal{P}} M(p)$$

$\mathcal{M}$ maps predicted entities to aligned dataset categories.

3.1. 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 1, 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.

Table 1. **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 (Liu et al., 2025)	–	3.75 / 5.12	–	–
	OvSGTR (Chen et al., 2024)	1.82 / 2.32	13.45 / 16.19	–	–
	OvSGTR+RAHP (Chen et al., 2024)	3.01 / 4.04	15.59 / 19.92	–	–
	PGSG (Li et al., 2024)	3.7 / 5.2	–	7.4 / 11.3	–
	SGTR (Li et al., 2022)	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 (Yu et al., 2023)	–	7.4 / 9.7	–	–
	PGSG (Li et al., 2024)	5.2 / 7.7	–	–	–
	SGTR+RAHP (Liu et al., 2025)	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

3.2. 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 2. While our models are not expected to outperform baselines trained explicitly on 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 (Jiang et al., 2025). 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.

3.3. 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 simulta-

Table 2. **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 (Xu et al., 2017)	11.7 / 14.8 / 16.1	– / 44.8 / 53.1	–
		MOTIFS(Zellers et al., 2018)	11.7 / 14.8 / 16.1	58.5 / 65.2 / 67.1	– / 10.9 /14.5
		VCTree+HIERCOM (Jiang et al., 2025)	17.6 / 26.3 / 31.8	55.9 / 69.8 / 75.8	– / 17.8 / 24.8
		CooK (Kim et al., 2024)	– / 33.7 / 35.8	– / 62.1 / 64.2	–
		CaCao (Yu et al., 2023)	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 (Teng & Wang, 2022)	– / 18.6 / 22.5	– / 23.7 / 27.3	– / 3.1 / 4.5
		SGTR (Li et al., 2022)	– / 12.0 / 15.2	–/ 24.6 / 28.4	– / 2.5 / 5.8
		PGSG (Li et al., 2024)	– / 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 (Li et al., 2022)	–	– / 59.9 / –	–
		ReIDN (Zhang et al., 2019)	–	– / 72.8 / –	–
		GPS-Net (Lin et al., 2020)	–	– / 74.7 / –	–
		HEIRCOM (Jiang et al., 2025)	–	– / 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 (Li et al., 2022)	– /38.6 / –	– / 59.1 / –	– / 19.4 / 31.6
		PGSG (Li et al., 2024)	– / 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 (Yang et al., 2022)	– / 20.3 / 21.5	– / 32.1 / 35.3	– / 3.1 / 6.4
		SGTR (Li et al., 2022)	– / 24.3 / 27.2	– / 33.1 / 36.3	– / 4.1 / 5.8
		PGSG (Li et al., 2024)	– / 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

neously. 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.

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 (Xu et al., 2017)	0.00 / 0.00	0.00 / 0.00	0.00 / 0.00†
MOTIFS (Zellers et al., 2018)	0.00 / 0.00	0.00 / 0.00	0.00 / 0.00†
VCTREE (Tang et al., 2019)	0.00 / 0.00	0.00 / 0.00	0.00 / 0.00†
TDE (Tang et al., 2020)	0.00 / 0.00	0.00 / 0.00	0.00 / 0.00†
VS3 (Zhang et al., 2023)	6.00 / 7.51	0.00 / 0.00	–
OvSGTR (Swin-B) (Chen et al., 2024)	17.58 / 21.72	14.56 / 18.20	5.97 / 10.06†
VS3+RAHP (Liu et al., 2025)	13.01 / 14.82	3.75 / 5.12	–
OvSGTR+RAHP (Swin-T) (Liu et al., 2025)	12.45 / 15.38	13.31 / 16.46	–
OwSGG (LLaVA-next)	2.9 / 3.7	2.33 / 3.04	1.92 / 2.56
OwSGG (Qwen7b)	1.16 / 1.73	1.15 / 1.67	0.86 / 1.18
OwSGG (Qwen72b)	2.48 / 3.44	2.19 / 3.06	1.61 / 2.41

4. Limitations and Conclusions

OwSGG depends on pre-trained components—Grounding-DINO for detection and SimCSE for similarity—which can introduce errors at various pipeline stages. Analyzing their impact is a key direction for future work. Prompting VLMs with all object pairs also raises scalability concerns due to context length limits, motivating the need for more efficient pair refinement. Despite these challenges, our results show that VLMs, when guided by prompts and lightweight modules, can predict scene graph relationships without task-specific training. This underscores the potential of zero-shot approaches for structured vision-language tasks and paves the way for more general and interpretable visual reasoning systems.

