TOWARDS GENERALIZED SCENE GRAPH GENERATION

ZUYAO CHEN

PhD

The Hong Kong Polytechnic University

2026

The Hong Kong Polytechnic University
Department of Computing

Towards Generalized Scene Graph Generation

Zuyao Chen

A thesis submitted in partial fulfillment of the requirements for
the degree of Doctor of Philosophy
November 2025

CERTIFICATE OF ORIGINALITY

I hereby declare that this thesis is my own work and that, to the best of my knowledge and belief, it reproduces no material previously published or written, nor material that has been accepted for the award of any other degree or diploma, except where due acknowledgment has been made in the text.

Signature: _____

Name of Student: Zuyao Chen

Abstract

Scene Graph Generation (SGG) is a crucial task in computer vision that seeks to represent images as structured graphs, capturing objects and their semantic relationships. While existing SGG methods have made remarkable progress, they often suffer from three key limitations: closed set assumptions, reliance on costly manual annotations, and a lack of alignment between training objectives and structured output formats. This thesis aims to overcome these limitations by exploring generalized SGG across four dimensions: open-vocabulary recognition, language-based supervision, reinforcement learning for structured prediction, and scene graph-guided image generation.

We first propose a unified transformer-based framework, *OvSGTR*, for Fully Open-Vocabulary SGG, enabling the recognition of novel objects and relations through visual-language alignment and retention. We then introduce a weakly-supervised paradigm, *GPT4SGG*, that reverses the traditional weakly-SGG pipeline by leveraging large language models (LLMs) to synthesize scene graphs from region and holistic descriptions. To further align model optimization with structured evaluation, we develop a reinforcement learning framework *R1-SGG*, which employs rule-based rewards for fine-grained and format-consistent scene graph output. Finally, we explore the inverse task of using scene graphs as controllable interfaces for image generation, and present a novel benchmark, *Scene-Bench*, with a new metric, *SGScore*, for evaluating object and relation fidelity in generated images.

Extensive experiments conducted on widely used public datasets such as Visual Genome, GQA, and PSG validate the effectiveness and generalizability of the proposed approaches across a diverse range of evaluation scenarios. These include traditional closed-set settings, where object and relation categories are predefined; open-vocabulary and zero-shot setups, which test the model's ability to recognize and reason about unseen concepts; and generative tasks, such as image synthesis guided by structured scene representations. The proposed methods exhibit consistent improvements in structural accuracy, semantic coverage, and compositional generalization under these varied conditions. Collectively, the models, algorithms, and benchmarks introduced in this thesis contribute to the development of scalable, interpretable, and robust scene understanding systems that are capable of operating effectively in open-world environments, where visual content is complex, diverse, and continually evolving.

Publications Arising from the Thesis

1. Zuyao Chen, Jinlin Wu, Zhen Lei, Zhaoxiang Zhang, and Chang Wen Chen, "Expanding scene graph boundaries: Fully open-vocabulary scene graph generation via visual-concept alignment and retention", in *European Conference on Computer Vision (ECCV)*, pp. 108–124, 2024. **(Best Paper Nomination)**

2. Zuyao Chen, Jinlin Wu, Zhen Lei, Zhaoxiang Zhang, and Chang Wen Chen, " GPT4SGG: Synthesizing scene graphs from holistic and region-specific narratives", *arXiv preprint arXiv:2312.04314*, 2023.

3. Zuyao Chen, Jinlin Wu, Zhen Lei, and Chang Wen Chen, "What makes a scene? Scene graph-based evaluation and feedback for controllable generation", *arXiv preprint arXiv:2411.15435*, 2024.

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

5. Zuyao Chen, Jinlin Wu, Zhen Lei, and Chang Wen Chen, "From data to modeling: Fully open-vocabulary scene graph generation", *arXiv preprint arXiv: 2505.20106*, 2025.

Acknowledgments

Every day is a good day! When I took the train to ETH Zürich, the Alps looked so vivid and splendid—it was an aha moment in my Ph.D. journey. While standing on Mount Rigi, I spotted the name "Bristen" on a viewpoint display board—the same name as a compute node at the Swiss National Supercomputing Centre that I used to train models. Now, I have come to realize that my Ph.D. journey—despite its tough and confusing moments—has reached a peak, much like conquering the Swiss mountains. This journey, marked by the pursuit of expertise and the creation of new knowledge, began with a short but meaningful meeting with my supervisor, Prof. Chang Wen Chen, who later provided a supportive and independent research environment in PolyU.

I would like to express my deepest gratitude to him, who set a simple yet demanding standard: pursue *original* research. Through both regular group meetings and personal discussions, I came to understand that originality means seeking the essential differences between our work and prior state-of-the-arts. I also learned that diving into details without a guiding hypothesis or plan is risky, whereas clear direction and critical reflection are indispensable for meaningful research. Beyond this, Prof. Chen always motivated us to think more deeply, to cultivate a broader research vision, and to maintain a warm heart toward the research community. These characteristics and lessons have profoundly shaped my growth as a Ph.D. student.

I would also like to thank the external examiners of my Ph.D. defense, Prof. HE

Zhihai and Prof. ZHANG Hanwang, for their insightful comments and constructive suggestions, which helped improve the quality and clarity of this thesis.

I would like to thank my collaborators—Dr. Jinlin Wu, Prof. Zhen Lei, and Prof. Marc Pollefeys—for their valuable contributions to this thesis. Dr. Wu and I frequently exchanged ideas through in-depth discussions and brainstorming sessions, which sparked many new research directions. I am grateful to Prof. Pollefeys for his insightful feedback and generous provision of computational resources, which empowered me to pursue more ambitious work.

I would also like to thank my friends, Mr. Zhida Zhou, Dr. Jinlin Wu, Dr. Tao Hu, Dr. Yang Bai, Mr. Ye Liu, Mr. Guanchen Ding, and all the other friends at Hong Kong PolyU. Their companionship helped me get through the tough and confusing moments of this journey.

Finally, I would like to dedicate this thesis to my father, my late mother, my brothers and sisters, and my beloved partner. The happiness and warmth they bring have always motivated me to explore the world. I would also like to thank myself for having the courage to embark on this challenging journey and for embracing the opportunity to meet people from all walks of life.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Scene Graph Generation

Scene Graph Generation (SGG) is a foundational task in visual scene understanding, aimed at representing visual information (*e.g.*, image, video, or 3D point clouds, *etc.*) as a graph structure where nodes denote localized objects and edges represent the semantic relationships between object pairs. Formally, given an input image I (as shown in Figure 1.1), the objective is to construct a graph $G = (V, E)$, where V is the set of objects and E is the set of directed edges indicating pairwise relationships. Each node is associated with a bounding box and a category label, while each edge corresponds to a predicate label describing the interaction or spatial configuration between two objects.

SGG extends beyond traditional object detection by incorporating relational reasoning, enabling a more holistic understanding of visual content. This structured representation facilitates a variety of downstream tasks, including image retrieval [52, 144], caption generation [141, 11, 37, 125, 92], visual question answering (VQA) [119, 95, 55, 63], and controllable image synthesis [51, 139], *etc.*. As such, SGG serves as an interpretable, symbolic abstraction layer that bridges low-level perception with

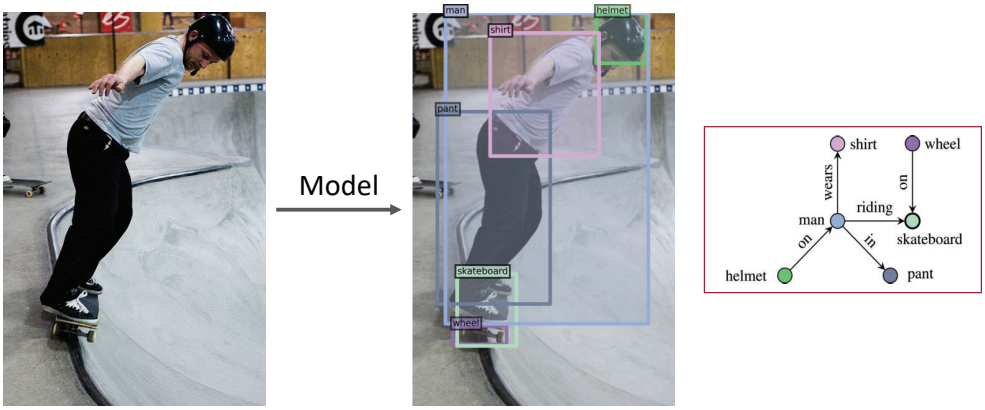


Figure 1.1: Illustration of Scene Graph Generation (SGG). Given an input image, objects are detected and localized with bounding boxes, and semantic relationships between object pairs are predicted. The resulting scene graph represents the image as a structured graph where nodes denote objects and edges denote relationships, providing an interpretable representation of the visual scene.

high-level reasoning.

Despite its potential, SGG remains a challenging problem. Early approaches focused on contextual feature aggregation and message passing mechanisms to enhance relational inference [135, 146], often relying on fixed label sets and large-scale annotated datasets such as Visual Genome [59]. However, these models exhibit strong biases toward frequent object-relation triplets, struggle with long-tail distributions, and operate under closed-set assumptions—where only a limited, predefined vocabulary of categories is used during both training and inference.

In real-world applications, visual scenes are inherently open-ended and compositional, comprising a rich and diverse set of entities and interactions not covered by standard benchmarks. To enable generalization, recent studies have explored open-vocabulary SGG [40, 150], where models are expected to detect and relate unseen object and predicate categories by leveraging vision-language alignment, zero-shot learning, or

external knowledge. These methods aim to move beyond rigid classification and toward more flexible and scalable frameworks.

Furthermore, acquiring high-quality scene graph annotations is labor intensive. Weakly-supervised and language-supervised alternatives attempt to alleviate this bottleneck by learning from image-caption pairs or synthetic data generated by large language models (LLMs). However, grounding ambiguity, sparsity, and noise in caption data present new challenges in generating accurate and complete scene graphs.

Another critical aspect of SGG is structured training and evaluation using LLMs. Traditional cross-entropy-based objectives are ill suited to a graph-structured output space. Thus, reinforcement learning (RL) techniques have been proposed to align model training with evaluation goals such as triplet recall, structural validity, and semantic fidelity.

Finally, SGG is increasingly explored not only as an end task but also as a controllable interface for generation. Scene-graph-to-image models use SGG outputs to guide high-fidelity image synthesis, opening a new frontier in structured generative modeling. This raises additional questions about consistency, interpretability, and feedback in the generation loop.

This thesis addresses these limitations and opportunities by proposing novel architectures, training paradigms, and benchmarks to advance the field of generalized Scene Graph Generation - where models are capable of recognizing novel entities, learning from limited supervision, reasoning over complex structures, and supporting downstream generation tasks.

1.2 Research Questions

The central goal of this thesis is to advance Scene Graph Generation (SGG) towards generalized and scalable systems that operate effectively in real-world, open-

vocabulary settings. To this end, we formulate the following key research questions that guide our investigation:

- **Q1: How can we design an SGG model that generalizes to unseen object and relation categories?**
  Traditional SGG models rely on fixed vocabularies and closed-set assumptions, limiting their applicability in open-world settings. This question examines strategies—such as transformer-based architectures, visual-language alignment, and pretraining pipelines—that enable SGG models to recognize novel concepts not seen during training.

