SGG-R^3: From Next-Token Prediction to End-to-End Unbiased Scene Graph Generation

Jiaye Feng[1], Qixiang Yin[2], Yuankun Liu[1], Tong Mo[1*], Weiping Li[1]
[1]School of Software and Microelectronics, Peking University, Beijing, China
[2]Zhongguancun Academy, Beijing, China
{jyfeng25, liuyk}@stu.pku.edu.cn,
buptyqx@bupt.edu.cn, {motong, wpli}@ss.pku.edu.cn
[*]Corresponding author

Abstract

Scene Graph Generation (SGG) structures visual scenes as graphs of objects and their relations. While Multimodal Large Language Models (MLLMs) have advanced end-to-end SGG, current methods are hindered by both a lack of task-specific structured reasoning and the challenges of sparse, long-tailed relation distributions, resulting in incomplete scene graphs characterized by low recall and biased predictions. To address these issues, we introduce SGG-R^3, a structured reasoning framework that integrates task-specific Chain-of-Thought (CoT)-guided Supervised Fine-Tuning (SFT) and Reinforcement Learning (RL) with Group Sequence Policy Optimization (GSPO), designed to engage in three sequential stages to achieve end-to-end unbiased scene graph generation. During the SFT phase, we propose a relation augmentation strategy by leveraging an MLLM and refined via embedding similarity filtering to alleviate relation sparsity. Subsequently, a stage-aligned reward scheme optimizes the procedural reasoning during RL. Specifically, we propose a novel dual-granularity reward which integrates fine-grained and coarse-grained relation rewards, simultaneously mitigating the long-tail issue via frequency-based adaptive weighting of predicates and improving relation coverage through semantic clustering. Experiments on two benchmarks show that SGG-R^3 achieves superior performance compared to existing methods, demonstrating the effectiveness and generalization of the framework.

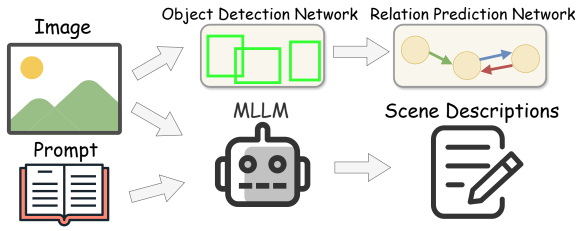


Figure 1: Pipeline comparison of Scene Graph Generation between the two-stage classification framework and the end-to-end generative MLLM method.

1 Introduction

Scene Graph Generation (SGG) aims to parse images into structured semantic graphs that represent objects and their interrelationships, providing a crucial bridge between vision and language that facilitates downstream applications including visual question answering (Kenfack et al., 2020), image retrieval (Schroeder and Tripathi, 2020), and embodied navigation (Zhu et al., 2021; Gadre et al., 2022; Huang et al., 2025a). Traditional SGG methods (Xu et al., 2017; Zellers et al., 2018; Tang et al., 2019) predominantly follow a two-stage paradigm, decoupling the task into object detection followed by relation prediction, with efforts concentrated on refining the latter module. To improve efficiency and mitigate error propagation, end-to-end SGG approaches (Lin et al., 2020; Li et al., 2022; Cong et al., 2023; Hayder and He, 2024) have emerged, which are often based on Transformer (Vaswani et al., 2017) architectures that unify object extraction and relation analysis. However, such methods tend to produce predictions that are biased toward high-frequency relations and have limited generalization to unseen concepts due to overfitting on annotated data.

Recently, Multimodal Large Language Models (MLLMs) (Liu et al., 2023; Wang et al., 2024b; Bai et al., 2025) have advanced rapidly, demonstrating considerable capabilities in object grounding and relational reasoning. Unlike conventional classification-based SGG methods that treat relation prediction as discrete labeling over fixed object proposals, MLLMs facilitate the end-to-end generation of scene descriptions in an open-ended manner. As illustrated in Figure 1, guided by the task prompt, the model directly generates a tex-

Findings of the Association for Computational Linguistics: ACL 2026, pages 19811–19830

tual representation of the scene graph, eliminating the need for the two-stage generation process required by conventional methods. However, applying MLLM to end-to-end SGG presents two critical challenges. First, the absence of SGG-specific structured reasoning pipeline compels model to navigate an excessively vast search space. For instance, using generic directives like "think step-by-step in <think></think> tags" without granular procedural guidance makes it difficult to form a valid Chain-of-Thought (CoT), leading to hallucinations and low recall. Second, the inherent sparsity and long-tailed distribution of relation data (detailed analysis in Appendix B.1) drive fine-tuned models to predict incomplete graphs with biased relations. While existing research (Chen et al., 2025; Li et al., 2025b) utilizes Supervised Fine-Tuning (SFT) and Reinforcement Learning (RL) to optimize output, these efforts often struggle with precise object grounding and unbiased relational modeling due to the absence of a specialized structured reasoning process.

Notably, SGG fundamentally differs from typical visual question answering, as it requires dense and sequential prediction of both objects and their relations, thus necessitating a tailored CoT process. To this end, we introduce SGG-R^3, a structured reasoning framework that integrates CoT-guided SFT and RL paradigm. To alleviate relation sparsity and the long-tailed distribution issue, we first generate relation-augmented data by leveraging Qwen2.5-VL-32B (Bai et al., 2025). This dataset is then filtered by computing the cosine similarity between Sentence-BERT (Reimers and Gurevych, 2019) embeddings of the generated relations and the original dataset relations, thereby maintaining semantic alignment and yielding higher-quality samples for SFT. To establish foundational reasoning capabilities, the model is supervised fine-tuned on relation-augmented data formatted as CoT, explicitly engaging in three sequential steps of object category detection, object instance grounding, and multi-type relation extraction. The RL phase further enhances the model's capabilities in visual grounding and relational reasoning. We introduce a novel dual-granularity reward that integrates fine-grained and coarse-grained relation rewards, which adaptively weights predicates based on their frequency to alleviate the long-tail issue, while also enhancing overall relation coverage by using DBSCAN-based (Ester et al., 1996) semantic clustering to group and match similar relations. Our main contributions are summarized as follows:

- We propose a structured reasoning framework for end-to-end unbiased scene graph generation, which enhances controllability and logical coherence by decomposing the generation into three sequential stages.
- We introduce a relation augmentation method for SFT and a novel dual-granularity reward for RL, jointly mitigating the long-tailed distribution issue and promoting the generation of unbiased scene graphs.
- Extensive experiments on two benchmarks show that SGG-R^3 outperforms previous methods across multiple metrics, demonstrating its superior capability in visual grounding and relation identification.

2 Related Works

2.1 VLM for Scene Graph Generation

Vision-Language Models (VLMs) provide a promising approach for SGG by leveraging cross-modal alignment. Current works (Zhang et al., 2023; Li et al., 2024b; Chen et al., 2024) typically retrieve objects and relations through global or regional feature matching for open-vocabulary SGG. (Yu et al., 2023) tackles the long-tailed distribution by mining fine-grained predicates via a VLM and an adaptive semantic clustering loss. Meanwhile, MLLMs are fine-tuned to generate scene graphs in an end-to-end manner via SFT and RL. (Chen et al., 2025) pioneers the application of RL to end-to-end SGG by designing a graph-constrained reward to optimize structured outputs, while (Li et al., 2025b) incorporates a cognitive CoT mechanism to improve relation recognition. However, both approaches still fail to address the sparsity and long-tail issues of relations.

2.2 Chain of Thought in Visual Reasoning

The success of CoT in Large Language Models (Wei et al., 2022) has motivated its adaptation to visual reasoning tasks. By breaking down complex problems into sequences of intermediate reasoning steps, CoT guides models to produce incremental solutions rather than direct answers. Several related methods (Shao et al., 2024a; Huang et al., 2025b; Liu et al., 2025; Chu

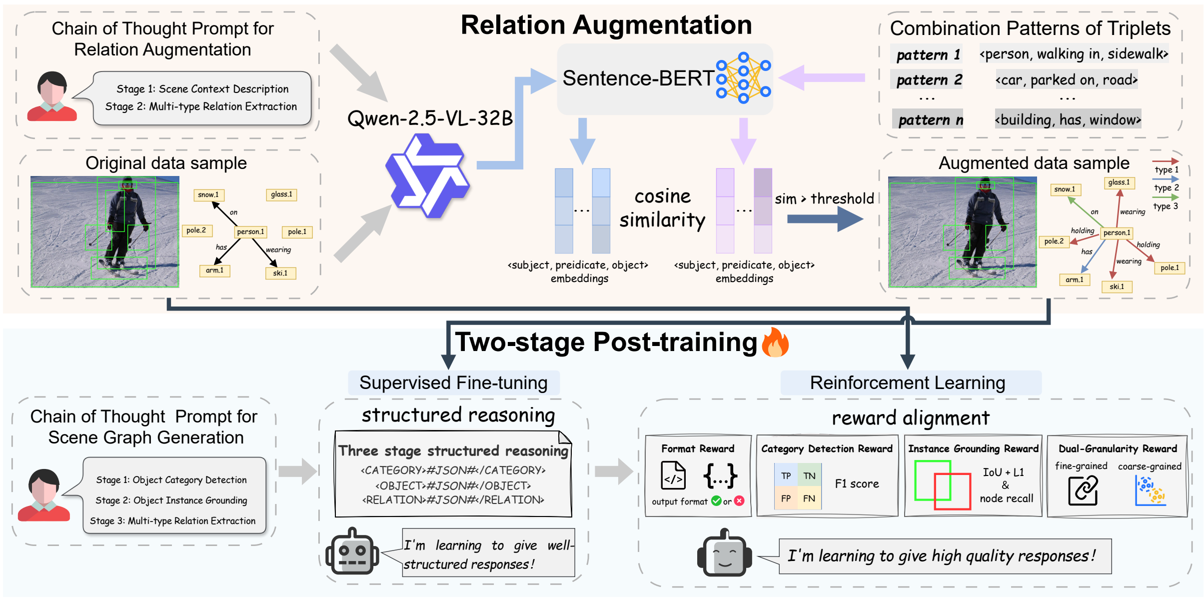


Figure 2: Overview of the SGG-R^3 framework. The "R^3" denotes three key contributions: **R**elation augmentation, structured **R**easoning, and **R**eward alignment. First, candidate relations generated by the Qwen2.5-VL-32B with CoT prompt are filtered via Sentence-BERT embedding similarity against the original data. The model is then supervised fine-tuned on the CoT-formatted augmented data with CoT prompt, followed by reward-driven reinforcement learning aligned with the original dataset.

et al., 2025; Yuan et al., 2026) integrate explicit visual evidence during generation to bridge the gap between visual perception and logical reasoning. (Xu et al., 2025) adopt structured sequential stages to achieve marked improvements on reasoning-intensive tasks. Unlike general visual reasoning, SGG's dense prediction nature necessitates a tailored structured reasoning process.

3 Methodology

This section begins with a formal definition of the SGG task, followed by the introduction of our three-stage progressive reasoning process. Furthermore, we present a type-aware relation augmentation strategy designed for the SFT phase. Finally, we detail the stage-aligned reward functions and the RL algorithm integrated in our framework. The overall architecture is illustrated in Figure 2.

3.1 Problem Formulation

The goal of SGG is to convert an input image I into a structured graph representation $G = (\mathcal{V}, \mathcal{E})$. Here, $\mathcal{V}$ denotes the set of object nodes, where each node $v \in \mathcal{V}$ corresponds to an object instance represented by a tuple (b, c). The vector $b \in \mathbb{R}^4$ describes the object's bounding box coordinates, and c is its category label. The set $\mathcal{E}$ comprises relation triplets of the form (v_s, p, v_o), where $v_s, v_o \in \mathcal{V}$ represent the subject and object of the triplet, respectively, and p is the predicate describing their interaction. In a closed-set setting, all object categories c and predicate categories p are drawn from predefined sets $\mathcal{C}$ and $\mathcal{P}$, respectively.