5. Acknowledgements

We thank the Advanced Research Computing Center at Virginia Tech for GPU resources and the Ohio Supercomputer Center for additional compute support. This work was performed by UT-Battelle, LLC under DOE contract DE-AC05-00OR22725; the U.S. Government retains a non-exclusive, irrevocable worldwide license to reproduce or publish this manuscript for government purposes, and results will be made publicly available per the DOE Public Access Plan (https://www.energy.gov/doe-public-access-plan). Support was also provided by NSF grant OAC-2118240 to the HDR Imageomics Institute.

References

Bai, J., Bai, S., Yang, S., Wang, S., Tan, S., Wang, P., Lin, J., Zhou, C., and Zhou, J. Qwen-vl: A frontier large vision-language model with versatile abilities. *arXiv preprint arXiv:2308.12966*, 2023.

Chen, Z., Wu, J., Lei, Z., Zhang, Z., and Chen, C. W. Expanding scene graph boundaries: fully open-vocabulary scene graph generation via visual-concept alignment and retention. In *European Conference on Computer Vision*, pp. 108–124. Springer, 2024.

Deitke, M., Clark, C., Lee, S., Tripathi, R., Yang, Y., Park, J. S., Salehi, M., Muennighoff, N., Lo, K., Soldaini, L., et al. Molmo and pixmo: Open weights and open data for state-of-the-art multimodal models. *arXiv preprint arXiv:2409.17146*, 2024.

Elskhawy, A., Li, M., Navab, N., and Busam, B. Prism-0: A predicate-rich scene graph generation framework for zero-shot open-vocabulary tasks. *arXiv preprint arXiv:2504.00844*, 2025.

Gao, T., Yao, X., and Chen, D. Simcse: Simple contrastive learning of sentence embeddings. *arXiv preprint arXiv:2104.08821*, 2021.

Gu, X., Lin, T.-Y., Kuo, W., and Cui, Y. Open-vocabulary object detection via vision and language knowledge distillation. *arXiv preprint arXiv:2104.13921*, 2021.

He, T., Gao, L., Song, J., and Li, Y.-F. Towards open-vocabulary scene graph generation with prompt-based finetuning. In *European Conference on Computer Vision*, pp. 56–73. Springer, 2022.

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

Jiang, B., Zhuang, Z., Shivakumar, S. S., and Taylor, C. J. Enhancing scene graph generation with hierarchical relationships and commonsense knowledge. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 8883–8894. IEEE, 2025.

Johnson, J., Krishna, R., Stark, L., Li, J., Shamma, D. A., Bernstein, M., and Fei-Fei, L. Image retrieval using scene graphs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3668–3678, 2015.

Kim, H., Kim, S., Ahn, D., Lee, J. T., and Ko, B. C. Scene graph generation strategy with co-occurrence knowledge and learnable term frequency. *arXiv preprint arXiv:2405.12648*, 2024.

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

Kuznetsova, A., Rom, H., Alldrin, N., Uijlings, J., Krasin, I., Pont-Tuset, J., Kamali, S., Popov, S., Malloci, M., Kolesnikov, A., 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.

Kwon, W., Li, Z., Zhuang, S., Sheng, Y., Zheng, L., Yu, C. H., Gonzalez, J. E., Zhang, H., and Stoica, I. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*, 2023.

Li, R., Zhang, S., and He, X. Sgtr: End-to-end scene graph generation with transformer. In *proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 19486–19496, 2022.

Li, R., Zhang, S., Lin, D., Chen, K., and He, X. 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*, pp. 28076–28086, 2024.

Lin, X., Ding, C., Zeng, J., and Tao, D. Gps-net: Graph property sensing network for scene graph generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3746–3753, 2020.

Liu, H., Li, C., Li, Y., Li, B., Zhang, Y., Shen, S., and Lee, Y. J. Llava-next: Improved reasoning, ocr, and world knowledge, January 2024a. URL https://llava-vl.github.io/blog/2024-01-30-llava-next/.

Liu, H., Li, C., Wu, Q., and Lee, Y. J. Visual instruction tuning. *Advances in neural information processing systems*, 36, 2024b.