- **Q2: Can scene graphs be synthesized from weak supervision, such as natural language descriptions, instead of costly manual annotations?**
  Manual scene graph annotation is expensive and time-consuming. This question explores whether language models can be used to synthesize scene graphs from image captions or region-level narratives, thereby enabling weakly-supervised learning of relational structures.

- **Q3: How can reinforcement learning be applied to optimize structured scene graph outputs?**
  Existing training paradigms of LLMs often neglect the structured nature of scene graphs. We investigate how reinforcement learning (RL) with rule-based rewards can improve model outputs by optimizing for structural accuracy, recall, and format consistency.

- **Q4: How effective are scene graphs as controllable interfaces for image generation?**
  While text-to-image models have achieved impressive results, they often struggle with complex compositional control. This question evaluates whether scene graphs can serve as interpretable, structured conditions for generating images, and how feedback mechanisms can improve image fidelity.

- **Q5: What are effective metrics and benchmarks for evaluating scene graph-based image generation?**
  Standard metrics may fail to capture the alignment between generated images and intended scene structures. This question addresses how to quantify object and relation accuracy in generated images using large vision-language models and proposes new benchmarks like *Scene-Bench*.

1.3 Thesis Contributions

This thesis makes the following key contributions toward generalized scene graph generation:

- **Fully Open-Vocabulary Scene Graph Generation.** We present a unified transformer-based framework, *OvSGTR*, capable of handling novel object and relation categories through visual-concept alignment and retention. We introduce four distinctive SGG settings: Closed-set, OvD-SGG, OvR-SGG, and OvD+R-SGG, and provide systematic evaluations across all scenarios, spanning both data and modeling perspectives.
- **Weakly-Supervised Scene Graph Learning.** We propose *GPT4SGG*, a language-only LLM pipeline that synthesizes scene graphs from holistic and region-specific narratives, and *MegaSG*, a large-scale scene graph dataset generated by multimodal LLMs. These contributions significantly reduce the need for costly manual annotations.
- **Reinforcement Learning with Structured Rewards.** We design a reinforcement learning framework *R1-SGG* to optimize M-LLM in scene graph generation tasks. Using rule-based rewards such as hard / soft recall and format consistency, we demonstrate improved relational accuracy and structural validity.

- **Scene Graph-based Image Generation and Evaluation.** We introduce *Scene-Bench*, a new benchmark to evaluate the factual consistency of image generation from scene graphs. A novel metric, *SGScore*, is developed to quantify the fidelity of objects and relationships using multimodal LLM. Furthermore, we propose a scene graph feedback mechanism that iteratively refines generation quality.

- **Extensive Empirical Validation.** We conduct thorough experiments across multiple datasets, including VG150, COCO, GQA200, PSG, and MegaSG. Our methods consistently outperform state-of-the-art baselines in both scene graph prediction and image generation tasks.

1.4 Thesis Organization

The remainder of this thesis is organized as follows, with Figure 1.2 providing a schematic overview of the relationships among the main technical chapters.

- **Chapter 2** reviews related work in Scene Graph Generation (SGG), including SGG approaches, open-vocabulary vision tasks, vision-language pretraining, reinforcement learning for LLMs, and text-/SG-to-image generation benchmarks.

- **Chapter 3** introduces a unified transformer-based framework, OvSGTR, for fully open-vocabulary scene graph generation. It defines four novel SGG settings and presents pretraining strategies leveraging scene parsers, LLMs, and multimodal LLMs.

- **Chapter 4** investigates weakly-supervised SGG from language supervision. It presents GPT4SGG, a language-only LLM pipeline, and MegaSG, a large-scale dataset synthesized by multimodal LLMs, along with comprehensive experiments.

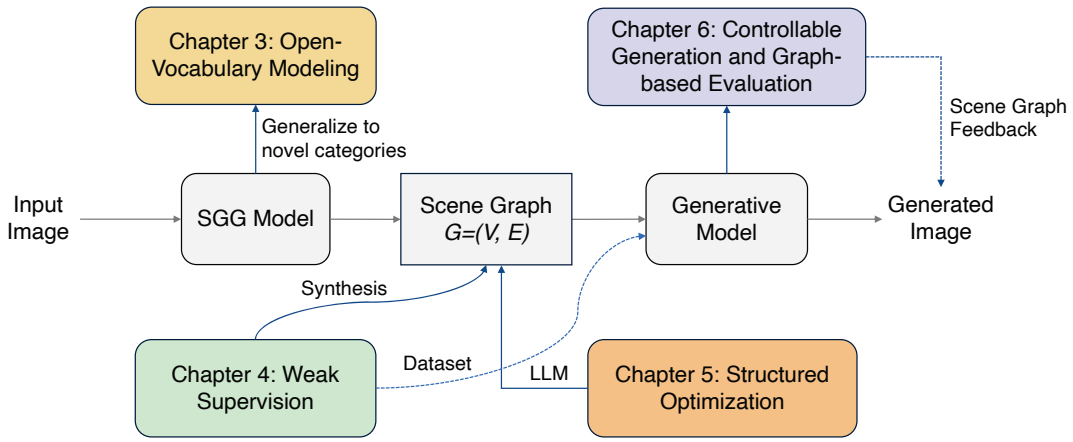


Figure 1.2: The logical structure of this thesis. Chapter 4 develops scalable supervision and data resources that support the modeling study in Chapter 3, the reinforcement learning framework in Chapter 5, and the benchmark construction in Chapter 6. Chapters 3 and 5 improve scene graph generation from complementary modeling and optimization perspectives, while Chapter 6 uses scene graphs as controllable interfaces for image generation and graph-based evaluation.

- **Chapter 5** proposes a reinforcement learning framework for enhancing M-LLMs in SGG tasks. It formulates a set of structured reward functions and introduces R1-SGG, improving scene graph consistency and relational accuracy.
- **Chapter 6** studies the inverse task: generating images from scene graphs. It proposes Scene-Bench, a new benchmark for evaluating image fidelity and factual consistency, and introduces SGScore for graph-based evaluation and feedback.
- **Chapter 7** concludes the thesis by summarizing key findings and outlining future research directions in generalized scene graph generation and multimodal reasoning.

Chapter 2

Related Work

2.1 Scene Graph Generation

Scene Graph Generation (SGG) aims to construct structured graphs that localize objects and describe the semantic relationships between them. Early works primarily focus on *contextual information aggregation* to enhance relational reasoning, employing techniques such as message passing [135, 74, 146, 118, 13, 70, 56, 23]. To address the severe long-tail distribution of relationships, a parallel line of research explores *unbiased learning* strategies that improve generalization on rare triplets [117, 20, 72, 115, 50, 49].

Most of these approaches assume a fixed vocabulary of object and relation categories, typically relying on closed-set detectors such as Faster R-CNN [105]. This constraint limits their applicability in real-world scenarios, where models must often recognize novel or rare categories unseen during training.

Recent works have attempted to relax the closed-world assumption by extending SGG to the open-vocabulary setting from an object-centric perspective [40, 150]. While promising, these methods still fall short in handling unseen *relations* or the co-

occurrence of novel objects and relations, motivating the need for more generalized SGG frameworks. In Chapter 3, we introduce a unified framework *OvSGTR* for addressing such open-vocabulary scenarios.

To reduce the reliance on costly manual annotations, recent studies have explored *weakly-supervised Scene Graph Generation* using natural language processing (NLP) techniques to extract relational knowledge from image-caption pairs. Notable examples include the work of Zhong *et al.* [152] and Zhang *et al.* [150], which utilize captions to supervise scene graph learning without access to ground-truth triplets.

However, these methods face significant limitations. Captions are often biased, sparse, and tend to describe only salient objects or interactions, leading to incomplete supervision. Moreover, grounding textual entities to visual regions is inherently ambiguous, which further hinders the generation of accurate and complete scene graphs [17].

To address these challenges, our work *GPT4SGG* (see Chapter 4) proposes a new paradigm that inverts the conventional pipeline: instead of extracting graphs from captions, we synthesize *global and region-specific narratives* from grounded object regions and then infer relationships using large language models (LLMs). This approach substantially enriches the semantic content of scene graphs while mitigating grounding ambiguity and caption sparsity.

2.2 Open-Vocabulary Vision Tasks

Open-vocabulary Vision tasks aim to generalize beyond a fixed set of predefined categories by enabling models to recognize novel or previously unseen classes at inference time. A representative example is *Open-vocabulary Object Detection (OvD)*, which eliminates the restriction to a closed label set, such as the 80 categories in COCO [14]. To this end, Ov-RCNN [145] leverages weak language supervision by transferring semantic knowledge from image captions into the object detection pipeline. Critically,

supervision signals for novel classes are withheld during training, even though those categories may appear in the textual supervision.

Beyond direct caption grounding, ViLD [38] further improves generalization by distilling knowledge from a large-scale open-vocabulary image classifier into a two-stage detector. This paradigm has since been extended to other vision tasks, including open-vocabulary segmentation [32], and open-vocabulary video understanding [127].

In the context of structured prediction, recent efforts have explored *open-vocabulary scene graph generation (OvSGG)* [40, 150, 71], where the goal is to recognize unseen objects or relationships during test time. However, most prior work focuses on object generalization alone, leaving open-vocabulary relation modeling an underexplored challenge.

For a broader overview of open-vocabulary learning, we refer readers to recent surveys [132, 154].

2.3 Vision-Language Pretraining

Vision-Language Pretraining (VLP) has recently attracted considerable attention in a wide range of vision-language tasks. The primary objective of VLP is to establish an alignment between visual and linguistic semantic spaces. For example, CLIP [101] demonstrates impressive zero-shot image classification capabilities through contrastive learning on large-scale image-text datasets. Building upon this success, several subsequent methods [69, 86, 153] have focused on learning fine-grained alignments between image regions and language data, thereby enabling object detectors to recognize unseen objects by leveraging linguistic information. This success on downstream tasks exemplifies the potential of aligning visual features with relational concepts, which is a fundamental step towards developing a fully open-vocabulary scene graph generation framework.

2.4 Reinforcement Learning for LLMs

Reinforcement learning (RL) has been increasingly adopted to enhance the reasoning capabilities of large models. Algorithms like Proximal Policy Optimization (PPO) [110] and Group Relative Policy Optimization (GRPO) [112] guide models using reward signals instead of relying solely on maximum likelihood estimation. In the context of large language models, DeepSeek-R1 [39] demonstrates that reinforcement learning (RL) can significantly enhance structured reasoning and planning by leveraging rule-based rewards instead of explicit reward models.

In multimodal learning, however, RL remains underutilized for generating structured outputs. Our work *R1-SGG* (see Chapter 5) addresses this by introducing rule-based reward functions at multiple levels, including three scene graph reward variants and a format consistency reward. These signals promote the generation of meaningful andcoherent scene graphs by explicitly evaluating alignment with ground-truths.