3.2 Three-stage Structured Reasoning

The model acquires structured reasoning capabilities via visual instruction tuning in a prompt-response format. Guided by the theory of prompt space complexity (Zhang et al., 2025), we design a CoT prompt that enforces explicit task constraints and a three-stage cognitive process (see Appendix A.4). During SFT, the response is structured to systematize the unordered SGG process into a rigorous, sequential three-stage pipeline (details in Appendix A.6). The structured reasoning stages are defined as follows.

Stage 1: Object Category Detection. Direct bounding box detection for predefined object categories is prone to error propagation, resulting in imprecise localization, hallucination artifacts, and suppressed recall. To mitigate this, our model first identifies only the categories present in the image, effectively narrowing the search space to relevant object categories.

Stage 2: Object Instance Grounding. Based on the category set from Stage 1, the model performs instance grounding by sequentially processing each identified category. Specifically, it detects and localizes all object instances belonging to the current category before proceeding to the next one.

Stage 3: Multi-type Relation Extraction. This stage focuses on extracting *(subject-predicate-object)* relation triplets. To capture the semantic and visual diversity of predicates, the model explores latent inter-object associations across three predefined predicate types. For example, in VG150 (Xu et al., 2017) we consider spatial, possessive, and interactive relations, while in PSG (Yang et al., 2022) we distinguish spatial, static-interactive, and dynamic-interactive relations, with details in Appendix A.3. The output order of relations within each type adheres to the subject instance sequence in Stage 2.

To maintain structural integrity during reasoning, SGG-R[3] explicitly marks each stage with dedicated tags: `<CATEGORY></CATEGORY>`, `<OBJECT></OBJECT>`, and `<RELATION></RELATION>`. These tags correspond to the start and end of each stage respectively, with outputs standardized in JSON format. All stages are completed in a single inference pass.

3.3 Type-aware Relation Augmentation

SFT enables the model to learn structured outputs and the target data distribution, yet its generalization is often constrained by the sparsity and long-tailed nature of relation annotations. To address this, we propose a type-aware relation augmentation strategy based on Qwen2.5-VL-32B (Bai et al., 2025) to increase the number of relation instances for each predefined predicate type.

Following the CoT prompt (see Appendix A.5), the strategy first generates an image description conditioned on ground-truth entities. This description, alongside predefined predicate types, guides the subsequent generation of relation triplets. To ensure validity, rule-based heuristics enforce that subjects and objects align with ground-truth annotations and predicates strictly belong to the available category set. This approach substantially increases training examples and introduces semantic diversity (detailed analyses in Appendix B.1 and B.3). To mitigate the generation of implausible relations, we maintain augmentation quality by encoding each generated triplet t into an embedding $\mathbf{e}_t$ via Sentence-BERT (Reimers and Gurevych, 2019). A triplet t is retained exclusively when its cosine similarity with any ground-truth triplet embedding exceeds a threshold $\theta \in [0, 1]$:

$$\mathrm{A}(t) = \mathbb{I}\Big[\exists\, t_{\text{gt}} \in \mathcal{T}_{\text{gt}} : \ \mathrm{sim}(\mathbf{e}_t, \mathbf{e}_{t_{\text{gt}}}) \geq \theta\Big] \tag{1}$$

where $\mathcal{T}_{\text{gt}}$ denotes the set of ground-truth triplets. This filtering mechanism improves the quality of SFT data, facilitating a generalizable model that provides robust initialization for subsequent RL.

3.4 Reward Modeling

Reward modeling is central to reinforcement learning with verifiable rewards (RLVR). We introduce a composite reward scheme meticulously aligned with the CoT pipeline, incorporating both process rewards to guide reasoning at each stage and format rewards to standardize structured output. This joint optimization simultaneously enhances visual grounding and relation extraction, facilitating robust end-to-end SGG.

3.4.1 Format Reward

The format reward R_{format} optimizes structural integrity by verifying the completeness of required tags and correctness of JSON syntax at each stage.

3.4.2 Category Detection Reward

The category recognition reward R_{category} is measured on the F1 score between predicted and ground-truth category sets, encouraging comprehensive detection while penalizing over-prediction to mitigate reward hacking. Here, $P = \mathrm{TP}/(\mathrm{TP} + \mathrm{FP})$ and $R = \mathrm{TP}/(\mathrm{TP} + \mathrm{FN})$, where TP, FP, and FN denote true positives, false positives, and false negatives, respectively.

$$R_{\text{category}} = \frac{2PR}{P + R} \tag{2}$$

3.4.3 Instance Grounding Reward

The instance grounding reward R_{node} including R_{box} and R_{recall} is computed via bipartite matching between predicted and ground-truth object pairs, as detailed in the Appendix A.7. Following prior work (Chen et al., 2025), R_{box} combines IoU and L_1 distance to measure bounding box accuracy. Recognizing the limitations of MLLMs in object detection recall (Li et al., 2025a), we introduce a recall-oriented reward R_{recall} to explicitly incentivize high-recall matching. For a ground-truth object $v_j = (b_j, c_j)$ matched to a predicted object $v_i = (b_i, c_i)$, the per-pair recall r_j is formulated as:

$$r_j = \begin{cases} 1.0, & \mathrm{IoU}(b_j, b_i) > 0.5 \wedge (c_j = c_i) \\ 0.5, & \mathrm{IoU}(b_j, b_i) > 0.5 \oplus (c_j = c_i) \\ 0.0, & \text{otherwise} \end{cases} \tag{3}$$

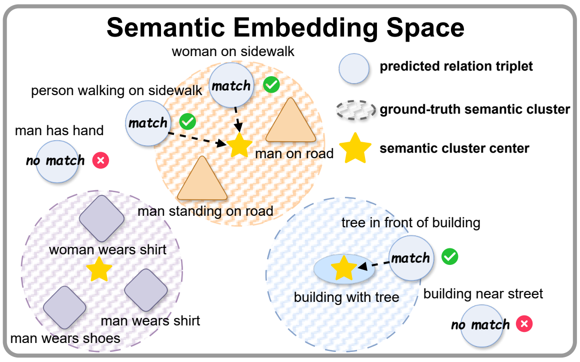


Figure 3: Coarse-grained semantic clustering reward. A triplet is matched if its semantic embedding aligns with any ground-truth cluster centroid beyond a threshold, relaxing the strict matching requirement.

The overall instance grounding reward is then defined as the average of these per-pair scores across all ground-truth instances.

3.4.4 Dual-granularity Reward

The dual-granularity reward combines fine-grained and coarse-grained rewards to jointly optimize relation recall and semantic coverage, leveraging Sentence-BERT to obtain the relation embeddings. The fine-grained reward incorporates frequency-based adaptive weighting of predicates to alleviate long-tail bias in relation extraction. For each ground-truth triplet t_j and its matched prediction t_i, let $\mathbf{e}_t$ and $\mathbf{e}_p$ denote the embeddings of the triplet and its predicate, respectively. The similarity $\mathrm{sim_j}$ is calculated as:

$$\mathrm{sim_j} = \mathrm{sim_t}(\mathbf{e}_{t_j}, \mathbf{e}_{t_i}) \times \mathrm{sim_p}(\mathbf{e}_{p_j}, \mathbf{e}_{p_i}) \tag{4}$$

where $\mathrm{sim_t}$ and $\mathrm{sim_p}$ denote the cosine similarities of triplet and predicate embeddings, respectively (with $\mathrm{sim_j} = 0$ for unmatched triplets). The fine-grained reward is formulated as:

$$R_{\mathrm{fine}} = \frac{\sum_{t_j \in \mathcal{T}_{\mathrm{gt}}} \mathrm{sim_j} \times w(p_j)}{\sum_{t_j \in \mathcal{T}_{\mathrm{gt}}} w(p_j)} \tag{5}$$

where $w(p)$ is an adaptive weight based on the frequency of ground-truth predicate, defined as:

$$w(p) = w_{\mathrm{base}} + w_{\mathrm{inc}} \cdot \alpha(p) \tag{6}$$

$$\alpha(p) = \frac{\log(1/f_p) - \log(1/f_{\max})}{\log(1/f_{\min}) - \log(1/f_{\max})} \tag{7}$$

Here, f_p, $f_{\max}$, and $f_{\min}$ denote the frequency of predicate p, the maximum and the minimum predicate frequency in the training data. Meanwhile, w_{base} and w_{inc} represent the base weight and the incremental weight.

The coarse-grained reward R_{coarse} is designed to enhance model generalization beyond the limited sparse annotated data. It employs a cluster algorithm to encourage the model to generate relation triplets that are semantically proximate to ground-truth, without requiring exact subject and object matching, as illustrated in Figure 3. First, ground-truth triplets are embedded and clustered into semantic prototypes $\mathcal{C}_{\mathrm{GT}}$ using DBSCAN. Each resulting cluster $c_j \in \mathcal{C}_{\mathrm{GT}}$ is represented by the mean embedding $\mathbf{e}_{c_j}$ of its member triplets. The reward is decomposed into two complementary metrics. Cluster coverage N_{covered} measures the number of ground-truth clusters that are matched by at least one predicted triplet t_i, where the similarity threshold is set as $\tau \in (0,1)$:

$$N_{\mathrm{covered}} = \sum_{c_j \in \mathcal{C}_{\mathrm{GT}}} \mathbb{I}\left[\exists\, t_i \in \mathcal{T}_{\mathrm{pred}} : \mathrm{sim}(\mathbf{e}_{t_i}, \mathbf{e}_{c_j}) \geq \tau\right] \tag{8}$$

where $\mathcal{T}_{\mathrm{pred}}$ is the set of predicted triplets. Cluster density ρ_{covered} assesses the concentration of predictions within the matched clusters:

$$\rho_{\mathrm{covered}} = \frac{\sum_{c_j \in \mathcal{C}_{\mathrm{covered}}} N_{\mathrm{pred}}^{c_j}}{\sum_{c_j \in \mathcal{C}_{\mathrm{covered}}} N_{\mathrm{gt}}^{c_j}} \tag{9}$$

Here, $N_{\mathrm{pred}}^{c_j}$ and $N_{\mathrm{gt}}^{c_j}$ denote the number of predicted and ground-truth triplets in cluster c_j, respectively, and $\mathcal{C}_{\mathrm{covered}}$ is the subset of ground-truth clusters that are successfully matched. Finally, the coarse-grained reward combines coverage and density to form a unified metric:

$$R_{\mathrm{coarse}} = \left(\frac{N_{\mathrm{covered}}}{|\mathcal{C}_{\mathrm{GT}}|}\right) \times \rho_{\mathrm{covered}} \tag{10}$$

3.5 Reinforcement Learning with Group Sequence Policy Optimization

We adopt Group Sequence Policy Optimization (GSPO) (Zheng et al., 2025) as the online algorithm for the RL phase. To address the high-variance training noise and instability associated with token-level importance sampling in large-scale RL, the GSPO objective $\mathcal{J}_{\mathrm{GSPO}}$ extends GRPO (Shao et al., 2024b) by adopting a sequence-level importance ratio $s_i(\theta)$, replacing the original token-level version, which is formulated as:

$$\mathcal{J}_{\mathrm{GSPO}}(\theta) = \mathbb{E}_{x \sim \mathcal{D},\, \{y_i\}_{i=1}^{G} \sim \pi_{\theta_{\mathrm{old}}}(\cdot|x)} \left[\frac{1}{G} \sum_{i=1}^{G} \min \left(s_i(\theta)\widehat{A}_i,\ \mathrm{clip}\big(s_i(\theta), 1-\varepsilon, 1+\varepsilon\big)\widehat{A}_i \right) \right] \tag{11}$$