Liu, S., Zeng, Z., Ren, T., Li, F., Zhang, H., Yang, J., Jiang, Q., Li, C., Yang, J., Su, H., et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In *European Conference on Computer Vision*, pp. 38–55. Springer, 2024c.

Liu, T., Li, R., Wang, C., and He, X. Relation-aware hierarchical prompt for open-vocabulary scene graph generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 5576–5584, 2025.

Lu, C., Krishna, R., Bernstein, M., and Fei-Fei, L. 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*, pp. 852–869. Springer, 2016.

OpenAI. "hello gpt-4", 2024.

Reimers, N. and Gurevych, I. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*, 2019.

Tang, K., Zhang, H., Wu, B., Luo, W., and Liu, W. Learning to compose dynamic tree structures for visual contexts. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 6619–6628, 2019.

Tang, K., Niu, Y., Huang, J., Shi, J., and Zhang, H. Unbiased scene graph generation from biased training. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 3716–3725, 2020.

Team, G., Georgiev, P., Lei, V. I., Burnell, R., Bai, L., Gulati, A., Tanzer, G., Vincent, D., Pan, Z., Wang, S., et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*, 2024.

Teney, D., Liu, L., and van Den Hengel, A. Graph-structured representations for visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–9, 2017.

Teng, Y. and Wang, L. Structured sparse r-cnn for direct scene graph generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19437–19446, 2022.

Wang, P., Bai, S., Tan, S., Wang, S., Fan, Z., Bai, J., Chen, K., Liu, X., Wang, J., Ge, W., et al. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024.

Xu, D., Zhu, Y., Choy, C. B., and Fei-Fei, L. Scene graph generation by iterative message passing. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5410–5419, 2017.

Yang, J., Ang, Y. Z., Guo, Z., Zhou, K., Zhang, W., and Liu, Z. Panoptic scene graph generation. In *European Conference on Computer Vision*, pp. 178–196. Springer, 2022.

Yang, L., Kang, B., Huang, Z., Zhao, Z., Xu, X., Feng, J., and Zhao, H. Depth anything v2. *Advances in Neural Information Processing Systems*, 37:21875–21911, 2024.

Yang, S., Li, G., and Yu, Y. Cross-modal relationship inference for grounding referring expressions. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4145–4154, 2019.

Yu, Q., Li, J., Wu, Y., Tang, S., Ji, W., and Zhuang, Y. 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*, pp. 21560–21571, 2023.

Zellers, R., Yatskar, M., Thomson, S., and Choi, Y. Neural motifs: Scene graph parsing with global context. arXiv preprint arXiv:1711.06640, 2018.

Zhang, J., Shih, K. J., Elgammal, A., Tao, A., and Catanzaro, B. Graphical contrastive losses for scene graph parsing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11535–11543, 2019.

Zhang, Y., Pan, Y., Yao, T., Huang, R., Mei, T., and Chen, C.-W. Learning to generate language-supervised and open-vocabulary scene graph using pre-trained visual-semantic space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2915–2924, 2023.

A. Effectiveness of Pair Refinement

Fig. 2 (a) illustrates how varying the hyperparameter α, with a fixed `top_k` of 25, influences model performance. 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 pairs are filtered out. As expected, increasing `top_k` generally improves recall by increasing the likelihood of retrieving Ground Truth object pairs. 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 for the OwSGG (Qwen2-72B) model. We observe that while depth-only refinement performs well in a close-vocabulary setting, combining semantic and depth-based filtering yields consistently better performance as the evaluation setting becomes more open and data-limited.

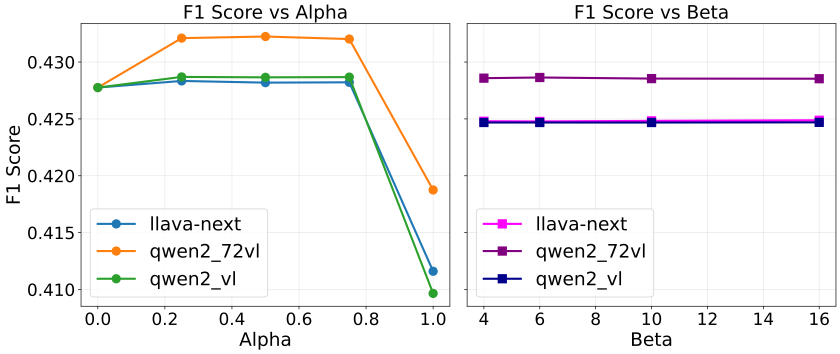

Figure 2. Ablation Study: F1 scores across different (a) α and (b) β values for the Qwen-72B model.