2.5 Controllable Image Generation

Text-to-Image Generation (T2I). The field of text-to-image generation has seen significant advancements with the transition from GANs [35] to diffusion models. Early GAN-based methods like StackGAN [148] and AttnGAN [137] generated images from textual descriptions but often struggled with image quality and diversity. The introduction of diffusion models marked a substantial improvement. Models such as DALL-E [102], GLIDE [94], and Stable Diffusion [107] have achieved high-quality image synthesis with better text-image alignment by iteratively refining images from noise, conditioned on text prompts. Despite their success, these models face challenges in generating complex scenes involving multiple objects and ensuring consistency in object relationships.

Scene Graph-to-Image Generation (SG2IM). Scene graphs offer a structured represen-

tation of objects and their relationships, providing a promising scheme for controllable image generation. Johnson *et al.* [51] introduced SG2Im, a model that generates images from scene graphs using graph convolutional networks (GCN) and conditional GANs. Ashual and Wolf [2] extended this approach by incorporating more detailed scene representations and object attributes. Recent methods have integrated scene graphs with diffusion models to enhance compositionality [85, 29, 113, 83, 131]. However, these approaches often require training on large-scale scene graph datasets and rely on additional guidance like bounding boxes(*e.g.*, [29]) or specialized graph encoders (*e.g.*, [139]), which cannot be adapted to open-vocabulary scenarios. More importantly, evaluating these models remains challenging due to the lack of metrics that effectively capture scene-level fidelity, including object presence and relationship accuracy in generated images.

LLMs in Image Generation. The integration of LLMs has opened new possibilities in image generation. Recent works [31, 134, 140, 75] have leveraged LLMs to enhance compositionality and controllability in image synthesis. For example, LayoutGPT [31] utilizes LLMs as visual planners to generate layouts from textual descriptions, improving user controllability. Similarly, methods like RPG [140] and Complex Diffusion [84] leverage the reasoning capabilities of LLMs to decompose complex prompts into simpler tasks, aiding in the generation of complex scenes with multiple objects and relationships. However, their potential for providing feedback to iteratively refine scene graph-based generation has not been fully explored.

Improving Relationship Consistency. Addressing the limitations in capturing object relationships, several approaches have been proposed. Feng *et al.* [30] focused on improving compositional generalization in diffusion models through a modulated cross-attention mechanism. Park *et al.* [99] introduced benchmarks specifically targeting compositional understanding in generative models. Chatterjee *et al.* [9] designed a benchmark to evaluate the capability of modeling spatial relationships. Despite these efforts, ensuring accurate depiction of relationships in complex scenes remains a sig-

nificant challenge, and existing methods often do not provide mechanisms for iterative refinement based on explicit relationship feedback.

Chapter 3

Fully Open-Vocabulary Scene Graph Generation

In this chapter, we discuss the topic of open-vocabulary scene graph generation (OvSGG). Unlike traditional closed-set settings, OvSGG requires models to recognize unseen object or relation categories.

3.1 Problem Definition

Scene Graph depicts visual objects and their inter-relationships in a scene. The study of Scene Graph Generation (SGG) has achieved significant advances, including standard benchmark datasets like Visual Genome [135], classical frameworks such as IMP [135], and Motifs [146]. Despite such achievements, current methods primarily work within a limited framework, restricting object and relationship categories to a predefined set. This constraint restricts the wider applicability of SGG models across various real-world applications.

This drawback of prevalent SGG models originates in two aspects. First, the complex

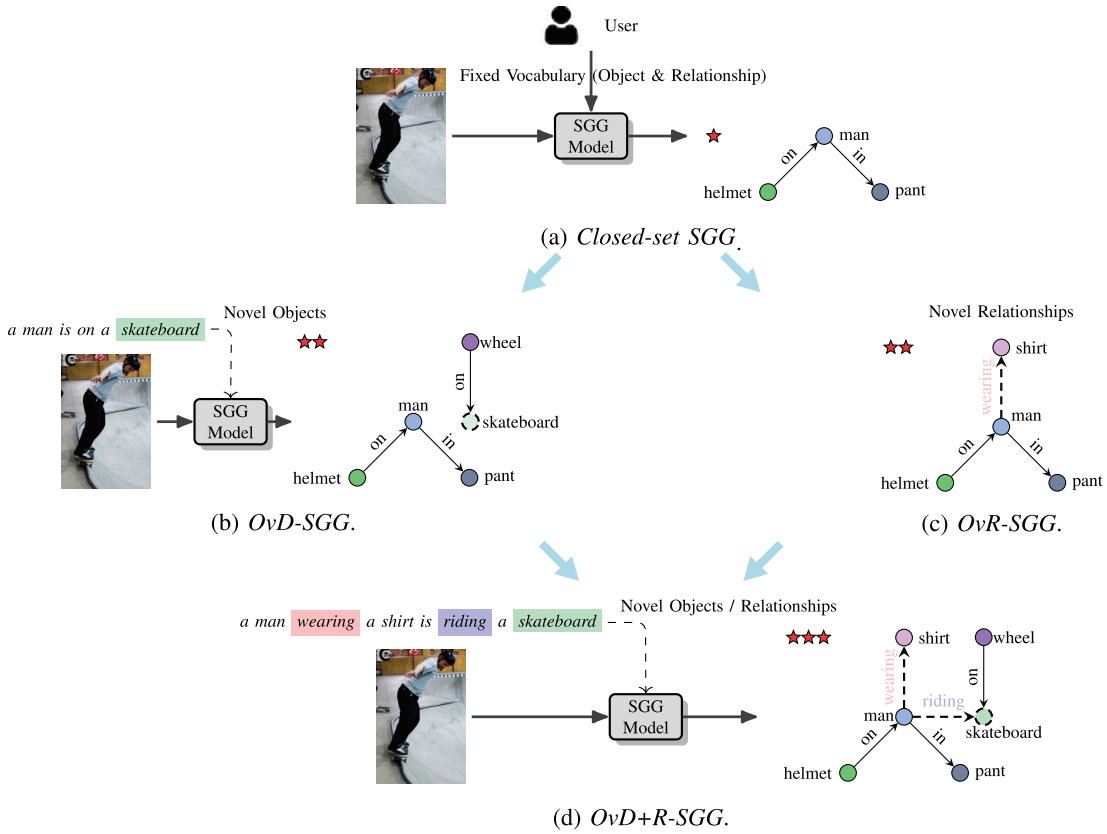


Figure 3.1: Illustration of SGG Scenarios (best view in color). Dashed nodes or edges in (a) - (d) refer to unseen category instances (during training), and stars refer to the difficulty of each setting. Previous works [135, 146, 118, 117, 20, 72, 12, 149] mainly focus on *Closed-set SGG* and few studies [40, 150] cover *OvD-SGG*. In this chapter, we give a more comprehensive study towards fully open vocabulary SGG.

and costly annotation of scene graphs leads to a shortage of large-scale scene graph datasets with a rich vocabulary. To alleviate this issue, some weakly-supervised methods [152, 73] utilize a language scene parser to obtain ungrounded relationship triplets as pseudo labels for learning scene graphs. However, as pointed out in *GPT4SGG* [19] (see Chapter 4), these methods suffer from inaccurate parsing results, ambiguous grounding of relationship triplets, and biased caption data. Second, prevalent SGG models treat object and relationship recognition as a multi-class classification problem, typically relying on a fixed classifier, such as a fully connected

layer, in such cases. Inspired by advancements in open-vocabulary object detection [145, 133, 69, 153, 26], recent studies [40, 150] have sought to augment the SGG task from a closed-set to an open-vocabulary domain. However, these efforts primarily adopt an object-centric perspective, focusing solely on scene graph nodes. A holistic approach to open-vocabulary SGG requires a thorough analysis of both nodes and edges. This raises two key questions that drive our research: ***Q1: Can the model predict unseen objects or relationships ? Q2: What happens when the model encounters both unseen objects and unseen relationships in a scene?***

To address the drawback, we compare different pipelines for relation-aware pretraining, where only weak supervision of scene graphs is provided. In this work, we evaluate three types of pipelines: (1) a *scene parser-based pipeline* [90], which leverages caption data and a natural language parser to extract ungrounded relationship triplets; (2) an *LLM (Large Language Model)-based pipeline*, such as *GPT4SGG* [19], which replaces the scene parser with a more powerful LLM to synthesize a dense scene graph; and (3) a *multimodal LLM-based pipeline*, which directly generates a dense scene graph with grounded objects. These pipelines enable the learning of transferable scene graph knowledge with minimal or no human annotations, significantly enhancing SGG model performance, particularly in open-vocabulary settings.

Building on our analysis of the limitations in current SGG models and the pipelines for relation-aware pretraining, we re-evaluate the conventional SGG settings and propose four distinct scenarios (see Figure 3.1):

1. ***Closed-set SGG***: Having been extensively studied in previous works [135, 146, 118, 117, 20, 72, 12, 149], this setting involves predicting nodes (*i.e.*, objects) and edges (*i.e.*, relationships) from a predefined set. Research in this area mainly focuses on feature aggregation and unbiased learning to mitigate long-tail distribution issues.

2. ***OvD-SGG*** (Open Vocabulary object Detection-based SGG): As explored in [150], this scenario extends *Closed-set SGG* from the object (node) perspective by enabling the recognition of unseen object categories during inference. However, the set of relationships remains fixed.

3. ***OvR-SGG*** (Open Vocabulary Relation-based SGG): This scenario introduces an open-vocabulary setting for relationships, requiring the model to predict unseen relation categories. This task is inherently more challenging due to the absence of pre-trained relation-aware models and reliance on less accurate scene graph annotations. Specifically, while *OvD-SGG* omits unseen object categories during training—yielding graphs with fewer nodes but correct edges—*OvR-SGG* excludes unseen relation categories, resulting in graphs with fewer edges. Consequently, the model must distinguish novel relationships from the "background".

4. ***OvD+R-SGG*** (Open Vocabulary Detection+Relation-based SGG): The most challenging scenario, where both unseen objects and unseen relationships are present, leads to sparser and less accurate training graphs. This setting demands robust detection and relation modeling under open-vocabulary conditions.

These distinct scenarios exhibit unique intrinsic characteristics and challenges, motivating further investigation into open-vocabulary scene graph generation.

With these distinct challenges in mind, we introduce *OvSGTR* (*Open-vocabulary Scene Graph Transformers*), a novel framework designed to tackle the complexities of open-vocabulary SGG. Our approach not only predicts unseen objects and relationships but also effectively handles the scenario in which both nodes and edges correspond to categories absent during training. Our framework consists of three main components: (1) a frozen image backbone for visual feature extraction, (2) a frozen text encoder for textual feature extraction, and (3) a transformer decoder for scene graph generation. During relation-aware pre-training, we explored three pipelines to generate scene graph supervision. In the fine-tuning phase, relation triplets with

location information (*i.e.*, bounding boxes) are sampled from manual annotations. These triplets are then associated with visual features, and visual-concept similarities are computed for both nodes and edges. The resulting similarities are used to predict object and relation categories, thereby enhancing the model's ability to generalize to unseen categories.