Here, the sequence-level importance ratio $s_i(\theta)$ is defined as the geometric mean of the policy ratio $\pi_\theta/\pi_{\theta_{\text{old}}}$ over the entire output sequence y_i for numerical stability. This sequence-level formulation enhances optimization robustness for the generation of long-form, structured JSON outputs.

$$s_i(\theta) = \exp\left(\frac{1}{|y_i|}\sum_{t=1}^{|y_i|}\log\frac{\pi_\theta(y_{i,t}|x, y_{i,<t})}{\pi_{\theta_{\text{old}}}(y_{i,t}|x, y_{i,<t})}\right) \tag{12}$$

The group-normalized advantage $\widehat{A}_i$ is retained as the advantage, and estimated by normalizing the reward $r(x, y_i)$ using the mean and standard deviation of rewards within the current group, without relying on a learned value function V:

$$\widehat{A}_i = \frac{r(x, y_i) - \text{mean}\left(\{r(x, y_i)\}_{i=1}^{G}\right)}{\text{std}\left(\{r(x, y_i)\}_{i=1}^{G}\right)} \tag{13}$$

Method	Params	VG150 Dataset			
		Recall	mRecall	Mean	zsRecall
Non-VLM Methods					
MOTIFS	–	30.3	6.6	18.5	0.2
VCTree	–	31.3	8.0	19.7	0.7
GPS-Net	–	31.7	9.8	20.8	–
PE-Net	–	35.2	14.5	24.9	3.6
RelTR	–	30.7	12.6	21.7	2.4
SGTR	–	28.4	15.2	21.8	5.8
SGTR+	–	30.1	17.0	23.6	5.8
HiLo	–	29.2	**17.6**	23.4	–
Pair-Net	–	29.3	15.4	22.4	–
DSGG	–	38.5	<u>17.3</u>	**27.9**	3.9
EGTR	–	34.3	10.1	22.2	–
VLM-based Methods					
VS[3]	–	<u>40.9</u>	7.8	24.4	–
OvSGTR	–	**41.3**	8.8	25.1	–
PGSG	–	23.6	12.7	18.2	**7.6**
OwSGG†	72B	3.2	3.4	3.3	2.0
R1-SGG†	7B	27.6	10.9	19.3	4.4
Ours					
SGG-R[3]‡ (SFT)	3B	17.5	8.2	12.9	2.4
SGG-R[3]‡ (SFT + RL)	3B	36.0	14.8	<u>25.4</u>	<u>6.1</u>

Table 1: Performance (%) comparison of our proposed method against non-VLM and VLM-based baselines on the VG150 test set. Bold text indicates best performance, underlined text indicates second best. The "Mean" denotes the average of Recall and mean Recall. † and ‡ denote Qwen2-VL and Qwen2.5-VL as the base model respectively.

4 Experiments

4.1 Experiment Setup

Datasets. We evaluate our method on two standard SGG benchmarks. The first is the widely adopted Visual Genome (VG150) dataset (Xu et al., 2017), which includes 150 object categories and 50 relation categories, and follows the prior work (Chen et al., 2024, 2025) to split image-graph pairs with 56,224 training, 5,000 validation, and 14,700 testing. The second is the Panoptic Scene Graph (PSG) dataset (Yang et al., 2022), constructed based on COCO (Lin et al., 2014), containing 80 thing categories, 53 stuff categories, and 56 relation categories. It provides 46,563 training pairs and 2,186 test pairs for evaluation.

Evaluation Metrics. We follow the standard SGDET (Xu et al., 2017) evaluation protocol for SGG, in which models must generate scene graphs directly from images without ground-truth object bounding boxes. The main evaluation metrics are Recall, mean Recall (mRecall), and zero-shot Recall (zsRecall). A triplet is correct only if both the subject and object bounding boxes have IoU ≥ 0.5 with their matched ground-truth boxes, and the subject, object, and predicate categories are all correctly predicted. Recall is computed among all predicate categories, mRecall averages the recall per predicate category, and zsRecall specifically measures the models generalization to relation triplets that are unseen during training. We adopt $K = 100$ for all recall calculations.

Implementation Details. Our implementation is built upon the TRL (von Werra et al., 2020) library, with vLLM (Kwon et al., 2023) for accelerated sampling during the RL stage. The training is conducted on 8 NVIDIA A800 (40GB) GPUs. Implementation details are provided in the Appendix A.1.

4.2 Comparisons with State-of-the-Arts

We benchmark our framework on the VG150 and PSG test sets against a comprehensive suite of baselines, including traditional non-VLM methods (Zellers et al., 2018; Tang et al., 2019; Lin et al., 2020; Yang et al., 2022; Li et al., 2022; Zheng et al., 2023; Cong et al., 2023; Zhou et al., 2023; Im et al., 2024; Li et al., 2024a; Wang et al., 2024a; Hayder and He, 2024) and VLM-based approaches (Zhang et al., 2023; Chen et al., 2024; Li et al., 2024b; Wang et al., 2024c; Xu et al., 2025; Dutta et al., 2025; Chen et al., 2025; Li et al., 2025b), as summarized in Table 1 and Table 2.

We adopt Qwen2.5-VL-3B (Bai et al., 2025) as the base model and conduct experiments under two settings: one trained solely with SFT, and another further optimized via RL. Initially, it can be observed that the model's performance after SFT is lower than that of previous methods on both datasets. However, after further RL training initialized with the SFT model, the model shows substantial improvements across all metrics, with

Method	Params	PSG Dataset			
		Recall	mRecall	Mean	zsRecall
Non-VLM Methods					
MOTIFS	–	22.0	9.1	15.6	–
VCTree	–	20.6	9.7	15.2	–
GPS-Net	–	20.6	7.2	13.9	–
PSGFormer	–	21.0	19.8	20.4	4.9
PSGTR	–	35.3	21.5	28.4	6.4
SGTR	–	32.0	23.5	27.8	3.6
HiLo	–	51.4	40.9	46.2	–
Pair-Net	–	42.4	29.7	36.1	–
DSGG	–	50.0	43.4	46.7	–
VLM-based Methods					
OvSGTR	–	41.4	28.3	34.9	–
PGSG	–	33.4	22.1	27.8	**8.9**
ASMv2	13B	14.8	11.8	13.3	–
LLaVA-SpaceSGG	13B	15.4	13.2	14.3	–
OwSGG†	72B	14.0	13.7	13.9	6.4
R1-SGG†	7B	43.5	33.2	38.4	7.7
Relation-R1‡	3B	25.9	21.3	23.6	–
Ours					
SGG-R^3‡ (SFT)	3B	33.3	26.1	29.7	0.0
SGG-R^3‡ (SFT + RL)	3B	**52.5**	**44.3**	**48.4**	7.7

Table 2: Performance (%) comparison of our proposed method against non-VLM and VLM-based baselines on the PSG test set. Bold text indicates best performance, underlined text indicates second best. The "Mean" denotes the average of Recall and mean Recall. † and ‡ denote Qwen2-VL and Qwen2.5-VL as the base model respectively.

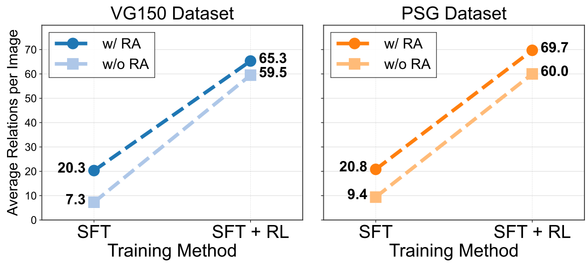


Figure 4: Quantitative analysis of the average number of relations generated per image on the VG150 and PSG test sets: SFT, SFT + RL; with/without relation augmentation (RA).

qualitative comparison provided in Appendix B.4. This indicates that RL significantly enhances the capability for relation extraction.

On VG150 shown in Table 1, our method achieves the highest mRecall among all VLM-based approaches while maintaining a competitive Recall. Although our mRecall outperforms most baselines, it lags slightly behind some non-VLM methods (Zhou et al., 2023; Wang et al., 2024a; Hayder and He, 2024). We attribute this gap to the exhaustive object-pair coverage inherent in classification frameworks, which remains a persistent challenge for generative models. Notably, the mean of our Recall and mRecall ranks second among all evaluated methods. Our approach also surpasses all non-VLM models in zsRecall, demonstrating superior generalization and the ability to recognize diverse relations.

Furthermore, our method demonstrates leading performance on the PSG dataset, as detailed in Table 2. It significantly outperforms all multimodal methods and marginally exceeds non-VLM approaches in terms of Recall and mRecall, indicating a substantial reduction in relation bias. The average of our Recall and mRecall metrics also ranks first among all evaluated methods, substantially surpassing most baselines. In terms of zsRecall, our model achieves highly competitive performance, although it falls slightly short of (Li et al., 2024b). In summary, our framework achieves leading overall performance against both non-VLM and VLM-based baselines on the two benchmarks.

4.3 Comparisons with MLLM-based Methods

Relation Recognition Evaluation. As shown in Table 3, we compare our proposed method with state-of-the-art MLLM-based baselines (Chen et al., 2025), which directly generate objects and relations without a structured reasoning process. Following the baselines, we use the Failure Rate metric to assess the structural integrity and validity of the generated JSON format. For a fair comparison, we report results based on both Qwen2-VL (Wang et al., 2024b) and Qwen2.5-VL, evaluating the performance with and without our relation augmentation. Furthermore, a detailed analysis of MLLM zero-shot performance in SGG is provided in Appendix B.2.

When trained on equivalent data volumes, our 2B and 3B SFT models demonstrate superior robustness over baselines. Specifically, the 2B model achieves a 57.23% reduction in VG150 failure rates, validating the CoT framework's efficacy in improving instruction following ability and structural correctness. While relation sparsity on PSG initially limits gains, subsequent RL optimization enables the 3B model to surpass the 7B baseline across all metrics, with mRecall increasing by 1.25% on VG150 and 4.62% on PSG. These results indicate that for models of comparable capacity, refined post-training is the primary performance driver.

During the SFT phase, relation augmentation further reduces the output failure rate. For instance, our 2B SFT model achieves a 26.17% lower failure rate than the 7B baseline on VG150,

Method	Training	Params	VG150 Dataset				PSG Dataset			
			Failure Rate	Recall	mRecall	Mean	Failure Rate	Recall	mRecall	Mean
Baselines										
R1-SGG†	SFT	2B	75.01	4.82	1.04	2.93	9.79	21.83	13.79	17.81
R1-SGG‡	SFT	3B	63.16	6.60	1.61	4.11	6.30	22.63	14.60	18.62
R1-SGG†	SFT	7B	39.54	9.62	3.30	6.46	0.96	24.73	17.11	20.92
R1-SGG†	SFT + RL	2B	0.10	21.09	7.48	14.29	2.70	38.49	31.21	34.85
R1-SGG†	SFT + RL	7B	0.12	23.75	11.43	17.59	**0.00**	43.48	33.71	38.60
Ours (w/o RA)										
SGG-R^{3†}	SFT	2B	17.78	9.93	2.08	6.01	0.23	22.66	12.25	17.46
SGG-R^{3‡}	SFT	3B	6.24	11.98	2.36	7.17	0.18	22.27	11.45	16.86
SGG-R^{3‡}	SFT + RL	3B	0.12	27.85	12.68	20.27	**0.00**	47.63	38.33	42.98
Ours (w/ RA)										
SGG-R^{3†}	SFT	2B	13.37	13.90	8.08	10.99	0.05	33.06	24.67	28.87
SGG-R^{3‡}	SFT	3B	3.30	14.88	8.80	11.84	0.05	33.31	26.13	29.72
SGG-R^{3‡}	SFT + RL	3B	**0.06**	**30.66**	**14.52**	**22.59**	**0.00**	**52.49**	**44.30**	**48.40**

Table 3: Performance (%) comparison with MLLM-based end-to-end SGG baselines on the VG150 validation set and PSG test set. The "Mean" denotes the average of Recall and mean Recall. † and ‡ denote Qwen2-VL and Qwen2.5-VL as the base model, respectively.

suggesting a synergistic effect between the increased quantity of relational data and our structured reasoning process, which in turn enhances the structural integrity of the generated outputs. Specifically, our 2B and 3B SFT models demonstrate superior Recall and mRecall on both datasets. Subsequent RL training successfully maintains this performance advantage. Notably, our 3B model maintains a substantial lead over the 7B baseline: on VG150, it outperforms the baseline by 6.91% in Recall and 3.09% in mRecall, while on the PSG dataset, these gains are even more pronounced at 9.01% and 10.59%, respectively. Qualitative visualizations of our SGG results are provided in Appendix B.5.