Table 4. Effect of depth and semantic filtering on PSG (PredCls, Qwen72B). Bold = best, underline = 2$^{\text{nd}}$ best.

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

B. 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) (Krishna et al., 2017), Open Images V6 (OIV6) (Kuznetsova et al., 2020), and Panoptic Scene Graph (PSG) (Yang et al., 2022), using the standard splits from prior work (Xu et al., 2017; Zellers et al., 2018; Yang et al., 2022). Since our method requires no training, we evaluate only on the test data. For VG (Krishna et al., 2017), we follow the cleaning protocol of (Xu et al., 2017; Zellers et al., 2018), removing images with insufficient annotations. This yields 26,446 test images (from 32,422), covering 150 object and 50 relation classes. For OIV6 (Kuznetsova et al., 2020), we use the test split with 5,322 images, 601 object classes, and 30 relations. For PSG (Yang et al., 2022), 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 (He et al., 2022).
- **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 (Xu et al., 2017; Zellers et al., 2018). For PSG, we follow the official code and splits distributed at[5].

C. Implementation Details

C.1. Vision Language Models

All VLMs used are instruction-tuned to interpret structured prompts better. For inference, we leverage the vLLM framework (Kwon et al., 2023), 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.

C.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 is to enable nucleus sampling with a large cutoff to avoid premature truncation of less frequent but relevant entities

5. `presence_penalty=0.4`: To penalize repetitions to encourage novel mentions without being too aggressive

6. `repetition_penalty=1.1`: To mildly discourage duplicate tokens during generation