Furthermore, in our evaluation of relation-involved open-vocabulary SGG settings (*i.e.*, *OvR-SGG* and *OvD+R-SGG*), we empirically identified a catastrophic forgetting problem concerning relation categories. This forgetting degrades the model's ability to retain information learned from the relation-aware pre-training stage when it is subsequently exposed to fine-grained SGG annotations. To address this challenge, we propose a visual-concept retention mechanism coupled with a knowledge distillation strategy. In our approach, a pre-trained model serves as a teacher to guide the student model, ensuring that the rich semantic space of relations is preserved while adapting to new, fine-grained annotations. Simultaneously, the visual-concept retention mechanism helps maintain the model's proficiency in recognizing novel relations.

To our knowledge, *OvSGTR* is the first framework towards fully open-vocabulary SGG. We hope that this work will inspire a series of follow-up studies to further advance open-vocabulary scene graph generation. In summary, our key contributions are as follows:

- We conduct an in-depth study of open-vocabulary SGG from both node and edge perspectives, distinguishing four distinct settings—*Closed-set SGG*, *OvD-SGG*, *OvR-SGG*, and *OvD+R-SGG*. Our quantitative and qualitative analyses provide a holistic understanding of the unique challenges inherent to each scenario.

- We introduce a novel SGG framework, *OvSGTR*, where both objects (nodes) and relationships (edges) are extendable to unseen categories, greatly broadening the practical applicability of SGG models in real-world scenarios.

- By exploring three relation-aware pre-training pipelines, our approach significantly enriches the representation of relationships in open-vocabulary settings.

- Extensive experiments on the VG150 benchmark validate the effectiveness of our framework, demonstrating state-of-the-art performance across all evaluated scenarios.

3.2 Unified Transformer Framework – OvSGTR

Given an image I, the goal of the SGG task is to generate a descriptive graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where each node $v_i \in \mathcal{V}$ contains location (bounding box) and object category information, and each edge $e_{ij} \in \mathcal{E}$ represents the relationship between nodes v_i and v_j. In open-vocabulary settings, the label set $\mathcal{C}$ is divided into two disjoint subsets: *base classes* $\mathcal{C}_B$ and *novel classes* $\mathcal{C}_N$ ($\mathcal{C}_B \cup \mathcal{C}_N = \mathcal{C}$, $\mathcal{C}_B \cap \mathcal{C}_N = \emptyset$).

In this work, we unify node- and edge-level open-vocabulary scene graph generation within a single end-to-end transformer framework. Compared to our previous conference paper [17], we studied three relation-aware pre-training pipelines—namely, a scene parser-based, an LLM-based, and a multimodal LLM-based pipeline—to enrich visual relationship understanding. In addition, we expand our experimental evaluation by including comprehensive experiments on the GQA dataset, which demonstrate improved generalization in diverse, real-world scenarios.

3.2.1 Relation-aware Pre-training

The common paradigm in SGG is to use a pre-trained object detector (*e.g.*, Faster R-CNN) to provide a good prior for detecting objects, while this neglects the relationship between objects. To learn a transferable knowledge of visual relationships, three pipelines have been explored in this work, as shown in Figure 3.2:

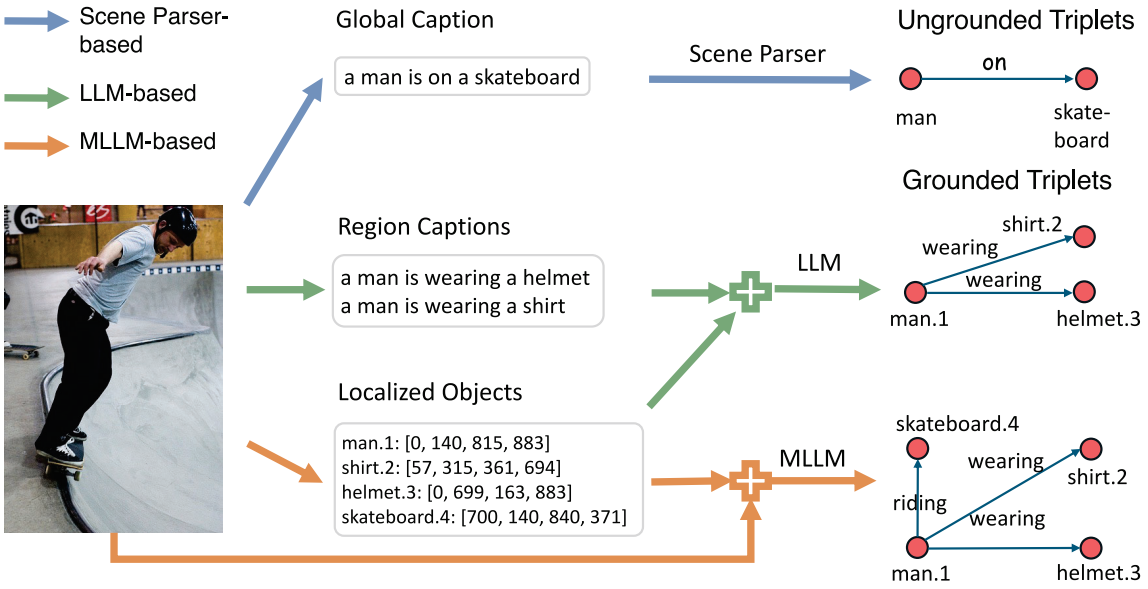


Figure 3.2: Comparison of different pipelines for relation-aware pre-training. (a) Early weakly-supervised methods [152, 73] utilize a language scene parser [90] to extract relationship triplets; (b) LLM-based methods replace the scene parser with a more powerful LLM to synthesize a more dense scene graph, *e.g.*, *GPT4SGG* [19]; (c) Multimodal LLM (MLLM)-based pipeline directly digests an input image and outputs a dense scene graph, *e.g.*, MegaSG [16].

(a) *Scene Parser-based Pipeline*: Learning from the caption as in previous methods [152, 73]. These approaches leverage a language parser to extract ungrounded triplets from captions. However, they face challenges such as inaccurate parsing, ambiguity in grounding triplets, and biased as well as sparse caption data (See *GPT4SGG*[19]; details in Chapter 4).

(b) *LLM-based Pipeline*: Learning from the LLM as in *GPT4SGG* [19]. This method adopts a two-stage pipeline: first, captioning dense regions, and then synthesizing the scene graph using a large language model (*e.g.*, GPT-4). The principal advantage of this pipeline is its powerful language understanding and reasoning capabilities, intrinsic to LLMs, which enable it to overcome limita-

tions (*e.g.*, inaccurate parsing) of conventional scene parser-based pipeline.

(c) *Multimodal LLM-based Pipeline*: Learning from the multimodal LLM as in MegaSG [16]. This approach directly processes an input image to generate a dense scene graph. A potential drawback is the limited spatial information (*i.e.*, bounding boxes) provided, as multimodal LLMs are less effective at localizing objects compared to dedicated object detectors. Nonetheless, unlike an LLM-based pipeline, this paradigm directly captures visual details, thereby enabling a more precise association between nodes and edges.

These pipelines aim to learn knowledge about visual relationships without or with fewer human annotations, significantly boosting the performance of SGG models under different settings.

3.2.2 Fully Open Vocabulary Architecture

As shown in Figure 3.3, *OvSGTR* is a DETR-like architecture composed of three main components: a visual encoder for image feature extraction, a text encoder for processing textual data, and a transformer that simultaneously performs object detection and relationship recognition. When provided with paired image-text data, *OvSGTR* adeptly generates the corresponding scene graphs. To reduce the optimization burden, the weights of both the image backbone and the text encoder are frozen during training.

Feature Extraction. Given an image–text pair, we extract multi-scale visual features using an image backbone (*e.g.*, Swin Transformer [87]) and obtain text features via a text encoder (*e.g.*, BERT [24]). The visual and text features are then fused and enhanced using cross-attention within the deformable encoder module of the transformer.

Prompt Construction. We construct the text prompt by concatenating all possible

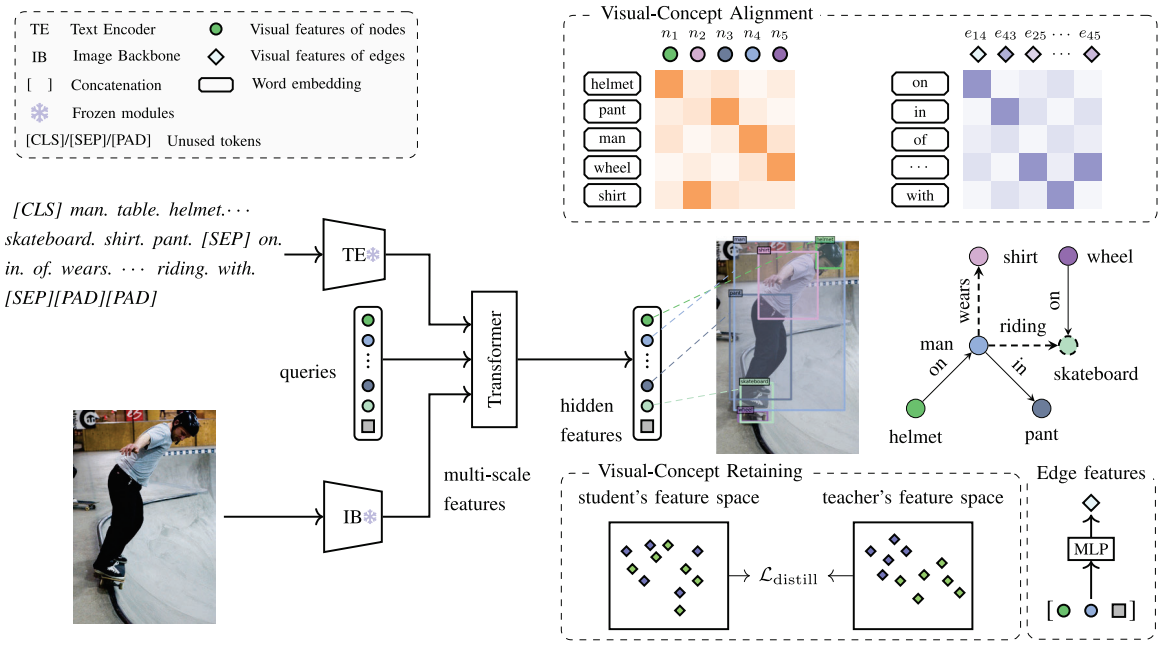


Figure 3.3: Overview of our proposed *OvSGTR*. The proposed *OvSGTR* is equipped with a frozen image backbone to extract visual features, a frozen text encoder to extract text features, and a transformer for decoding scene graphs. Visual features for nodes are the output hidden features of the transformer; Visual features for edges are obtained via a light-weight relation head (*i.e.*, with only two-layer MLP). Visual-concept alignment associates visual features of nodes/edges with corresponding text features. Visual-concept retention aims to transfer the teacher's capability of recognizing unseen categories to the student.

(or sampled) noun phrases and relation categories. For example, a prompt may be formatted as *[CLS] girl. umbrella. table. bathing. ⋯ zebra. [SEP] on. in. wears. ⋯ walking. [SEP][PAD][PAD]*, which is analogous to the strategy employed in GLIP [69] and Grounding DINO [86] that concatenate all noun phrases. For a large vocabulary during training, we randomly sample negative words and constrain the total number of positive and negative words to be M (*e.g.*, $M = 80$).