Object Detection Evaluation. In SGG, we adopt Recall and mean Recall as the primary metrics for evaluating object detection, rather than the standard Average Precision (AP). This choice stems from the task's requirement to identify all plausible objects (even unlabeled ones) to support relation prediction. Here, a predicted object is considered a match if its IoU with a ground-truth box ≥ 0.5 and their categories match.

As reported in Table 4, our SFT models significantly outperforms the baselines across all metrics on both datasets. For instance, the 3B model achieves gains of 19.14% in Recall and 14.35% in mRecall on the VG150 dataset, demonstrating superior object detection capabilities. Notably, RL optimization yields a substantial boost over the 3B SFT model, it simultaneously enhances both Recall and mRecall, surpassing the 7B baseline by 10.58% in Recall and 9.31% in mRecall on VG150. Meanwhile, the zero-shot performance

Method	Training	Params	VG150		PSG	
			Recall	mRecall	Recall	mRecall
Open-sourced MLLMs						
Qwen2.5-VL	zero-shot	32B	24.84	25.39	35.66	39.52
Baselines						
R1-SGG†	SFT	2B	10.72	14.76	47.43	46.72
R1-SGG‡	SFT	3B	14.98	18.89	51.97	49.94
R1-SGG†	SFT + RL	7B	39.49	38.26	52.08	52.94
Ours						
SGG-R^{3†}	SFT	2B	32.04	31.15	56.52	53.78
SGG-R^{3‡}	SFT	3B	34.12	33.24	56.50	53.46
SGG-R^{3‡}	SFT + RL	3B	**50.07**	**47.57**	**61.32**	**59.27**

Table 4: Object detection performance (%) comparison of our method against the baselines on the VG150 and PSG test sets. † and ‡ indicate the use of Qwen2-VL and Qwen2.5-VL as the base models.

of Qwen2.5-VL-32B falls short of both the baseline and our proposed approach. These results indicate that our CoT framework improves both the precision and coverage of object recognition. Furthermore, the category and instance grounding rewards during RL effectively guide the model toward identifying a more exhaustive set of valid object instances, thereby establishing a robust foundation for comprehensive scene graph generation.

4.4 Ablation Study

To verify the efficacy of the proposed relation augmentation (RA) and dual-granularity reward (DGR), we conduct extensive ablation studies. Specifically, we categorize predicates into Head, Body, and Tail classes based on training frequency to evaluate each module's contribution to long-tail relation recognition. Details regarding these partitions are provided in Appendix A.2.

Effectiveness of Relation Augmentation. Table 5 shows that RA significantly boosts performance in both SFT and RL stages. Specifically,

Setup			VG150 Dataset						PSG Dataset					
RA	DGR	Training	Recall	mRecall	zsRecall	Head	Body	Tail	Recall	mRecall	zsRecall	Head	Body	Tail
Relation Augmentation (RA) Ablation														
×	×	SFT	12.88	2.41	0.90	13.99	1.13	0.53	22.27	11.45	0.00	22.32	14.92	1.21
✓	×	SFT	17.74	8.23	2.35	18.67	9.90	4.96	33.31	26.13	0.00	32.24	33.04	16.01
×	✓	SFT + RL	31.57	12.62	5.03	32.95	14.90	4.76	47.63	38.33	0.00	46.98	43.10	24.77
✓	✓	SFT + RL	35.95	14.78	6.11	36.98	15.58	8.13	52.49	44.30	7.69	50.95	50.84	34.14
Dual-granularity Reward (DGR) Ablation														
✓	×	SFT + RL	34.55	13.94	5.01	35.58	15.07	7.22	50.40	42.26	7.21	49.25	46.61	32.33
✓	✓	SFT + RL	35.95	14.78	6.11	36.98	15.58	8.13	52.49	44.30	7.69	50.95	50.84	34.14

Table 5: Performance (%) comparison for ablation studies of relation augmentation (RA) and dual-granularity reward (DGR) modules. ✓ indicates the component is enabled, × indicates the component is disabled.

RA enhances the SFT model's ability to recognize body and tail predicates, yielding gains of 8.77% and 4.43% on VG150, and 18.12% and 14.80% on PSG. Building on this RA-enhanced initialization, RL further elevates Recall and mRecall, with tail performance increasing by 3.37% on VG150 and 9.37% on PSG. These findings suggest that RA provides a superior foundation for RL to explore long-tail relations. Notably, while RL training without RA fails to achieve any zsRecall on PSG, its incorporation yields a 7.69% improvement, which underscores RA's pivotal role in facilitating a generalized "cold start".

Additionally, Figure 4 compares the average volume of generated relations per image with and without RA. Experimental results demonstrate that while the inherent sparsity of the raw data limits relation density, RA significantly promotes relation generation. Specifically, RA-enhanced models output an additional 13.0 and 11.4 relations per image on VG150 and PSG after SFT. Furthermore, RL training further stimulates reasoning depth, leading to a continued increase in the total number of relations. Throughout the RL phase, the model initialized with RA consistently maintains a higher relation density than its non-RA counterpart across both datasets.

Impact of Dual-granularity Reward. We evaluate the impact of DGR by comparing it against a baseline utilizing a standard relation recall reward. As shown in Table 5, with the RA-enhanced SFT model as initialization, DGR yields consistent gains across both Recall and mRecall. For instance, on the PSG dataset, DGR improves Recall by 2.09% and mRecall by 2.04%, demonstrating its effectiveness in mitigating the long-tail distribution of relations. Specifically, DGR facilitates balanced improvements across all predicate types, thereby mitigating generative bias. Furthermore, by leveraging semantic clustering to guide the optimization process, DGR enhances the model's generalization to novel or rare relational contexts, as evidenced by a 1.10% increase in zsRecall on the VG150 dataset.

5 Conclusion

This paper presents SGG-R^3, a structured reasoning framework that aligns next-token prediction with end-to-end unbiased scene graph generation via a two-phase training pipeline. We propose a structured reasoning process encompassing category detection, instance grounding, and multi-type relation extraction. To generate comprehensive and unbiased scene graphs, we employ a relation-augmentation strategy to mitigate the cold-start challenge during SFT and introduce a dual-granularity reward scheme during RL. Empirical evaluations on the VG150 and PSG benchmarks demonstrate that SGG-R^3 achieves superior capabilities of visual grounding and relation extraction, exhibiting significant potential for diverse downstream applications.

Limitations

The proposed framework has limitations in efficiency, data fidelity, and label coverage. First, the use of MLLMs leads to non-negligible inference latency, which may constrain its use in real-time applications. Second, although we use filtering strategies, the relation augmentation process cannot guarantee entirely accurate triples; some erroneous or hallucinated data may still persist. The quality of this augmented data essentially sets the upper bound for model performance. Finally, our training is currently limited to a closed-set vocabulary, which may restrict the model's ability to recognize novel relations in the wild. Future work will focus on extending this framework to open-vocabulary settings to improve its generalization across diverse, real-world scenarios.

Acknowledgements

This work was supported in part by the National Key Research and Development Program of China (Grant No. 2023YFC3304902).

References

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, and 1 others. 2025. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*.

Zuyao Chen, Jinlin Wu, Zhen Lei, Marc Pollefeys, and Chang Wen Chen. 2025. Compile scene graphs with reinforcement learning. *arXiv preprint arXiv:2504.13617*.

Zuyao Chen, Jinlin Wu, Zhen Lei, Zhaoxiang Zhang, and Chang Wen Chen. 2024. 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.

Xu Chu, Xinrong Chen, Guanyu Wang, Zhijie Tan, Kui Huang, Wenyu Lv, Tong Mo, and Weiping Li. 2025. Qwen look again: Guiding vision-language reasoning models to re-attention visual information. *arXiv preprint arXiv:2505.23558*.

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

Amartya Dutta, Kazi Sajeed Mehrab, Medha Sawhney, Abhilash Neog, Mridul Khurana, Sepideh Fatemi, Aanish Pradhan, M Maruf, Ismini Lourentzou, Arka Daw, and 1 others. 2025. Open world scene graph generation using vision language models. *arXiv preprint arXiv:2506.08189*.

Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, and 1 others. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. In *kdd*, pages 226–231.

Samir Yitzhak Gadre, Kiana Ehsani, Shuran Song, and Roozbeh Mottaghi. 2022. Continuous scene representations for embodied ai. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14849–14859.

Zeeshan Hayder and Xuming He. 2024. Dsgg: Dense relation transformer for an end-to-end scene graph generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 28317–28326.

Jiani Huang, Amish Sethi, Matthew Kuo, Mayank Keoliya, Neelay Velingker, JungHo Jung, Ser-Nam Lim, Ziyang Li, and Mayur Naik. 2025a. Esca: Contextualizing embodied agents via scene-graph generation. *arXiv preprint arXiv:2510.15963*.

Jing Huang, Zhiya Tan, Shutao Gong, Fanwei Zeng, Joey Tianyi Zhou, Changtao Miao, Huazhe Tan, Weibin Yao, and Jianshu Li. 2025b. Lav-cot: Language-aware visual cot with multi-aspect reward optimization for real-world multilingual vqa. *arXiv preprint arXiv:2509.10026*.

Jinbae Im, JeongYeon Nam, Nokyung Park, Hyungmin Lee, and Seunghyun Park. 2024. Egtr: Extracting graph from transformer for scene graph generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 24229–24238.

Franklin Kenghagho Kenfack, Feroz Ahmed Siddiky, Ferenc Balint-Benczedi, and Michael Beetz. 2020. Robotvqaa scene-graph-and deep-learning-based visual question answering system for robot manipulation. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 9667–9674. IEEE.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th symposium on operating systems principles*, pages 611–626.

Jincheng Li, Chunyu Xie, Ji Ao, Dawei Leng, and Yuhui Yin. 2025a. Lmm-det: Make large multimodal models excel in object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 308–318.

Lin Li, Wei Chen, Jiahui Li, Kwang-Ting Cheng, and Long Chen. 2025b. Relation-r1: Progressively cognitive chain-of-thought guided reinforcement learning for unified relation comprehension. *arXiv preprint arXiv:2504.14642*.

Rongjie Li, Songyang Zhang, and Xuming He. 2022. 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.

Rongjie Li, Songyang Zhang, and Xuming He. 2024a. Sgtr+: End-to-end scene graph generation with transformer. *arXiv preprint arXiv:2401.12835*.

Rongjie Li, Songyang Zhang, Dahua Lin, Kai Chen, and Xuming He. 2024b. 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.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer.

Xin Lin, Changxing Ding, Jinquan Zeng, and Dacheng Tao. 2020. 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.

Bingshuai Liu, Chenyang Lyu, Zijun Min, Zhanyu Wang, Jinsong Su, and Longyue Wang. 2025. Retrieval-augmented multi-modal chain-of-thoughts reasoning for large language models. In *2025 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992.