Prompt examples. Datasets like (Krishna et al., 2017) have a lot of trivial objects such as *"hair"*, while (Kuznetsova et al., 2020) has generic as well as specific object names such as *"girl"* and *"woman"* as compared to cleaner object names in (Yang et al., 2022). Therefore, in order to instruct the VLMs to generate entities that are present in the target datasets, 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.
```

C.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 (Gao et al., 2021) 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) [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. C.4 we show how the negative mapping cases are handled by using Grounding DINO (Liu et al., 2024c) as our object detection module.

C.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) (Reimers & Gurevych, 2019) 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.

C.4. Entity Detection

We utilize Grounding-DINO (Liu et al., 2024c) 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 C.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.

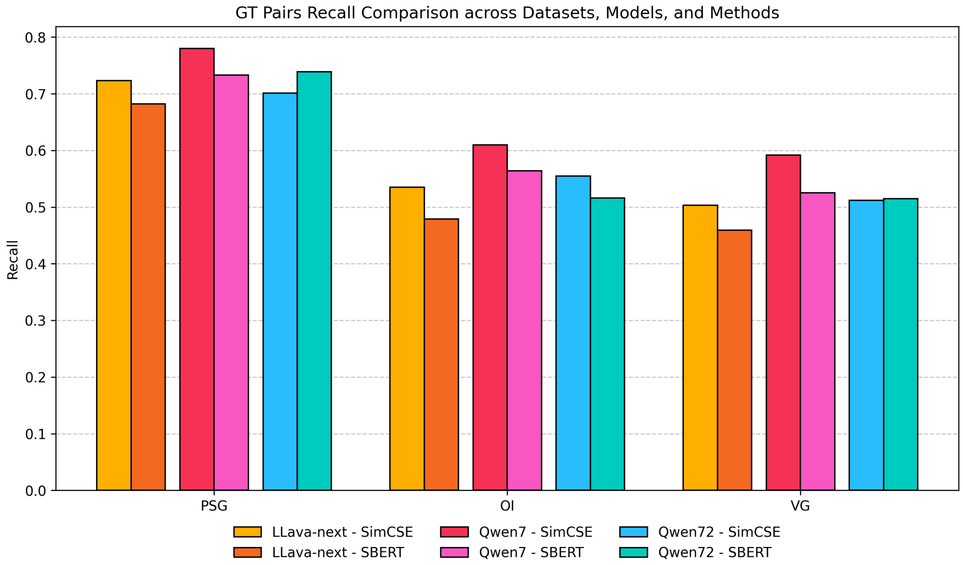

Figure 3. Recall comparison across datasets, models, and methods.

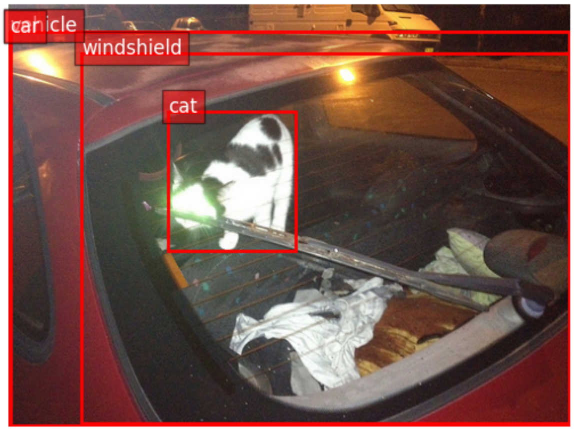

Figure 4. Example 1 — Initial mapped entities: `[windshield, vehicle, light, building, car, street, cat, bag]`. Irrelevant objects such as `light`, `building`, `street`, and `bag` are successfully filtered out.

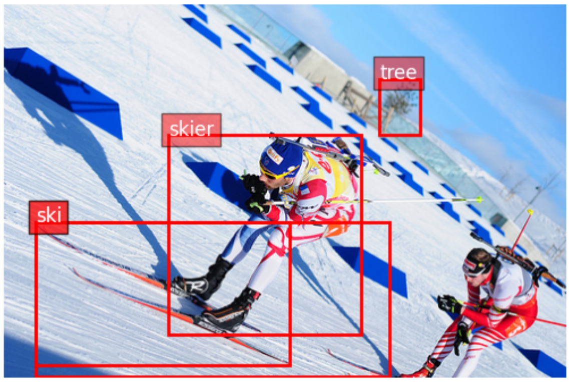

Figure 5. Example 2 — Initial mapped entities: `[ski, light, tree, skier, number, snow, roof]`. Irrelevant objects such as `light`, `number`, and `roof` are removed during filtering.

C.5. Pair Refinement

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

C.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. `Semantic Pair Scoring Prompt` box shows the exact prompt used for Fig. 6, illustrating how the model refines the image's relationships.

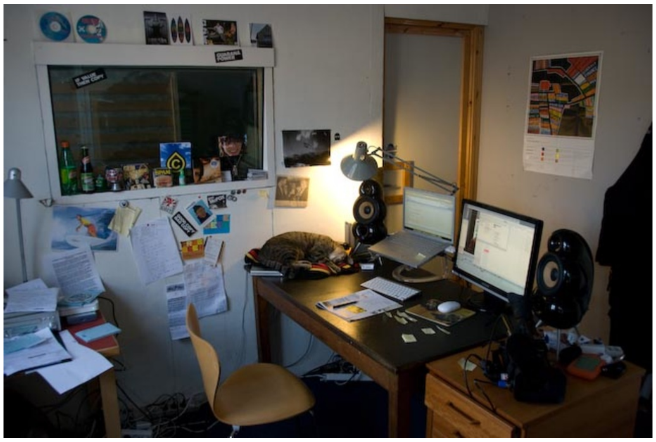

Figure 6. Example of semantic pair refinement. Given an image and a list of object pairs, the VLM is prompted to assign interaction likelihood scores, helping filter out semantically implausible relationships.

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:
```

C.5.2. Geometric Pair Refinement

Following prior work (Elskhawy et al., 2025), 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 (Elskhawy et al., 2025), 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 $\alpha = 0.25$.

C.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).

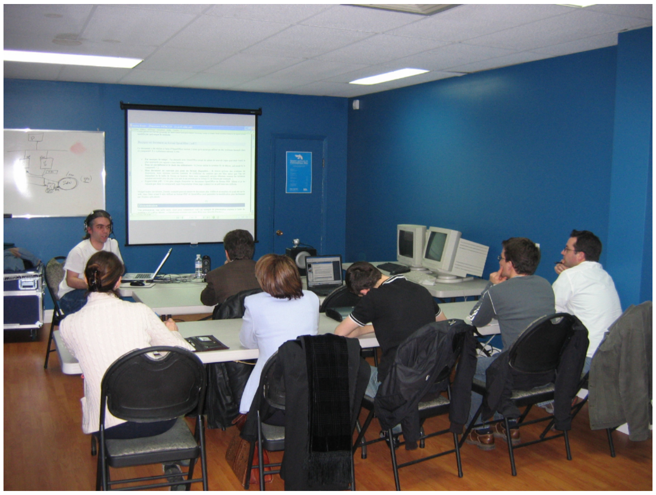

Figure 7. Final scene graph generation setup. Refined object pairs, along with their bounding box coordinates, are passed to the VLM to predict relationships.