Node Representation. Given K object queries, the model follows the standard DETR framework to output K hidden features $\{\boldsymbol{v}_i\}_{i=1}^{K}$. Each feature is processed by:

- a bounding box head (a three-layer fully connected network) to decode location information (*i.e.*, a 4-dimensional vector), and

- a parameter-free classification head that computes the similarity between hidden features and text features.

These hidden features serve as the visual representations for the predicted nodes.

Edge Representation. Instead of using a complex message passing mechanism for relation features, we design a lightweight relation head that concatenates the node features for the subject and object with relation query features. To learn a relation-aware representation, we initialize a random embedding for querying relations. This embedding interacts with the image and text features via cross-attention in the decoder stage. Based on this design, for any subject–object pair (s_i, o_j), the edge representation is computed as

$$\boldsymbol{e}_{s_i \to o_j} = f_\theta\Big([\boldsymbol{v}_{s_i}, \boldsymbol{v}_{o_j}, \boldsymbol{r}]\Big), \tag{3.1}$$

where $\boldsymbol{v}_{s_i}$ and $\boldsymbol{v}_{o_j}$ denote the node representations for the subject and object, respectively, $\boldsymbol{r}$ represents the relation query features, $[\cdot]$ indicates concatenation, and f_θ is a two-layer multilayer perceptron.

Loss Function. Following previous DETR-like methods [156, 86], we employ an L1 loss and a GIoU loss [106] for the bounding-box regression. For object or relation classification, we use Focal Loss [76] as a contrastive loss between predictions and language tokens.

To decode object and relation categories in a fully open-vocabulary manner, the fixed classifier (a fully connected layer) is replaced with a visual–concept alignment module (detailed in Section 3.2.3).

3.2.3 Learning Visual-Concept Alignment

A central component of our framework is the alignment of visual features with their corresponding textual concepts. At the node level, given an image, our model produces K predicted nodes $\{\tilde{v}_j\}_{j=1}^{K}$, which must be accurately paired with the N ground-truth nodes $\{v_i\}_{i=1}^{N}$. To achieve this, we employ a bipartite matching strategy akin to that used in standard DETR. The matching process is formulated as

$$\max_{\mathbf{M}} \sum_{i=1}^{N} \sum_{j=1}^{K} \mathrm{sim}(v_i, \tilde{v}_j) \cdot \mathbf{M}_{ij}, \tag{3.2}$$

where $\mathrm{sim}(\cdot, \cdot)$ measures the similarity between a ground-truth node and a predicted node, taking into account both spatial (*e.g.*, bounding box) and semantic (*e.g.*, category) cues. Here, $\mathbf{M} \in \mathbb{R}^{N \times K}$ is a binary assignment matrix with $\mathbf{M}_{ij} = 1$ if node v_i is matched with $\tilde{v}_j$, and $\mathbf{M}_{ij} = 0$ otherwise.

For each matched pair $(v_i, \tilde{v}_j)$, we enhance their alignment by jointly optimizing spatial and semantic consistencies. The spatial alignment is enforced by minimizing the L1 and GIoU losses between the corresponding bounding boxes. Concurrently, the categorical similarity is defined as

$$\mathrm{sim}_{\mathrm{cat}}(v_i, \tilde{v}_j) = \sigma\big(\langle \mathbf{w}_{v_i}, \mathbf{v}_j \rangle\big), \tag{3.3}$$

where $\mathbf{w}_{v_i}$ is the word embedding corresponding to the ground-truth node v_i, $\mathbf{v}_j$ is the visual representation of the predicted node $\tilde{v}_j$, $\langle \cdot, \cdot \rangle$ denotes the dot product, and $\sigma(\cdot)$ is the sigmoid function. This formulation serves to align the visual features of nodes with their semantic prototypes in the text space.

To extend relation recognition from a closed-set to an open-vocabulary setting, we further learn a joint visual-semantic space that aligns relation features with textual descriptions. Specifically, given a text input $\mathbf{t}$ processed by a text encoder E_t, and a relation feature $\mathbf{e}$, we compute the alignment score as

$$s(\mathbf{e}) = \langle \mathbf{e},\ f(E_t(\mathbf{t})) \rangle, \tag{3.4}$$

where $f(\cdot)$ is a fully connected layer that projects the text embedding into the relation feature space. The alignment is supervised via a binary cross-entropy loss defined as

$$\begin{aligned}\mathcal{L}_{\text{bce}} = &\frac{1}{|\mathcal{P}| + |\mathcal{N}|} \sum_{\mathbf{e} \in \mathcal{P} \cup \mathcal{N}} \Big[-y_{\mathbf{e}} \log \sigma\big(s(\mathbf{e})\big) - \\ &\big(1 - y_{\mathbf{e}}\big) \log\Big(1 - \sigma\big(s(\mathbf{e})\big)\Big)\Big],\end{aligned} \tag{3.5}$$

where $y_{\mathbf{e}} \in \{0, 1\}$ indicates whether $\mathbf{e}$ corresponds to a positive sample (with positive and negative samples denoted by $\mathcal{P}$ and $\mathcal{N}$, respectively) and $\sigma(\cdot)$ is the sigmoid function. This loss encourages high alignment scores for true correspondences, while penalizing false matches.

Learning visual-concept alignment is inherently challenging due to the scarcity of relation-aware pre-trained models on large-scale datasets. While object-language alignment has advanced considerably with models like CLIP [101] and GLIP [69], relation modeling remains largely underexplored. Moreover, the manual annotation of scene graphs is both labor-intensive and costly, limiting the scalability of SGG datasets. To address these challenges, we investigate three paradigms of relation-aware pre-training (see Section 3.2.1). By leveraging weak supervision, our approach facilitates the learning of rich representations for both objects and relationships.

3.2.4 Visual-Concept Retention with Knowledge Distillation

To enable the model to recognize a diverse set of objects and relationships beyond a limited fixed vocabulary, we build upon the visual-concept alignment introduced in Section 3.2.3. Empirically, we observe that directly optimizing the model on a new dataset using Eq. 3.5 leads to catastrophic forgetting—even when starting from a relation-aware pre-trained model. This issue is particularly evident in the *OvR-SGG* or *OvD+R-SGG* settings, where unseen (or novel) relationships are excluded from the scene graph. As a result, the model must discriminate novel relations from the background, a task that becomes increasingly challenging.

To alleviate this problem, we incorporate a knowledge distillation strategy that preserves the learned semantic space. Specifically, we designate the pre-trained model (obtained via weak supervision as described in Section 3.2.1) as the teacher, while the current model acts as the student. For each negative sample, we enforce that the student's edge features approximate those of the teacher. Formally, the distillation loss is defined as

$$\mathcal{L}_{\text{distill}} = \frac{1}{|\mathcal{N}|} \sum_{e \in \mathcal{N}} \left\| \boldsymbol{e}^s - \boldsymbol{e}^t \right\|_1 , \tag{3.6}$$

where $\boldsymbol{e}^s$ and $\boldsymbol{e}^t$ denote the student's and teacher's edge features, respectively, and $\mathcal{N}$ represents the set of negative samples. The overall training objective then becomes

$$\mathcal{L} = \mathcal{L}_{\text{bce}} + \lambda \mathcal{L}_{\text{distill}}, \tag{3.7}$$

with λ balancing the contributions of the ground-truth supervision and the distillation loss.

This knowledge distillation approach enables the model to acquire more precise information for base classes while preserving its capacity to generalize to unseen categories during fine-tuning. Consequently, our pipeline—pre-training on large-scale, coarse data followed by fine-tuning on a more fine-grained SGG dataset— becomes more effective and adaptable.

3.3 Experiments

3.3.1 Experimental setup

Datasets. The VG150 dataset [135] is a widely adopted benchmark that provides human-annotated labels for 150 object categories and 50 relation categories. It comprises a total of $108,777$ images, with 70% reserved for training, $5,000$ for validation, and the remainder for testing. Consistent with the evaluation of VS^3 [150], we remove

images utilized by the pre-trained object detector Grounding DINO [86], resulting in a test set of 14,700 images.

For relation-aware pre-training, we explored three different pipelines:

- *Scene Parser-based Pipeline:* We employ an off-the-shelf language parser [90] to extract relation triplets from image captions, yielding ∼104k images with coarse scene graphs from the COCO caption training set.
- *LLM-based Pipeline:* ∼113k COCO images are annotated with scene graphs by GPT-4 [97] in *GPT4SGG* [19].
- *Multimodal LLM-based Pipeline:* 1M images are annotated using Gemini 1.5 Flash [104] in *MegaSG* [16]. To prevent information leakage and reduce the computational burden, we select 644k images from *MegaSG* for pre-training.

GQA [46] utilizes the same image corpus as Visual Genome [59] but features more diverse annotations, comprising 1,703 object classes and 310 predicates. Similar to VQ150, GQA200 [25, 114] is a reduced version of GQA that contains 200 object classes and 100 predicate classes.

Metrics. We primarily adopt the *SGDet* protocol [135, 117] (also known as *SGGen*) for fair comparisons. SGDet generates a scene graph directly from an input image without any predefined object boxes. Performance is evaluated using Recall@K (K = 20/50/100), which measures the fraction of correctly predicted relation triplets among the top-K predictions. A triplet is considered correct if its subject, object, and predicate labels match the ground truth, and both subject and object regions have the same label or an IoU ≥ 0.5. To ensure comprehensive evaluation, we also report results for the *PredCls* setting, where ground-truth object labels and locations are provided. Since our approach detects objects in a one-stage manner, we implement *PredCls* by selecting image regions that best match the ground-truth objects in post-processing before performing relation recognition. For the *Closed-set SGG* setting,

Table 3.1: Zero-shot performance of state-of-the-art methods on the VG150 test set. For the COCO Caption dataset, a language parser [90] has been used for extracting triplets from the caption. To prevent information leakage, we sampled 644k images from MegaSG, ensuring that the CLIP similarity of each sampled image with the VG test set remained below 0.9.