Brigit Schroeder and Subarna Tripathi. 2020. Structured query-based image retrieval using scene graphs. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 178–179.

Hao Shao, Shengju Qian, Han Xiao, Guanglu Song, Zhuofan Zong, Letian Wang, Yu Liu, and Hongsheng Li. 2024a. Visual cot: Advancing multi-modal language models with a comprehensive dataset and benchmark for chain-of-thought reasoning. *Advances in Neural Information Processing Systems*, 37:8612–8642.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, and 1 others. 2024b. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Kaihua Tang, Hanwang Zhang, Baoyuan Wu, Wenhan Luo, and Wei Liu. 2019. 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.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Leandro von Werra, Younes Belkada, Lewis Tunstall, Edward Beeching, Tristan Thrush, Nathan Lambert, Shengyi Huang, Kashif Rasul, and Quentin Gallouédec. 2020. Trl: Transformer reinforcement learning. https://github.com/huggingface/trl.

Jinghao Wang, Zhengyu Wen, Xiangtai Li, Zujin Guo, Jingkang Yang, and Ziwei Liu. 2024a. Pair then relation: Pair-net for panoptic scene graph generation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, and 1 others. 2024b. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*.

Weiyun Wang, Yiming Ren, Haowen Luo, Tiantong Li, Chenxiang Yan, Zhe Chen, Wenhai Wang, Qingyun Li, Lewei Lu, Xizhou Zhu, and 1 others. 2024c. The all-seeing project v2: Towards general relation comprehension of the open world. In *European Conference on Computer Vision*, pages 471–490. Springer.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

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

Guowei Xu, Peng Jin, Ziang Wu, Hao Li, Yibing Song, Lichao Sun, and Li Yuan. 2025. Llava-cot: Let vision language models reason step-by-step. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2087–2098.

Jingkang Yang, Yi Zhe Ang, Zujin Guo, Kaiyang Zhou, Wayne Zhang, and Ziwei Liu. 2022. Panoptic scene graph generation. In *European conference on computer vision*, pages 178–196. Springer.

Qifan Yu, Juncheng Li, Yu Wu, Siliang Tang, Wei Ji, and Yueting Zhuang. 2023. 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.

Xiang Yuan, Xu Chu, Xinrong Chen, Haochen Li, Zonghong Dai, Hongcheng Fan, Xiaoyue Yuan, Weiping Li, and Tong Mo. 2026. More-r1: Guiding lvlm for multimodal object-entity relation extraction via stepwise reasoning with reinforcement learning. *arXiv preprint arXiv:2603.09478*.

Rowan Zellers, Mark Yatskar, Sam Thomson, and Yejin Choi. 2018. Neural motifs: Scene graph parsing with global context. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5831–5840.

Xiang Zhang, Juntai Cao, Chenyou You, and Dujian Ding. 2025. Why prompt design matters and works:

A complexity analysis of prompt search space in llms. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 32525–32555.

Yong Zhang, Yingwei Pan, Ting Yao, Rui Huang, Tao Mei, and Chang-Wen Chen. 2023. 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.

Chaofan Zheng, Xinyu Lyu, Lianli Gao, Bo Dai, and Jingkuan Song. 2023. Prototype-based embedding network for scene graph generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22783–22792.

Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang, Yuqiong Liu, Rui Men, An Yang, and 1 others. 2025. Group sequence policy optimization. *arXiv preprint arXiv:2507.18071*.

Zijian Zhou, Miaojing Shi, and Holger Caesar. 2023. Hilo: Exploiting high low frequency relations for unbiased panoptic scene graph generation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 21637–21648.

Yifeng Zhu, Jonathan Tremblay, Stan Birchfield, and Yuke Zhu. 2021. Hierarchical planning for long-horizon manipulation with geometric and symbolic scene graphs. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6541–6548. Ieee.

Appendix

A Implementation Details

A.1 Experiment Configurations

Our experiments are conducted on eight NVIDIA A800 (40GB) GPUs. During the supervised fine-tuning (SFT) stage, we employ the AdamW optimizer with a cosine learning rate scheduler, setting the initial learning rate to 1×10^{-5} for training 1 epoch. The per-device batch size is set to 2 with 2 gradient accumulation steps, resulting in an effective total batch size of 32 ($2 \times 2 \times 8$).

For the reinforcement learning (RL) phase, we utilize six GPUs for training and two GPUs for vLLM deployment to accelerate inference, using a sampling size of 8. The total batch size for RL is 96 ($8 \times 2 \times 6$). The model is trained for 1 epoch using the AdamW optimizer with a initial learning rate of 5×10^{-7}; further hyperparameter details are provided in Table 6. To compute semantic embeddings, we employ the all-MiniLM-L6-v2 (Reimers and Gurevych, 2019) model to project relation triplets or predicates into 384-dimensional embeddings.

The similarity threshold for filtering augmented relation triplets is set to 0.9. When applying DBSCAN to cluster ground-truth triplets, we empirically set the clustering threshold $\varepsilon = 0.75$ and $\texttt{min\_samples} = 1$ based on the distribution characteristics of the semantic space. During the evaluation phase, the similarity matching threshold between predicted triplets and their corresponding ground-truth clusters is likewise set to 0.75.

Hyperparameter	Value
Precision	bf16
Max completion length	2048
Per-device batch size	8
Gradient accumulation steps	2
Scheduler type	Cosine
Temperature	1.0
Top-p	0.9
Top-k	50
Group size	8
Num iterations per generation	1
Beta (β)	0.0
Random seed	42

Table 6: Hyperparameter settings for reinforcement learning training.

Category	VG150		PSG	
	Count	Freq.%	Count	Freq.%
Head	15	94.60	16	93.97
Body	15	3.95	16	5.32
Tail	20	1.45	24	0.71

Table 7: Distribution of long-tail predicate types across two datasets. The table shows the number and frequency of predicates in each category: Head (top 30%), Body (middle 30%), and Tail (bottom 40%).

A.2 Evaluation Setup

In the evaluation phase, we categorize predicates into **Head**, **Body**, and **Tail** based on their frequency in the training dataset to precisely assess model performance under long-tail distributions. Specifically, the top 30% most frequent predicates are designated as head, the middle 30% as body, and the bottom 40% as tail. We then evaluate the recall performance within each group. Detailed statistics regarding the predicate counts and frequencies for each category are provided in Table 7.

A.3 Semantic Taxonomy of Predicates

Recognizing that distinct relations necessitate specific semantic and visual cues and acknowledging that a single object pair may have multiple concurrent relationships depending on different perspectives, we partition the predicates of each dataset into three distinct taxonomies based on their underlying semantic properties. For VG150, relations are classified into **spatial** (e.g., *on, near*), **possessive** (e.g., *has, belong to*), and **interactive** (e.g., *wearing, standing on*) types. Similarly, for the PSG dataset, we distinguish between **spatial** (e.g., *on, in*), **static-interactive** (e.g., *sitting on, parked on*), and **dynamic-interactive** (e.g., *walking on, jumping over*) relations.

A.4 Structured Reasoning Prompt

We design a three-stage Chain-of-Thought (CoT) prompt (see Figure 9) to direct the structured output process of end-to-end Scene Graph Generation (SGG). In closed-set scenarios, the prompt incorporates the full set of object categories and relation predicate categories of three predefined types from the dataset. Specifically, the prompt guides the model through a sequential reasoning process based on these predefined categories, with each stage explicitly defining task objectives and constraints. To ensure robust parsing, all outputs are strictly constrained to standardized JSON formats within dedicated tags. Furthermore, we leverage in-context learning with representative examples to enhance the model's adherence to these structural constraints.

A.5 Relation Augmentation Prompt

To mitigate the inherent sparsity of relation annotations, we leverage Qwen2.5-VL-32B for data augmentation. Our CoT prompting framework (see Figure 10) integrates predicate categories of three predefined types and ground-truth object sets, including category names and bounding boxes. The prompt is structured into a sequential two-stage reasoning process, where each stage explicitly defines task objectives and requirements. To ensure precise instruction following and structural consistency, we also supplement representative in-context examples.

A.6 CoT Data Construction

We derive the CoT sequences from the original object annotations and augmented relation annotations. The data is organized into a three-stage sequential structure as illustrated in Figure 5:

Stage 1: Category Identification. We extract unique object categories from the object instance set to construct an initial list. Given M object instances $\mathcal{V} = \{(c_i, b_i)\}_{i=1}^M$, where c_i and b_i denote the category and bounding box respectively, we derive a unique category set $\mathcal{C} = \{c_n\}_{n=1}^N$ ($N \leq M$).

Stage 2: Sequential Instance Grounding. The instances in $\mathcal{V}$ are rearranged to align with the order of $\mathcal{C}$. To distinguish between multiple instances of the same category, we append unique numerical suffixes (e.g., *person.1, person.2*) to each instance.

Stage 3: Multi-relational Mapping. Finally, the set of K relation triplets $\mathcal{R} = \{(s_j, p_j, o_j)\}_{j=1}^K$ is organized. Triplets are first grouped by their predefined predicate types and then sorted based on the sequential order of subjects s_j established in Stage 2. This alignment ensures that relational reasoning consistently follows the grounded object sequence.

To maintain clarity throughout the reasoning process, we explicitly encapsulates each stage within dedicated tags, denoting the start and end of each stage: `<CATEGORY></CATEGORY>`, `<OBJECT></OBJECT>`, and `<RELATION></RELATION>`. All outputs are standardized in JSON format within these tags. This sequential organization aligns with the instruction structure of the CoT prompt.

A.7 Bipartite Matching

Following (Chen et al., 2025), the assignment between predicted and ground-truth entities is formulated as a bipartite matching problem. The objective is to find an optimal mapping between the set of predicted nodes $\mathcal{V}_{pred} = \{v_i = (c_i, b_i)\}_{i=1}^M$ and ground-truth nodes $\mathcal{V}_{gt} = \{v_j = (c_j, b_j)\}_{j=1}^N$, where each node consists of an object class c and its corresponding bounding box b. The matching cost between v_i and v_j is defined as:

$$\begin{aligned}\mathcal{L}_{match}(v_i, v_j) = &\lambda_1 \left(1 - \text{sim}(\mathbf{e}_{c_i}, \mathbf{e}_{c_j})\right) \\ &+ \lambda_2 \left(1 - \text{IoU}(b_i, b_j)\right) \\ &+ \lambda_3 \|b_i - b_j\|_1\end{aligned} \tag{14}$$

where $\text{sim}(\mathbf{e}_{c_i}, \mathbf{e}_{c_j})$ denotes the cosine similarity between semantic embeddings of object classes extracted via Sentence-BERT, $\text{IoU}(\cdot)$ is the Intersection over Union, and $\|\cdot\|_1$ represents the L_1 distance. The hyper-parameters λ_1, λ_2, and λ_3 serve as balancing coefficients that weight the relative contributions of semantic consistency, spatial overlap, and coordinate proximity, respectively.

To minimize false positives and enhance matching precision, we implement a threshold-based filtering mechanism. Only candidate pairs below a predefined cost threshold are retained for final assignment, which is determined by minimizing the global cost. This filtering provides a robust foundation for subsequent instance grounding optimization.

B Additional Analysis and Results

B.1 Dataset Analysis

Long-tail Analysis. Table 7 details the predicate distribution across both datasets, revealing a severe long-tail imbalance. In VG150, the head predicates comprising only 30% of the predicate categories, but accounting for over 90% of all training samples, leaving the body and tail categories significantly underrepresented. This extreme skewness highlights the substantial distribution bias inherent in these benchmarks, necessitating targeted strategies to improve rare category recognition.