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.

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:
```

D. 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. Tab. 5, 6, and 7 show the pairs detected from the images 8, 9 and 10 respectively.

Table 5. Qualitative comparison of top object pairs per method along with the Ground Truth object pairs for Fig. 8. Green = correct pair, Red = incorrect.

Semantic Pairs	**Depth Pairs**	**Fused Pairs**	**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]
girl[385,79,587,399] sunglasses[416,256,572,324]	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: Pair Refinement

Figure 9. Example 2: Pair Refinement

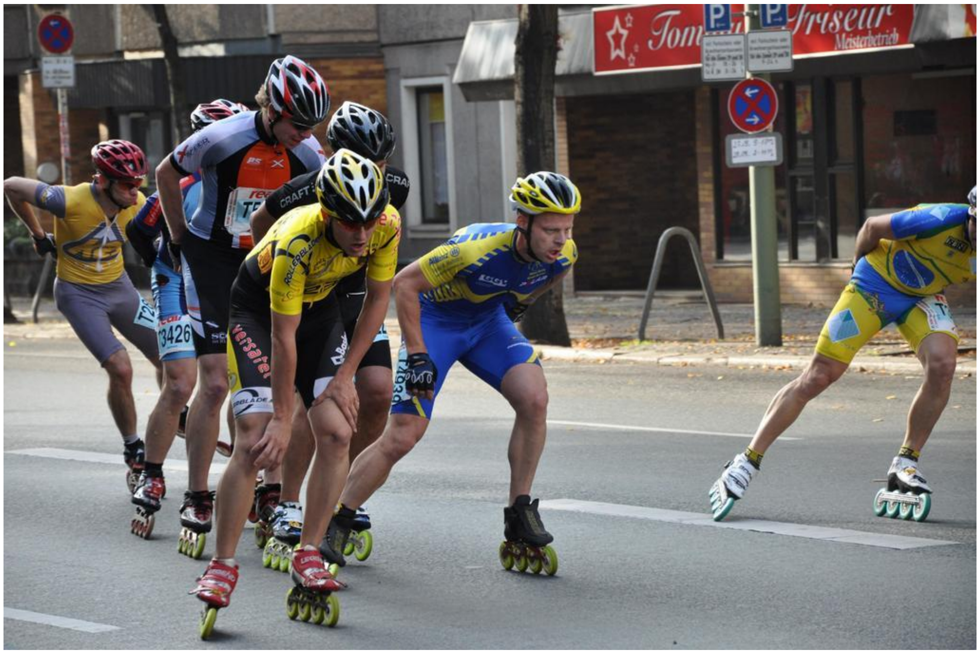

Figure 10. Example 3: Pair Refinement

Table 6. Qualitative comparison of top object pairs per method along with the Ground Truth object pairs for Fig. 9. Green = correct pair, Red = incorrect.

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]

Table 7. Qualitative comparison of top object pairs per method along with the Ground Truth object pairs for Fig. 10. Green = correct pair, Red = incorrect.

Semantic Pairs	Depth Pairs	Fused Pairs	GT Pairs
🟥 bicycle helmet[90,138,156,218 man[751,189,1022,520]	🟩 man[197,150,420,633] roller skates[272,523,317,595	🟩 man[197,150,420,633] roller skates[272,523,317,595	🟩 man[197,150,420,633] roller skates[272,523,317,595]
🟩 bicycle helmet[328,149,409,26 man[259,98,429,563]	🟩 roller skates[361,524,390,585 man[331,174,576,589]	🟩 roller skates[361,524,390,585 man[331,174,576,589]	🟩 roller skates[361,524,390,585] man[331,174,576,589]
🟥 bicycle helmet[90,138,156,218 man[0,141,195,491]	🟥 roller skates[262,503,280,571 man[331,174,576,589]	🟩 man[131,98,262,530] bicycle helmet[90,138,156,218	🟩 man[131,98,262,530] bicycle helmet[90,138,156,218]