SGG model	Training Data	SGDet						PredCls					
		R@20/50/100			mR@20/50/100			R@20/50/100			mR@20/50/100		
LSWS [143]		-	3.28	3.69		-			-			-	
MOTIFS [146]		5.02	6.40	7.33		-			-			-	
Uniter [15]	COCO [14]	5.42	6.74	7.62		-			-			-	
$VS^3_{(Swin\text{-}T)}$ [150]	Caption	4.56	5.79	6.79	2.18	2.59	3.00	12.30	16.77	19.40	3.56	4.79	5.51
$VS^3_{(Swin\text{-}L)}$ [150]	(104k)	4.82	6.20	7.48	2.29	2.70	3.09	12.54	17.28	19.89	3.57	4.83	5.56
$OvSGTR_{(Swin\text{-}T)}$		6.61	8.92	10.90	1.09	1.53	1.95	16.65	22.44	26.64	2.47	3.58	4.41
$OvSGTR_{(Swin\text{-}B)}$		6.85	9.33	11.47	1.28	1.79	2.18	16.82	22.79	27.04	2.94	4.24	5.26
$VS^3_{(Swin\text{-}T)}$ [150]		4.92	6.32	7.22	1.90	2.44	2.80	13.90	17.06	18.60	3.87	4.81	5.30
$VS^3_{(Swin\text{-}L)}$ [150]	COCO	5.07	7.40	9.50	1.30	1.93	2.42	14.53	18.72	20.73	3.87	4.91	5.46
$OvSGTR_{(Swin\text{-}T)}$	GPT4SGG [19]	7.35	9.66	11.14	2.73	3.67	4.52	21.03	24.43	25.98	6.61	7.82	8.54
$OvSGTR_{(Swin\text{-}B)}$	(113K)	7.65	10.10	11.73	2.92	3.84	4.69	21.13	24.78	26.25	7.10	8.49	9.37
$VS^3_{(Swin\text{-}T)}$ [150]		5.56	8.19	10.17	1.15	1.71	2.20	23.81	29.64	32.18	4.70	5.96	6.57
$VS^3_{(Swin\text{-}L)}$ [150]	MegaSG	9.74	14.80	18.80	1.57	2.71	3.75	31.88	38.77	41.76	5.32	6.88	7.58
$OvSGTR_{(Swin\text{-}T)}$	(644k)	**9.94**	**13.92**	**17.17**	**3.05**	**4.03**	**4.76**	**37.12**	**44.10**	**47.09**	**8.49**	**10.22**	**11.07**
$OvSGTR_{(Swin\text{-}B)}$		**10.36**	**14.61**	**18.13**	**3.01**	**4.10**	**4.98**	**39.04**	**45.86**	**48.54**	**8.45**	**10.30**	**11.12**

we additionally report Mean Recall@K (mR@K, K = 20/50/100) and inference speed.

Implementation Details. Our model is initialized with pre-trained Grounding DINO [86]. The visual backbone (*i.e.*, Swin-T or Swin-B) and the text encoder (*i.e.*, BERT-base [24]) remain frozen, while other components, including the relation-aware embedding, are randomly initialized. For pairwise relation recognition, we retain the top 100 detected objects per image. The model is trained on an 8×NVIDIA A100 GPU cluster using the AdamW [88] optimizer.

Our code is available at https://github.com/gpt4vision/OvSGTR/.

Table 3.2: Comparison with state-of-the-art methods on the VG150 test set (under SGDet protocol). The symbol ✧ denotes fully supervised methods. "VG Caption" is processed using the Scene Parser-based Pipeline, "VG@GPT4SGG" follows the LLM-based Pipeline, and "VG@Gemini" employs the Multimodal LLM-based Pipeline.

SGG model	Training Data	Grounding	R@20/50/100			mR@20/50/100		
IMP [135] ✧			17.73	25.48	30.71	2.66	4.10	5.29
MOTIFS [146] ✧	VG150		25.40	32.34	36.80	5.80	7.38	9.04
$VS^3_{(Swin\text{-}L)}$ [150] ✧	(∼56k)	-	27.34	36.04	40.88	4.43	6.45	7.81
$OvSGTR_{(Swin\text{-}B)}$ ✧			27.75	36.44	42.35	5.24	7.41	8.98
IMP [135]		GLIP-L [69]	6.51	9.54	11.86	0.88	1.41	1.89
LSWS [143]		-	-	3.85	4.04	-	-	-
MOTIFS [146]	VG Caption	Li *et al.* [73]	8.25	10.50	11.98	-	-	-
MOTIFS [146]	(∼73k)	GLIP-L [69]	13.88	18.46	21.81	3.71	4.85	5.58
$VS^3_{(Swin\text{-}L)}$ [150]		GLIP-L [69]	11.31	16.00	19.85	2.39	3.80	4.87
$OvSGTR_{(Swin\text{-}B)}$		GLIP-L [69]	16.36	22.14	26.20	3.80	5.24	6.25
IMP [135]			12.78	16.63	19.39	3.86	4.88	5.76
MOTIFS [146]	VG@GPT4SGG		16.41	20.46	23.23	5.86	7.36	8.51
$VS^3_{(Swin\text{-}L)}$ [150]	(∼46k)	-	17.77	22.42	25.29	4.24	5.82	6.97
$OvSGTR_{(Swin\text{-}B)}$			20.12	25.03	28.84	5.68	7.14	8.22
$VS^3_{(Swin\text{-}L)}$ [150]	VG@Gemini	-	17.10	22.70	26.10	3.60	5.11	6.26
$OvSGTR_{(Swin\text{-}B)}$	(∼50k)		19.78	26.08	30.66	5.07	6.72	7.86

3.3.2 Main Results

Relation-aware Pre-training. We systematically evaluate the zero-shot performance of recent state-of-the-art SGG methods under three different large-scale pre-training pipelines (see Section 3.2.1). As summarized in Table 3.1, large-scale pre-training yields consistent improvements in visual relationship recognition across a variety of models. In particular, our proposed *OvSGTR* (Swin-B) achieves substantial performance gains: *e.g.*, it improves R@20, R@50, and R@100 in the PredCls setting from baseline values of 16.82, 22.79, and 27.04 to 39.04, 45.86, and 48.54, re-

spectively. These improvements demonstrate the effectiveness of incorporating multimodal knowledge for relation-aware representation learning.

The superior results of *OvSGTR* are largely attributed to two key factors. First, the multimodal LLM-based pipeline produces high-quality scene graph annotations, leading to richer supervision signals for pre-training. Second, the use of large-scale data exposes the model to a diverse range of object and predicate instances, improving its ability to generalize to previously unseen visual relationships. The zero-shot improvements suggest that the learned representations capture more fine-grained semantic cues, enabling accurate predicate classification even when the specific *object-predicate-object* combinations were not explicitly encountered during training.

Moreover, the comparison in Table 3.2 shows that these relation-aware pre-training pipelines consistently improve SGG performance on the standard VG150 benchmark. The results demonstrate that large-scale, relation-centric pre-training strategies can effectively address the inherent data scarcity in SGG tasks. Compared with other baselines like [146, 117], our *OvSGTR* exhibits clear advantages in capturing visual relationships between objects, highlighting the importance of leveraging richer linguistic and visual cues during training. Overall, these findings underscore the idea that combining large-scale multimodal pre-training with specialized SGG methods can significantly enhance the recognition of complex visual relationships. By integrating knowledge from pre-trained language models and carefully curated scene graph annotations, *OvSGTR* achieves state-of-the-art performance, enhancing the recognition of complex visual relationships.

Closed-set SGG Benchmark. The *closed-set SGG* setting follows previous works [146, 117, 70, 135, 150] and employs the VG150 dataset [135] with full manual annotations for training and evaluation. Experimental results on the VG150 test set (Table 3.3) show that our proposed model outperforms all competitors. Notably, compared to the recent VS^3 [150], our *OvSGTR* (with Swin-T) achieves performance

Table 3.3: Experimental results of *Closed-set SGG* on VG150 test set. "40M/177M" in Params. refers to 40M trainable parameters and 177M total parameters. Inference time is benchmarked on an NVIDIA RTX 3090 GPU with batch size 1 and an input resolution 1000×600. Time for SGNLS [152] is benchmarked on an NVIDIA A100 GPU (80G) due to memory out of usage. Throughout this chapter, we use $\star$ to indicate relation-aware pre-training on MegaSG.

SGG model	Backbone	Detector	Params.	R@20/50/100			mR@20/50/100			Time (s)
IMP [135]	RX-101	Faster R-CNN	146M/308M	17.7	25.5	30.7	2.7	4.1	5.3	0.25
MOTIFS [146]	RX-101		205M/367M	25.5	32.8	37.2	5.0	6.8	7.9	0.27
VCTREE [118]	RX-101		197M/358M	24.7	31.5	36.2	-	-	-	0.38
SGNLS [152]	RX-101		165M/327M	24.6	31.8	36.3	-	-	-	> 7
HL-Net [79]	RX-101		220M/382M	26.0	33.7	38.1	-	-	-	0.10
FCSGG [82]	HRNetW48	-	87M/87M	13.6	18.6	22.5	2.3	3.2	3.9	0.13
SGTR [70]	R-101	DETR	36M/96M	-	24.6	28.4	-	-	-	0.21
VS^3 [150]	Swin-T	-	93M/233M	26.1	34.5	39.2	-	-	-	0.16
VS^3 [150]	Swin-L	-	124M/432M	27.3	36.0	40.9	4.4	6.5	7.8	0.24
OvSGTR	Swin-T	DETR	41M/178M	27.0	35.8	41.3	5.0	7.2	8.8	0.13
OvSGTR	Swin-B		41M/238M	27.8	36.4	42.4	5.2	7.4	9.0	0.19
OvSGTR*	Swin-T		41M/178M	27.3	36.2	41.9	5.3	7.7	9.4	0.13
OvSGTR*	Swin-B		41M/238M	**28.6**	**37.6**	**43.4**	**5.8**	**8.3**	**10.2**	0.19

gains of up to 4.9% for R@50 and 6.9% for R@100. Improvements in mR@K further indicate that our model better addresses long-tail bias. Moreover, the performance improvement (with vs. without relation-aware pretraining, *e.g.*, 27.8/36.4/42.4 vs. 28.6/37.6/43.4) demonstrates that large-scale relation-aware pre-training is beneficial for recognizing both visual relationships and objects.

While many prior approaches rely on complex message-passing mechanisms to extract relation features, our model attains strong performance with a simpler relation head consisting of only two MLP layers. For instance, *OvSGTR* (with Swin-T) achieves results comparable to, or even superior to, those of VS^3 (with Swin-L), while requiring

Table 3.4: Experimental results (R@20/50/100) of *OvD-SGG* setting on VG150 test set.

Method	Base+Novel (Object)		Novel (Object)	
	PredCls	SGDet	PredCls	SGDet
IMP [135]	46.79 / 54.21 / 56.85	2.09 / 2.85 / 3.45	42.02 / 51.15 / 54.36	0.00 / 0.00 / 0.00
MOTIFS [146]	45.88 / 53.54 / 56.23	2.61 / 3.30 / 3.83	39.58 / 49.63 / 53.00	0.00 / 0.00 / 0.00
VCTREE [118]	48.38 / 55.69 / 58.05	2.76 / 3.47 / 3.95	42.96 / 52.28 / 55.75	0.00 / 0.00 / 0.00
TDE [117]	52.49 / 60.18 / 62.47	2.68 / 3.49 / 4.01	48.90 / 58.69 / 61.50	0.00 / 0.00 / 0.00
VS^3 [150] (Swin-T)	39.83 / 46.73 / 49.11	9.85 / 14.49 / 17.87	36.16 / 44.51 / 47.63	6.02 / 10.24 / 13.44
OvSGTR (Swin-T)	54.05 / 60.58 / 62.10	12.34 / 18.14 / 23.20	51.47 / 59.01 / 60.65	6.90 / 12.06 / 16.49
OvSGTR (Swin-B)	53.81 / 59.83 / 61.34	**15.43** / **21.35** / **26.22**	51.16 / 58.30 / 60.00	10.21 / 15.58 / 19.96
OvSGTR* (Swin-T)	52.24 / 58.60 / 60.35	14.33 / 20.91 / 25.98	50.67 / 58.18 / 60.15	**10.52** / **17.30** / **22.90**
OvSGTR* (Swin-B)	**54.74** / **60.93** / **62.49**	15.21 / 21.21 / 26.12	**53.21** / **60.78** / **62.60**	10.31 / 15.78 / 20.47

fewer trainable parameters (41M vs. 124M) and exhibiting lower inference latency (0.13 s vs. 0.24 s).