Sparsity Analysis. As reported in Table 8, while the original datasets average approximately 11 objects per image, relation density remains disproportionately low at 4.8 for VG150 and 5.6 for PSG. This disparity indicates that most valid inter-object associations are unannotated, creating a sparse

Dataset	O/I	Before RA								After RA							
		Total		Head		Body		Tail		Total		Head		Body		Tail	
		Count	Type	Count	Type	Count	Type	Count	Type	Count	Type	Count	Type	Count	Type	Count	Type
VG150	11.7	269 196	25 151	254 656	20 592	10 645	2981	3985	1578	976 465	30 928	827 105	24 742	62 374	3320	86 986	2866
PSG	11.0	261 666	18 126	245 886	16 430	13 932	1287	1848	409	810 219	23 530	742 387	20 542	55 220	2150	12 612	838

Table 8: Statistics of relation augmentation on VG150 and PSG training sets, with breakdowns for head/body/tail categories. "O/I" denotes average objects per image; "Count" refers to the number of relation triplets; "Type" denotes the number of relation triple combination patterns.

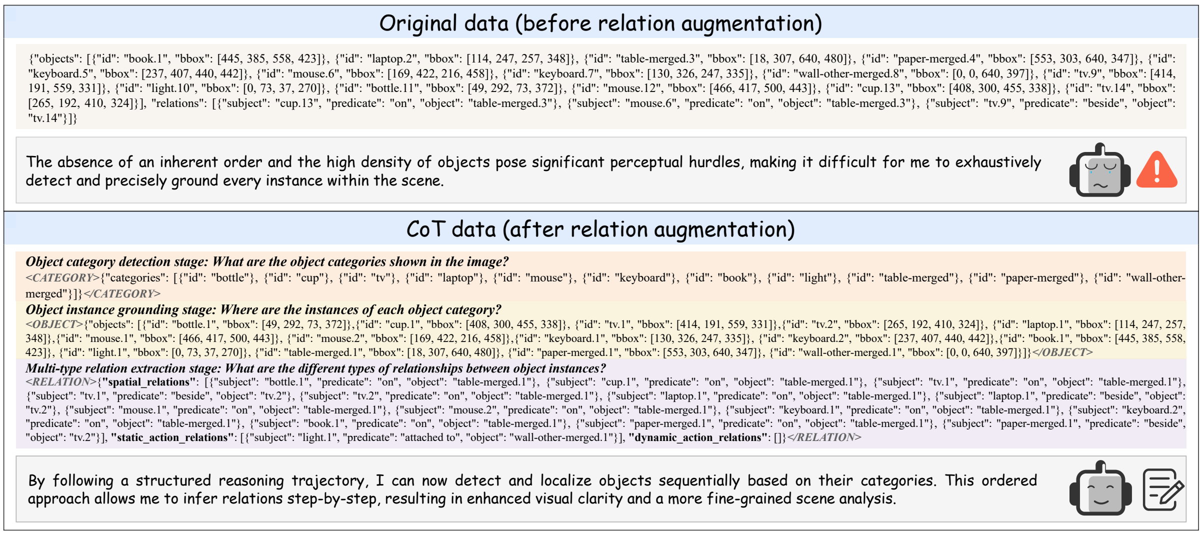


Figure 5: Comparison between original and CoT data. The original data are transformed into our three-stage format through a structured curation process.

scene graph that hinders MLLM fine-tuning. Following our RA strategy, the average relation count rises to 17.4 per image, and the number of relations increases across all frequency groups. By generating plausible annotations for previously neglected pairs, our approach substantially enriches the relational supervision for the SFT phase.

Diversity Analysis. As shown in Table 8, the results reveal that RA not only expands the total volume of relations but also significantly enriches the variety of triplet types across all frequency groups. This demonstrates its effectiveness in enhancing semantic diversity and provides more generalized triplet combination patterns for model learning in the SFT stage.

B.2 MLLM Zero-shot Performance in SGG

We evaluate the zero-shot end-to-end SGG capabilities of Qwen2.5-VL (3B, 7B, 32B) using our CoT prompt. As shown in Table 9, the results demonstrate that MLLMs possess the untapped potential to handle dense relational reasoning in a streamlined, end-to-end fashion, bypassing the need for complex, multi-stage pipelines.

However, zero-shot performance remains subpar across all metrics, with Recall, mRecall, and zsRecall persisting at extremely low levels compared to our fine-tuned methods. Severe deficits across head, body, and tail categories indicate a fundamental failure to capture complex relational dependencies. While scaling the model size significantly enhances instruction following ability and reduces failure rates, it does not resolve the core bottleneck in relational reasoning. For instance, despite the 32B model achieving a low 9.95% failure rate on VG150, its Recall of 8.40% is strikingly inferior to the 35.95% achieved by our fine-tuned approach. A similar disparity on PSG underscores the limitations of general-purpose MLLMs in dense relational reasoning under zero-shot settings, further validating the necessity of our proposed training pipeline.

B.3 Qualitative Analysis on Relation Augmentation

To mitigate implausible relations generated by Qwen2.5-VL-32B, such as <*hand.1, at, person.1*>, candidate relations are retained only if

Model	Params	VG150 Dataset							PSG Dataset						
		Failure Rate	Recall	mRecall	zsRecall	Head	Body	Tail	Failure Rate	Recall	mRecall	zsRecall	Head	Body	Tail
Zero-shot Performance															
Qwen2.5-VL	3B	76.95	0.46	0.34	0.12	0.46	0.31	0.13	51.05	3.69	3.69	0.00	2.66	4.15	2.42
Qwen2.5-VL	7B	24.89	3.40	2.38	0.78	3.09	3.26	1.98	8.69	8.78	11.02	0.00	6.82	10.45	7.25
Qwen2.5-VL	32B	9.95	8.40	5.66	1.68	8.20	6.13	4.96	5.67	15.69	17.38	0.00	14.03	18.60	13.90
Fine-tuned Performance															
Qwen2.5-VL	3B	0.06	35.95	14.78	6.11	36.98	15.58	8.13	0.00	52.49	44.30	7.69	50.95	50.84	34.14

Table 9: Performance (%) comparison of Qwen2.5-VL and our fine-tuned approach on the VG150 and PSG test sets.

their maximum embedding cosine similarity to any ground-truth predicate exceeds a threshold. This refinement filters out 23.19% and 13.83% of initial outputs for VG150 and PSG, respectively, enhancing data quality.

As illustrated in Figure 6, while the original datasets feature dense object annotations, they are often characterized by severe relation sparsity. Our augmentation strategy effectively recovers plausible inter-object associations, substantially enhancing the connectivity and completeness of the resulting scene graphs. For instance, it supplements missing spatial relations, such as <*paper-merged.1, beside, tv.1*> and <*mouse.2, on, table-merged.1*> for PSG samples, while clarifying possessive dependencies like <*building.1, has, window.1*> and <*sign.1, attached to, pole.1*> for VG150 samples. These relations are frequently omitted in the original annotations, and our method effectively mitigates this sparsity by generating more plausible relations.

B.4 Qualitative Analysis of End-to-End SGG with SFT and RL

Comparative analysis in Figure 7 indicates that the SFT-only model is prone to hallucinations during dense object localization, exhibiting autoregressive degeneration such as redundant instances of *window*. This behavior triggers sequence length limits, leading to premature truncation of outputs and underscoring the limitations of SFT. In contrast, RL optimization effectively eliminates these repetitive loops, enabling precise localization of eight *window* instances. Notably, the outputs maintain strict cross-stage sequentiality: object localization adheres to the identified category order, while relation extraction follows the subject sequence. This structural consistency focuses the model's attention, ensuring complete and coherent reasoning, which significantly enhances recall performance.

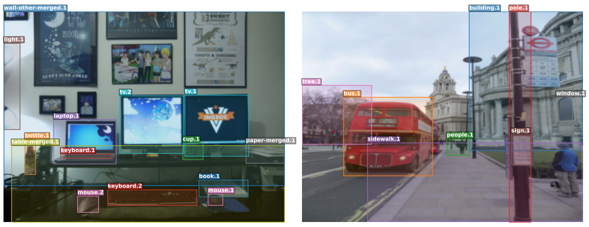
Object Detection Performance. The output of the SFT model demonstrates incomplete object detection results with low recall. Compared to the SFT model outputs, it is evident that for objects such as *sidewalk*, *building*, *pole*, and *sign*, the RL model can detect a greater number of reasonable instances of these categories, which is beneficial for improving ground truth coverage. Furthermore, the detection accuracy of the RL model is also enhanced, with fine-grained objects such as *sign*, *shirt*, and *short* all being well-detected. Despite the sparse annotation of objects in the original images, the model is capable of detecting objects like *window* even when they are unlabeled, demonstrating strong generalization.

Relation Recognition Performance. In terms of relation output, the SFT output suffers from truncation, leading to failed relation recognition. In contrast, the RL output is complete and includes all three types of relations. It can be observed that the model strictly adheres to recognizing and outputting these three types of relations, identifying relationships between objects from spatial, possessive, and interactive perspectives. Even with the original sparse relation annotations, our two-stage fine-tuning framework enables the model to output numerous reasonable relations, thereby improving recall and exhibiting strong generalization.

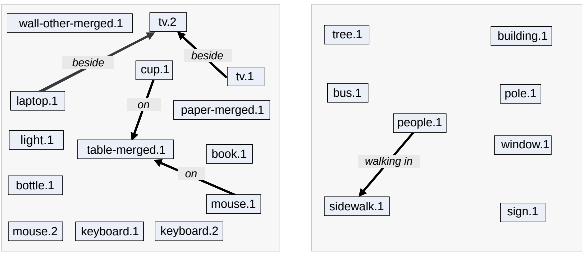
B.5 Qualitative Comparison of End-to-End SGG

Qualitative comparisons in Figure 8 demonstrate the efficacy of our method relative to the baseline method R1-SGG (Chen et al., 2025). Although R1-SGG produces scene graphs after two-stage post-training (SFT and RL), its outputs are notably shorter, resulting in lower overall recall of correct predictions. In contrast, our approach consistently generates structured three-stage outputs with reasoning trajectories that frequently surpass the ground truth in depth, leading to significantly higher overall recall.

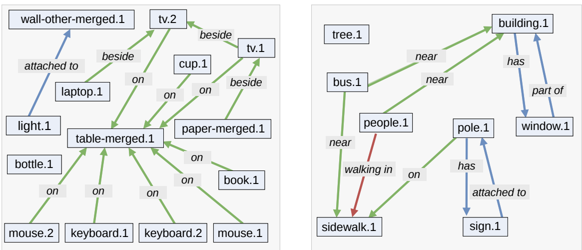
Object Detection Performance. Regarding ob-



(a) Ground-truth image samples with object bounding boxes for **PSG (left)** and **VG150 (right)**.



(b) Original sparse scene graphs, where rectangles denote objects and directed edges represent relations.



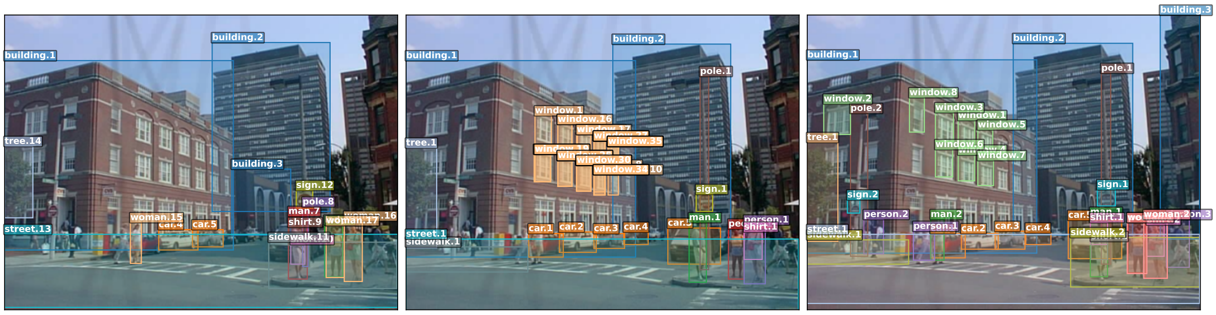
(c) Augmented scene graphs with color-coded edges: for PSG, green and blue edges signify *spatial* and *static-interactive* relations; for VG150, green, blue, and red edges represent *spatial*, *possessive*, and *interactive* relations, respectively.