OvD-SGG Benchmark. Following previous works [40, 150], the *OvD-SGG* setting ensures that models do not see novel object categories during training. Specifically, 70% of the VG150 object categories are designated as base categories, and the remaining 30% serve as novel categories. The experiments mirror those in the *Closed-set SGG* scenario, except that annotations for novel object categories are omitted during training. After excluding unseen object nodes, the VG150 training set contains 50,107 images.

Table 3.4 reports the performance under the *OvD-SGG* setting in terms of "Base+Novel (Object)" and "Novel (Object)." The results show that our proposed model substantially outperforms previous methods, indicating a strong open-vocabulary recognition and generalization capability. Compared to VS^3 [150], our approach yields improvements of up to 68.9% and 70.4% in R@50 and R@100, respectively, on novel categories. Since *OvD-SGG* only excludes nodes of novel object categories, the learning process for relations remains unaffected. This highlights that the open-vocabulary ability of

Table 3.5: Experimental results of *OvR-SGG* setting on VG150 test set. ‡ denotes *w.o.* relation-aware pre-training. † denotes *w.o.* distillation but *w.* relation-aware pre-training on COCO Caption (*i.e.*, using the Scene Parser-based Pipeline).

Method	Base+Novel (Relation)		Novel (Relation)	
	PredCls	SGDet	PredCls	SGDet
IMP [135]	23.41 / 25.74 / 26.59	9.16 / 12.42 / 14.48	0.00 / 0.00 / 0.00	0.00 / 0.00 / 0.00
MOTIFS [146]	24.82 / 27.13 / 27.99	12.41 / 15.29 / 16.81	0.00 / 0.00 / 0.00	0.00 / 0.00 / 0.00
VCTREE [118]	26.91 / 29.34 / 30.16	12.38 / 15.11 / 16.81	0.00 / 0.00 / 0.00	0.00 / 0.00 / 0.00
TDE [117]	26.69 / 28.98 / 29.76	12.43 / 15.37 / 17.21	0.00 / 0.00 / 0.00	0.00 / 0.00 / 0.00
VS^3 [150] (Swin-T)	22.27 / 24.32 / 24.99	12.45 / 15.63 / 17.29	0.00 / 0.00 / 0.00	0.00 / 0.00 / 0.00
OvSGTR (Swin-T)‡	26.57 / 28.46 / 29.09	14.60 / 18.09 / 20.41	0.00 / 0.00 / 0.00	0.00 / 0.00 / 0.00
OvSGTR (Swin-T)†	25.57 / 27.63 / 28.28	14.32 / 17.69 / 19.96	0.17 / 0.25 / 0.27	0.05 / 0.10 / 0.15
OvSGTR (Swin-T)	31.83 / 37.58 / 40.68	15.85 / 20.46 / 23.86	21.01 / 27.42 / 31.30	10.17 / 13.45 / 16.19
OvSGTR$^⋆$ (Swin-T)	40.60 / 46.82 / 49.03	19.38 / 25.40 / 29.71	28.91 / 36.78 / 39.96	12.23 / 17.02 / 21.15
OvSGTR (Swin-B)‡	26.83 / 28.83 / 29.42	15.39 / 19.07 / 21.37	0.00 / 0.00 / 0.00	0.00 / 0.00 / 0.00
OvSGTR (Swin-B)†	26.21 / 28.30 / 28.96	15.00 / 18.58 / 20.84	0.10 / 0.13 / 0.14	0.06 / 0.08 / 0.10
OvSGTR (Swin-B)	34.37 / 41.04 / 44.67	17.63 / 22.89 / 26.65	24.95 / 32.85 / 37.87	12.09 / 16.39 / 19.72
OvSGTR$^⋆$ (Swin-B)	**44.53** / **51.31** / **53.71**	**21.09** / **27.92** / **32.74**	**38.27** / **47.42** / **50.90**	**16.59** / **22.86** / **27.73**

the object detector plays a more decisive role in *OvD-SGG*.

OvR-SGG Benchmark. Different from *OvD-SGG* which removes all unseen nodes, *OvR-SGG* only removes unseen edges but retains the original nodes. Considering that VG150 has 50 relation categories, we randomly select 15 of them as unseen (novel) relation categories. During training, only base relation annotations are available. After removing these unseen edges, 44,333 images from VG150 remain for training.

Similar to *OvD-SGG*, Table 3.5 reports the performance of *OvR-SGG* in terms of both "Base+Novel (Relation)" and "Novel (Relation)." From Table 3.5, the proposed *OvSGTR* notably outperforms other competitors, even without distillation. However, a marked decline in performance is observed across all methods (including *OvSGTR* without distillation) in the "Novel (Relation)" categories, highlighting the inherent difficulty of recognizing novel relations in the *OvR-SGG* paradigm. Nevertheless, with

Table 3.6: Experimental results of *OvD+R-SGG* setting on VG150 test set. † denotes *w.o.* distillation but *w.* relation-aware pre-training on COCO Caption (*i.e.*, using the Scene Parser-based Pipeline).

Method	Joint Base+Novel			Novel (Object)			Novel (Relation)		
	R@20	R@50	R@100	R@20	R@50	R@100	R@20	R@50	R@100
IMP [135]	0.57	0.81	0.99	0.00	0.00	0.00	0.00	0.00	0.00
MOTIFS [146]	0.77	0.95	1.10	0.00	0.00	0.00	0.00	0.00	0.00
VCTREE [118]	0.84	1.06	1.17	0.00	0.00	0.00	0.00	0.00	0.00
TDE [117]	0.81	0.99	1.13	0.00	0.00	0.00	0.00	0.00	0.00
VS^3 [150] (Swin-T)	4.03	5.87	7.19	3.73	5.98	7.47	0.00	0.00	0.00
OvSGTR (Swin-T)†	5.20	7.88	10.06	4.18	6.82	9.23	0.00	0.00	0.00
OvSGTR (Swin-T)	10.02	13.50	16.37	10.56	14.32	17.48	7.09	9.19	11.18
OvSGTR* (Swin-T)	10.67	15.15	18.82	8.22	12.49	16.29	9.62	13.68	17.19
OvSGTR (Swin-B)†	7.85	11.23	14.21	9.26	13.27	16.83	1.20	1.78	2.57
OvSGTR (Swin-B)	12.37	17.14	21.03	**12.63**	**17.58**	**21.70**	10.56	14.62	18.22
OvSGTR* (Swin-B)	**12.54**	**17.84**	**21.95**	10.29	15.66	19.84	**12.21**	**17.15**	**21.05**

visual-concept retention, the performance of *OvSGTR* (*w/* Swin-T) on novel relations is significantly improved from 0.10 (R@50, SGDet) to 13.45 (R@50, SGDet).

Further, with large-scale relation-aware pre-training on MegaSG, *OvSGTR* (*w/* Swin-B) achieves a notable performance gain (*e.g.*, 16.59/22.86/27.37 vs. 12.09/16.39/19.72 in SGDet for novel relations) compared to pre-training on COCO Caption. This result showcases that scaling relation-aware pre-training is both valuable and effective.

OvD+R-SGG Benchmark. To further extend the SGG task to a fully open-vocabulary domain, we propose the *OvD+R-SGG* benchmark, where both novel object and novel relation categories are excluded during training. Specifically, we merge the splits from *OvD-SGG* and *OvR-SGG* using their respective base object and base relation categories, resulting in 36,425 training images from VG150.

Table 3.6 presents the performance of *OvD+R-SGG* under three settings: *Joint Base+Novel* (*i.e.*, all object and relation categories), *Novel (Object)* (*i.e.*, only novel object categories), and *Novel (Relation)* (*i.e.*, only novel relation categories). As with *OvR-SGG*, we observe catastrophic forgetting in *OvD+R-SGG*, but our proposed visual-concept retention significantly mitigates this issue. Compared to existing methods, our model achieves notable gains across all metrics. For instance, under the *Joint Base+Novel* setting, the proposed *OvSGTR* (w/ Swin-B) achieves 17.84% (R@50, SGDet), surpassing VS^3 (5.87%) by 11.97 points. Moreover, for *Novel (Object)* categories, our approach attains 15.66% (R@50, SGDet, compared to 5.98% from VS^3), and for *Novel (Relation)* categories, *OvSGTR* (w/ Swin-B) reaches 17.15% (R@50, SGDet), outperforming VS^3, which failed to recognize novel relationships.

Overall Analysis. Experimental results highlight distinct challenges across the four settings. Based on these findings: **1)** Many previous methods rely on a two-stage object detector, Faster R-CNN [105], and complex message-passing mechanisms. However, our model demonstrates that a one-stage DETR-based framework can significantly outperform R-CNN-like architectures, even when using only a single MLP for relation feature representation. **2)** Previous methods employing a closed-set object detector struggle to recognize objects without textual cues in object-involved open-vocabulary SGG (*i.e.*, *OvD-SGG* and *OvD+R-SGG*). **3)** The performance drop in *OvD+R-SGG* compared to other settings suggests it is significantly more challenging, underscoring the need for further exploration toward fully open-vocabulary SGG. **4)** Large-scale relation-aware pre-training substantially enhances the generalization ability of SGG models across both closed-set and open-vocabulary scenarios. This indicates that the synthesized SGG dataset plays a crucial role in bridging the gap between these settings.

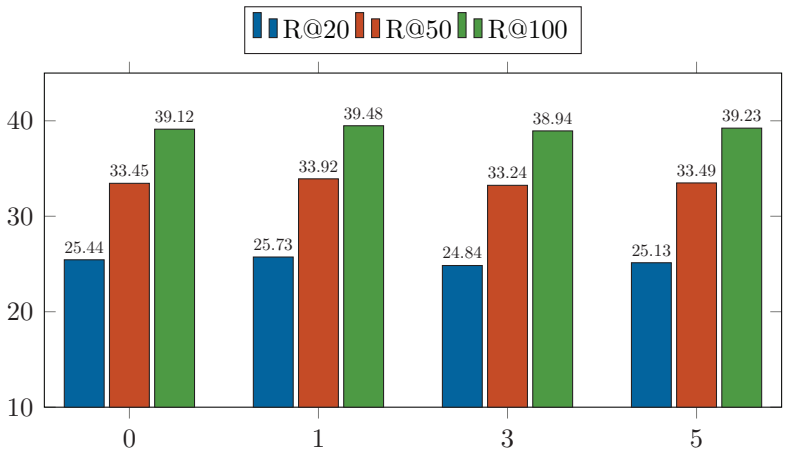


Figure 3.4: Ablation study of relation queries on VG150 validation set under the setting of *Closed-set SGG*.