Figure 6: Visualization of relation augmentation. (a) Original images with ground-truth object bounding boxes and category labels. (b) Visualization of original sparse relation annotations. (c) Visualization of the augmented relation annotations.

ject detection, it is evident that the baseline method exhibits restricted recall and frequently overlooks common objects such as *cup* and *shelf*, resulting in a sparse representation of the scene. In contrast, our model identifies a comprehensive range of plausible objects, including *handle*, *glass*, and *cup*, even when these instances are omitted from the original annotations, thereby significantly improving object detection recall. These results underscore the superior grounding capabilities of our framework.

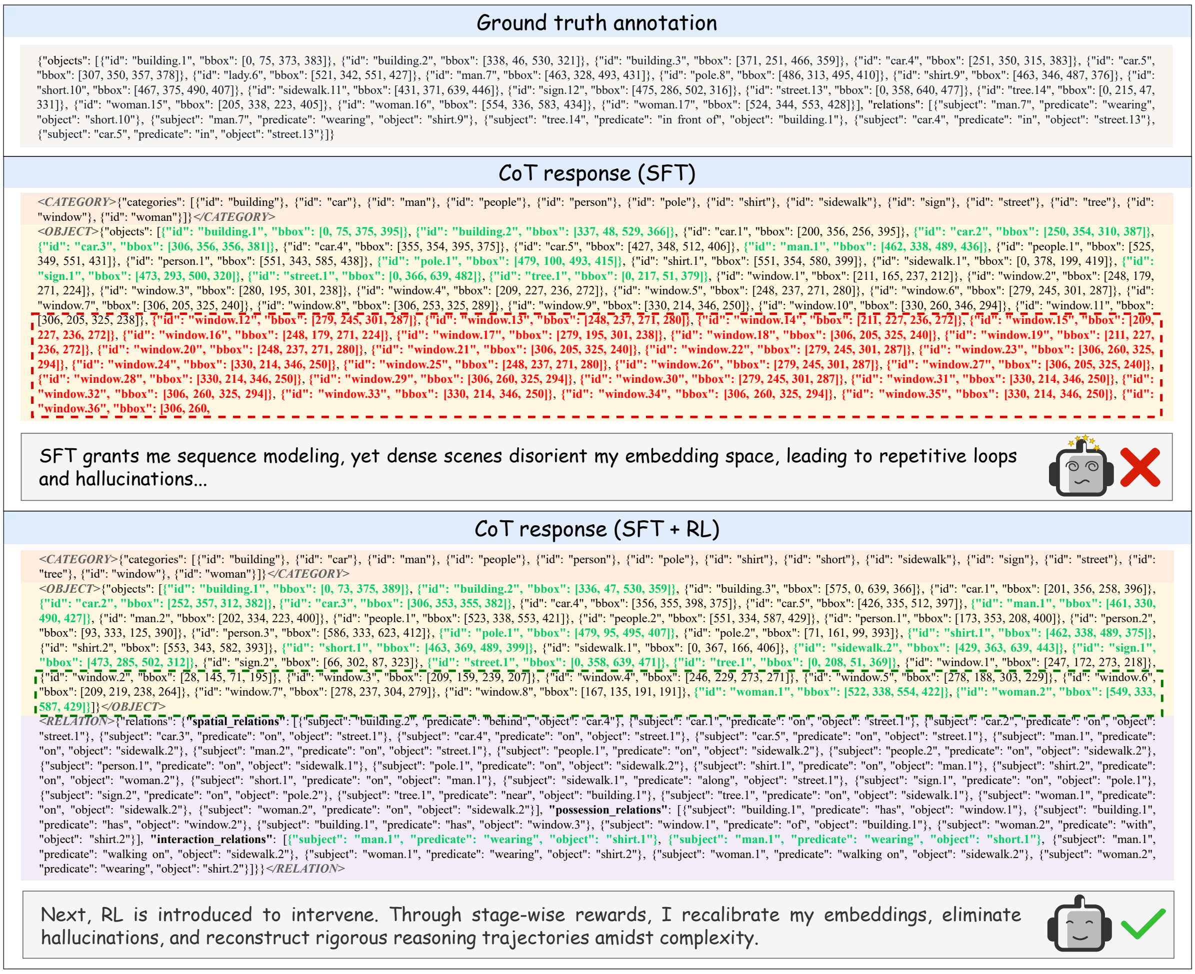
Relation Recognition Performance. The baseline yields sparse, often semantically trivial or erroneous relations, including unreasonable ones (e.g., <*cup.5, on, desk.6*> with misinterpreted referents) and incorrect predicates (e.g., <*draw.7, on, desk.6*>, <*book.2, on, desk.6*>). In contrast, our approach generates a higher density of valid triplets across three semantic levels-spatial (e.g., <*drawer.1, in, desk.1*>), possessive (e.g., <*drawer.1, has, handle.1*>), and complex interactions (e.g., <*person.1, wearing, glasses.1*>)-achieving superior ground-truth coverage and semantic depth, thereby mitigating relation bias. Moreover, the model detects plausible relations even without explicit localization (e.g., <*person.1, using, computer*> when the *computer* instance is undetected), demonstrating strong generalization.

Collectively, these results demonstrate that our three-stage reasoning framework effectively enhances the model's capacity to recognize fine-grained objects and capture intricate semantic dependencies, advancing the pursuit of robust scene understanding in real-world environments.



Ground Truth SFT SFT + RL

(a) Qualitative visualization of object detection. Results are shown for **ground truth (left)**, the **SFT only output (middle)**, and **output after SFT followed by RL (right)**.



(b) Comparison between SFT-only and SFT + RL outputs. Predictions are color-coded: green indicates alignment with ground truth, red denotes implausible repeated predictions.

Figure 7: Qualitative comparison of scene graph generation under different fine-tuning strategies. We visualize the ground truth, SFT-only outputs, and SFT + RL outputs. The comparisons highlight the performance gaps in both object detection (a) and relation recognition (b).

Ground Truth Baseline Method Our Method

(a) Qualitative visualization of object detection. Results are shown for **ground truth (left)**, the **baseline (middle)**, and **our method (right)**.

Ground truth annotation

{"objects": [{"id": "desk.1", "bbox": [12, 386, 565, 599]}, {"id": "man.2", "bbox": [111, 286, 377, 599]}, {"id": "shelf.3", "bbox": [0, 0, 222, 238]}, {"id": "shirt.4", "bbox": [111, 373, 335, 598]}, {"id": "desk.5", "bbox": [580, 428, 799, 592]}, {"id": "paper.6", "bbox": [582, 315, 763, 436]}, {"id": "chair.7", "bbox": [14, 478, 145, 599]}, {"id": "drawer.8", "bbox": [362, 493, 510, 582]}, {"id": "book.9", "bbox": [121, 39, 135, 133]}, {"id": "pant.10", "bbox": [188, 529, 364, 598]}, {"id": "drawer.11", "bbox": [603, 545, 789, 598]}, {"id": "shelf.12", "bbox": [0, 213, 207, 239]}, {"id": "paper.13", "bbox": [668, 289, 800, 338]}, {"id": "book.14", "bbox": [89, 158, 138, 217]}, {"id": "paper.15", "bbox": [475, 334, 531, 403]}, {"id": "hand.16", "bbox": [335, 402, 378, 436]}, {"id": "bag.17", "bbox": [702, 289, 778, 305]}, {"id": "cup.18", "bbox": [140, 166, 165, 216]}, {"id": "cup.19", "bbox": [444, 396, 496, 471]}], "relations": [{"subject": "cup.19", "predicate": "on", "object": "desk.1"}, {"subject": "man.2", "predicate": "wears", "object": "shirt.4"}, {"subject": "bag.17", "predicate": "on", "object": "paper.13"}, {"subject": "drawer.8", "predicate": "of", "object": "desk.1"}, {"subject": "cup.18", "predicate": "on", "object": "shelf.12"}, {"subject": "paper.13", "predicate": "on", "object": "desk.5"}, {"subject": "man.2", "predicate": "has", "object": "hand.16"}, {"subject": "book.9", "predicate": "on", "object": "shelf.12"}, {"subject": "book.14", "predicate": "on", "object": "shelf.12"}, {"subject": "man.2", "predicate": "wearing", "object": "pant.10"}, {"subject": "man.2", "predicate": "has", "object": "shirt.4"}, {"subject": "shelf.3", "predicate": "of", "object": "book.9"}]}

Answer from R1-SGG (without CoT)

direct prompting

Generate a structured scene graph for an image in JSON format, specifying objects with unique IDs (from a predefined set) and bounding boxes, and relationships with subjects, predicates (from a predefined set), and objects, following the provided guidelines and example.

unstructured reasoning

{"objects": [{"id": "book.1", "bbox": [107, 262, 173, 356]}, {"id": "book.2", "bbox": [6, 62, 186, 366]}, {"id": "chair.3", "bbox": [17, 795, 333, 996]}, {"id": "cup.4", "bbox": [560, 666, 622, 787]}, {"id": "cup.5", "bbox": [893, 783, 953, 878]}, {"id": "desk.6", "bbox": [13, 596, 710, 996]}, {"id": "drawer.7", "bbox": [457, 820, 637, 967]}, {"id": "hair.8", "bbox": [197, 473, 313, 633]}, {"id": "man.9", "bbox": [139, 473, 476, 996]}, {"id": "paper.10", "bbox": [727, 517, 956, 733]}, {"id": "shirt.11", "bbox": [139, 620, 426, 996]}, {"id": "window.12", "bbox": [281, 0, 737, 598]}, {"id": "person.13", "bbox": [139, 473, 476, 992]}], "relationships": [{"subject": "man.9", "predicate": "wearing", "object": "shirt.11"}, {"subject": "man.9", "predicate": "sitting on", "object": "chair.3"}, {"subject": "man.9", "predicate": "at", "object": "desk.6"}, {"subject": "man.9", "predicate": "has", "object": "hair.8"}, {"subject": "cup.4", "predicate": "on", "object": "desk.6"}, {"subject": "drawer.7", "predicate": "on", "object": "desk.6"}, {"subject": "book.2", "predicate": "on", "object": "book.1"}, {"subject": "person.13", "predicate": "at", "object": "desk.6"}, {"subject": "cup.5", "predicate": "on", "object": "desk.6"}, {"subject": "book.2", "predicate": "on", "object": "desk.6"}, {"subject": "man.9", "predicate": "on", "object": "chair.3"}, {"subject": "chair.3", "predicate": "at", "object": "desk.6"}, {"subject": "cup.4", "predicate": "above", "object": "desk.6"}, {"subject": "shirt.11", "predicate": "on", "object": "man.9"}, {"subject": "hair.8", "predicate": "on", "object": "man.9"}, {"subject": "man.9", "predicate": "using", "object": "desk.6"}, {"subject": "drawer.7", "predicate": "in", "object": "desk.6"}, {"subject": "book.1", "predicate": "on", "object": "book.2"}, {"subject": "desk.6", "predicate": "has", "object": "drawer.7"}, {"subject": "man.9", "predicate": "has", "object": "cup.4"}, {"subject": "cup.4", "predicate": "near", "object": "man.9"}, {"subject": "man.9", "predicate": "with", "object": "cup.4"}]}

Answer from SGG-R^3 (with CoT)

cot prompting

Execute a three-stage visual reasoning analysis for the provided image: Stage 1 detects unique object categories from predefined sets, outputting a JSON array within <CATEGORY> tags; Stage 2 localizes all instances of identified categories with sequential IDs and bounding boxes, outputting a JSON array within <OBJECT> tags; Stage 3 extracts spatial, possessive, and interactive relations between object pairs using predefined predicates, outputting a JSON array within <RELATION> tags. The process strictly follows predefined categories and sequential analysis requirements.

structured reasoning

<CATEGORY>{"categories": [{"id": "book"}, {"id": "chair"}, {"id": "cup"}, {"id": "desk"}, {"id": "drawer"}, {"id": "glass"}, {"id": "hair"}, {"id": "hand"}, {"id": "handle"}, {"id": "man"}, {"id": "paper"}, {"id": "shelf"}, {"id": "shirt"}, {"id": "window"}]}</CATEGORY>
<OBJECT>{"objects": [{"id": "book.1", "bbox": [0, 33, 151, 224]}, {"id": "book.2", "bbox": [0, 137, 91, 223]}, {"id": "chair.1", "bbox": [12, 464, 373, 587]}, {"id": "cup.1", "bbox": [453, 389, 502, 459]}, {"id": "cup.2", "bbox": [725, 458, 767, 514]}, {"id": "cup.3", "bbox": [139, 161, 168, 211]}, {"id": "desk.1", "bbox": [35, 388, 573, 587]}, {"id": "desk.2", "bbox": [584, 452, 798, 587]}, {"id": "drawer.1", "bbox": [373, 481, 512, 569]}, {"id": "drawer.2", "bbox": [45, 443, 362, 510]}, {"id": "glass.1", "bbox": [238, 315, 260, 334]}, {"id": "hair.1", "bbox": [161, 276, 248, 370]}, {"id": "hand.1", "bbox": [336, 390, 381, 430]}, {"id": "handle.1", "bbox": [419, 515, 463, 538]}, {"id": "handle.2", "bbox": [405, 566, 477, 587]}, {"id": "man.1", "bbox": [110, 276, 380, 587]}, {"id": "paper.1", "bbox": [587, 308, 773, 425]}, {"id": "paper.2", "bbox": [264, 256, 304, 315]}, {"id": "shelf.1", "bbox": [0, 0, 221, 238]}, {"id": "shelf.2", "bbox": [0, 119, 222, 143]}, {"id": "shirt.1", "bbox": [110, 365, 338, 587]}, {"id": "window.1", "bbox": [229, 0, 591, 321]}]}</OBJECT>
<RELATION>{"relations": {"spatial_relations": [{"subject": "book.1", "predicate": "on", "object": "shelf.1"}, {"subject": "book.1", "predicate": "on", "object": "shelf.2"}, {"subject": "book.2", "predicate": "on", "object": "shelf.2"}, {"subject": "chair.1", "predicate": "in front of", "object": "desk.1"}, {"subject": "cup.1", "predicate": "on", "object": "desk.1"}, {"subject": "cup.2", "predicate": "on", "object": "desk.2"}, {"subject": "cup.3", "predicate": "on", "object": "shelf.2"}, {"subject": "drawer.1", "predicate": "in", "object": "desk.1"}, {"subject": "drawer.2", "predicate": "in", "object": "desk.1"}, {"subject": "glass.1", "predicate": "on", "object": "man.1"}, {"subject": "hair.1", "predicate": "on", "object": "man.1"}, {"subject": "hand.1", "predicate": "on", "object": "desk.1"}, {"subject": "handle.1", "predicate": "on", "object": "drawer.1"}, {"subject": "handle.2", "predicate": "on", "object": "drawer.1"}, {"subject": "man.1", "predicate": "at", "object": "desk.1"}, {"subject": "paper.1", "predicate": "on", "object": "desk.2"}, {"subject": "paper.2", "predicate": "on", "object": "desk.1"}, {"subject": "shelf.1", "predicate": "behind", "object": "man.1"}, {"subject": "shirt.1", "predicate": "on", "object": "man.1"}, {"subject": "window.1", "predicate": "in front of", "object": "man.1"}], "possession_relations": [{"subject": "drawer.1", "predicate": "has", "object": "handle.1"}, {"subject": "drawer.2", "predicate": "of", "object": "desk.1"}, {"subject": "man.1", "predicate": "has", "object": "shirt.1"}, {"subject": "man.1", "predicate": "has", "object": "hair.1"}, {"subject": "man.1", "predicate": "has", "object": "hand.1"}, {"subject": "man.1", "predicate": "has", "object": "glass.1"}, {"subject": "man.1", "predicate": "with", "object": "hair.1"}, {"subject": "man.1", "predicate": "with", "object": "shirt.1"}, {"subject": "man.1", "predicate": "with", "object": "glass.1"}, {"subject": "shelf.1", "predicate": "with", "object": "book.2"}, {"subject": "shelf.2", "predicate": "with", "object": "book.2"}, {"subject": "shelf.2", "predicate": "with", "object": "book.1"}], "interaction_relations": [{"subject": "man.1", "predicate": "sitting on", "object": "chair.1"}, {"subject": "man.1", "predicate": "wearing", "object": "shirt.1"}, {"subject": "man.1", "predicate": "wearing", "object": "glass.1"}, {"subject": "man.1", "predicate": "using", "object": "computer"}, {"subject": "man.1", "predicate": "looking at", "object": "screen"}]}}</RELATION>

(b) Qualitative visualization of reasoning trajectories. Object instances and relation triplets are color-coded: green indicates alignment with ground truth, red denotes implausible predictions, and orange highlights plausible predictions for undetected objects.

Figure 8: Qualitative comparison of scene graph generation between our method and the baseline. We visualize the ground truth, baseline reasoning outputs, and our method's CoT-based reasoning results. The comparisons highlight the performance gaps in both object detection (a) and relation recognition (b).

CoT prompt for scene graph generation

[Task Definition]

Execute a structured, three-stage visual reasoning analysis for the provided image, strictly adhering to the predefined object and relation categories. Proceed through each stage sequentially and encapsulate your outputs using the specified tags.
Object Categories: {OBJ_CLS}
Relation Predicates: Spatial Relations: {SPATIAL_REL}; Possessive Relations: {POSSESSIVE_REL}; Interactive Relations: {INTERACTIVE_REL}

[Chain-of-Thought]

Stage 1: Object Category Detection
Task: Identify all unique object categories in the image from the provided "Object Categories".
Requirements: Only output categories that are clearly visible and identifiable in the image. The categories must conform to the predefined object categories. Maintain uniqueness (no duplicates). Output Format: Output as a JSON array within <CATEGORY> and </CATEGORY> tags.
Stage 2: Object Instance Grounding
Task: Detect and localize every individual instance of the categories identified in Stage 1 sequentially.
Requirements: Process categories in the order identified in Stage 1. The object names must strictly correspond to the categories identified in Stage 1. Assign sequential instance numbers to objects within each category (e.g., man.1, man.2, car.1). Provide precise bounding boxes in [x1, y1, x2, y2] format (integer coordinates). All categories listed in Stage 1 must have instances detected in Stage 2. Output Format: Output as a JSON array within <OBJECT> and </OBJECT> tags.
Stage 3: Multi-type Relation Extraction
Task: Independently analyze relations in three predicate types, examining all object pairs within each category. Three Relation Categories: 1. Spatial Relations: Analyze all object pairs for spatial/topological relations. 2. Possessive Relations: Analyze all object pairs for ownership, composition, and part-whole relations. 3. Interactive Relations: Analyze all object pairs for action-oriented and functional relations.
Requirements: Sequentially analyze each object in Stage 2 as the subject against all other objects in the scene to extract meaningful relations. All relations must be between objects localized in Stage 2. All relation predicates MUST belong to their respective predefined category. Output Format: Output as a JSON array within <RELATION> and </RELATION> tags.

[In-Context Example]

Complete Output Example:
<CATEGORY>{"categories": [{"id": "tree"}, {"id": "sidewalk"}, {"id": "building"}, {"id": "man"}, {"id": "bike"}]}</CATEGORY>
<OBJECT>{"objects": [{"id": "tree.1", "bbox": [0, 0, 799, 557]}, {"id": "sidewalk.1", "bbox": [75, 306, 798, 596]}, {"id": "building.1", "bbox": [0, 0, 222, 538]}, {"id": "man.1", "bbox": [369, 262, 446, 512]}, {"id": "man.2", "bbox": [236, 246, 296, 508]}, {"id": "bike.1", "bbox": [335, 317, 362, 353]}]}</OBJECT>
<RELATION>{"relations": {"spatial_relations": [{"subject": "tree.1", "predicate": "beside", "object": "sidewalk.1"}, {"subject": "building.1", "predicate": "near", "object": "tree.1"}], "possession_relations": [{"subject": "man.1", "predicate": "has", "object": "bike.1"}], "interaction_relations": [{"subject": "man.1", "predicate": "standing on", "object": "sidewalk.1"}, {"subject": "man.2", "predicate": "walking on", "object": "sidewalk.1"}, {"subject": "man.2", "predicate": "talking to", "object": "man.1"}, {"subject": "bike.1", "predicate": "parked on", "object": "sidewalk.1"}]}}</RELATION>

Generate the three-stage analysis for the image:

Figure 9: Three-stage CoT prompt template for end-to-end scene graph generation. The template is applied to both VG150 and PSG datasets, with the predefined object classes and predicate categories adjusted accordingly for each benchmark.

CoT prompt for relation augmentation

[Task Definition]

Execute a structured, two-stage relation augmentation analysis for the provided image with pre-annotated objects. Your goal is to generate plausible relations.
Relation Predicates: Spatial Relations: {SPATIAL_REL}; Possessive Relations: {POSSESSIVE_REL}; Interactive Relations: {INTERACTIVE_REL}
Pre-annotated Objects ({OBJ_COUNT} objects, USE EXACT IDs): {OBJ_LIST}
Each object includes: id: Unique identifier (e.g., "person.1", "car.1"); box: Bounding box coordinates [x1, y1, x2, y2] representing object position

[Chain-of-Thought]

Stage 1: Scene Context Understanding
Task: Create a comprehensive natural language description for the image that incorporates ALL pre-annotated objects with their exact IDs.
Requirements: Naturally include all {OBJ_COUNT} objects using their specific IDs. Describe spatial arrangements, potential interactions, and scene context. Be descriptive but concise.
Stage 2: Multi-type Relation Extraction
Task: Based on the global understanding from Stage 1, independently analyze relations in three predicate types, examining all object pairs within each category. Three Relation Categories: 1. Spatial Relations: Analyze object pairs for spatial/topological relations. 2. Possessive Relations: Analyze for ownership/composition/part-whole relations. 3. Interactive Relations: Analyze for action-oriented/functional relations.
Requirements: Sequentially analyze each object in Stage 2 as the subject against all other objects in the scene to extract meaningful relations. All relations must be between objects in pre-annotated objects. All relation predicates MUST belong to their respective predefined category.

[In-Context Example]

Return ONLY a JSON object with this exact structure:
{"caption": "A detailed scene description naturally incorporating all object IDs like person.1, car.1, etc.", "spatial_relations": [{"subject": "object.number", "predicate": "SPATIAL_PREDICATE", "object": "object.number"}...], "possession_relations": [{"subject": "object.number", "predicate": "POSSESSIVE_PREDICATE", "object": "object.number"}...], "interaction_relations": [{"subject": "object.number", "predicate": "INTERACTIVE_PREDICATE", "object": "object.number"}...]}}

Generate the two-stage relation augmentation analysis, focusing on creating meaningful relationships:

Figure 10: Two-stage CoT prompt template for relation augmentation. The template is applied to both VG150 and PSG datasets, with the predefined object list and predicate categories adjusted accordingly for each benchmark.