3.3.3 Ablation Studies

Effect of Relation Queries. We first consider removing the relation query embedding. In this case, the relation feature is computed by

$$\boldsymbol{e}_{s_i \to o_j} = f_\theta\big([\boldsymbol{v}_{s_i}, \boldsymbol{v}_{o_j}]\big), \tag{3.8}$$

which only encodes features from the subject and object nodes. We further extend Eq. (3.1) to a more general form:

$$\boldsymbol{e}_{s_i \to o_j} = \frac{1}{M} \sum_{n=1}^{M} f_\theta\big([\boldsymbol{v}_{s_i}, \boldsymbol{v}_{o_j}, \boldsymbol{r}_n]\big), \tag{3.9}$$

where multiple relation queries are averaged. As shown in Figure 3.4, the model achieves the highest performance when the number of relation queries is set to 1. This result can be interpreted from two perspectives: (i) relation queries interact with all edges during training, thereby capturing global information from the entire dataset; (ii) increasing the number of relation-aware queries does not introduce additional supervision but increases the optimization burden.

Hyper-parameter λ for Distillation. Table 3.7 shows the impact of varying the hyper-parameter λ. When $\lambda = 0.1$, the model achieves the best performance. By contrast,

Table 3.7: Impact of hyper-parameter λ for distillation loss on VG150 validation set under the setting of *OvR-SGG*. $a \rightarrow b$ refers to the performance shift from a (initial checkpoint's performance) to b during training.

λ	Base+Novel		Novel	
	R@50	R@100	R@50	R@100
0	7.25 → 13.74	8.98 → 16.11	10.78 → 0.32	13.24 → 0.38
0.1	7.25 → **16.00**	8.98 → **19.20**	10.78 → **11.54**	13.24 → **13.94**
0.3	7.25 → 14.35	8.98 → 17.04	10.78 → 10.71	13.24 → 12.71
0.5	7.25 → 13.34	8.98 → 16.08	10.78 → 10.90	13.24 → 13.22

removing distillation leads to a substantial drop in performance for novel categories, indicating that the model struggles to retain the knowledge inherited from pre-trained models for these categories.

3.3.4 Extension: Comparison on GQA Benchmark

Beyond the widely used VG150 benchmark, we also validate our framework on the GQA dataset. Following prior works [25, 114, 65], we use the GQA200 split that contains 200 objects and 100 predicates. As shown in Table 3.8, our proposed *OvSGTR* demonstrates a strong generalization capability on GQA200. Without fine-tuning, both Swin-T and Swin-B variants of *OvSGTR* achieve remarkable performance in the PredCls setting. However, their SGDet scores remain low due to domain shift and lack of localization supervision.

After fine-tuning the GQA200 training set, both variants show significant improvements, particularly under the SGDet protocol. Notably, OvSGTR-B achieves 33.9% (R@50) and 39.2% (R@100) in SGDet, surpassing all previous competitors. These results highlight the effectiveness of our relation-aware pre-training and visual-concept alignment mechanisms, which enable the model to adapt to novel domains with min-

Table 3.8: Experimental results on the GQA200 test set. OvSGTR is pre-trained on MegaSG. "OvSGTR-T" refers to our model with the Swin-T backbone, while "OvSGTR-B" denotes the variant with the Swin-B backbone.

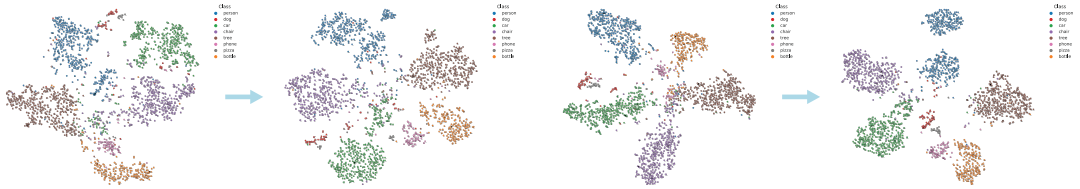
Method	PredCls		SGDet	
	R@50	R@100	R@50	R@100
IMP [135]	57.5	59.6	22.3	26.7
MOTIFS [146]	60.2	62.1	24.8	28.8
VCTREE [118]	62.2	63.9	27.1	30.8
TDE [117]	64.2	66.0	27.2	31.5
OvSGTR-T (*w/o* fine-tuning)	38.0	40.5	11.6	13.9
OvSGTR-T (*w/* fine-tuning)	60.7	62.7	32.8	37.9
OvSGTR-B(*w/o* fine-tuning)	38.2	40.9	11.4	13.9
OvSGTR-B (*w/* fine-tuning)	61.5	63.4	**33.9**	**39.2**

imal supervision.

Overall, the GQA results further validate the scalability and effectiveness of *OvSGTR* under different dataset distributions, reinforcing its applicability in real-world, diverse visual environments.

3.3.5 Visualization and Discussion

Relation-aware pre-training. To enrich the understanding of relation-aware pre-training, we visualize the object and relation features using t-SNE [121] in Figure 3.5. Object and relation features become more discriminative and semantically clustered after pre-training on *MegaSG*. For instance, object features like "dog", "pizza", and "phone" form tighter, more coherent clusters. Similarly, relation features such as "on," "riding", and "holding" become more separable, resulting in better classification of relationships.



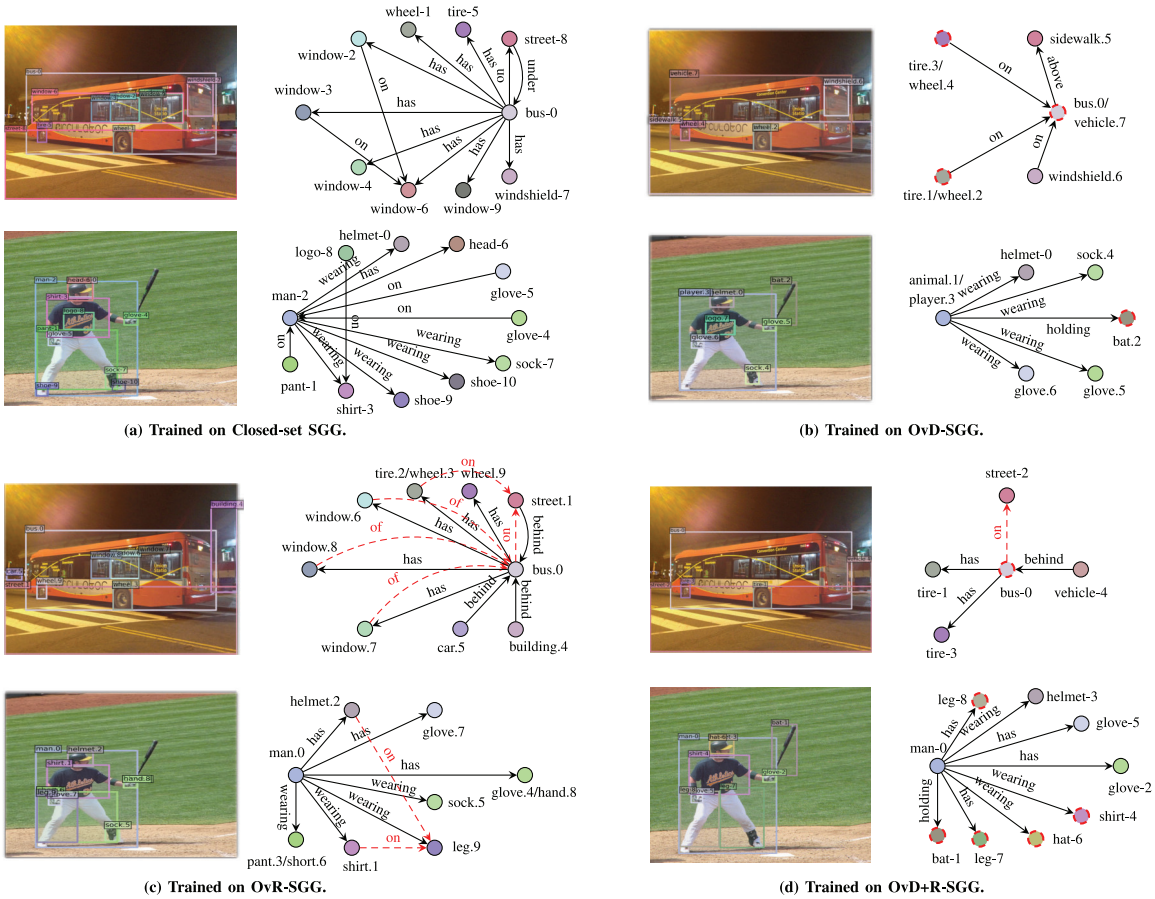
(a) Object features (*w/o* relation-aware pre-training). (b) Object features (*w/* relation-aware pre-training).

(c) Relation features (*w/o* relation-aware pre-training).(d) Relation features (*w/* relation-aware pre-training).

Figure 3.5: t-SNE [121] visualizations of object and relation features. (a)–(b) show object features extracted by OvSGTR-B without and with relation-aware pre-training, respectively. (c)–(d) show relation features under the same settings. Arrows indicate the evolution of features after fine-tuning on VG150.

This relation-aware pre-training significantly boosts generalization, especially in open-vocabulary scenarios, where models must reason over novel object and relation categories. By learning transferable representations from weak supervision, our framework achieves superior zero-shot performance and mitigates reliance on fully annotated datasets.

Qualitative Results. We present qualitative results of our model trained under four settings, as shown in Figure 3.6. From the figure, the model trained on the *Closed-set SGG* setting tends to generate denser scene graphs, as all object and relationship categories are available during training. In contrast, the model trained under the *OvD-SGG* setting demonstrates the ability to detect novel objects (*e.g.*, "bat", "bus"), although the relationships are limited to the seen predicate set. Under the *OvR-SGG* setting, all object nodes are retained, but the model must infer novel relationships. This results in sparser graphs; however, the model still predicts some



(a) Trained on Closed-set SGG. **(b) Trained on OvD-SGG.**

(c) Trained on OvR-SGG. **(d) Trained on OvD+R-SGG.**

Figure 3.6: Qualitative results of our model (w. Swin-T) on VG150 test set (best view in color). For clarity, we only show triplets with high confidence in top-20 predictions. Dashed nodes or arrows refer to novel object categories or novel relationships. It is worth noticing that the "bat" class does not exist in the VG150 dataset.

unseen relations (*e.g.*, "on", "of") with reasonable accuracy. Notably, the model trained on *OvD+R-SGG*, despite lacking full supervision of both novel objects and relationships, is still able to recognize unseen object categories such as "bus" and "bat" (which does not exist in the VG150 dataset), along with novel relationships like "on". This highlights the strong generalization ability of our approach under fully open-vocabulary settings.

3.4 Summary

This chapter presents a significant advancement in Scene Graph Generation (SGG) by extending the conventional closed-set formulation to a fully open-vocabulary setting that simultaneously addresses novel objects and relationships. To comprehensively explore this open-world challenge, we define four distinct task settings—*Closed-SGG*, *OvD-SGG*, *OvR-SGG*, and *OvD+R-SGG*—each representing a different level of openness in object and relation vocabularies.

To tackle these settings, we propose *OvSGTR*, a unified transformer-based framework that aligns visual features with semantic representations from both base and novel categories. This design empowers the model to capture intricate inter-object relationships and generalize effectively to previously unseen entities and interactions.

In addition, we explore three relation-aware pre-training pipelines to acquire transferable relational representations and employ a knowledge distillation strategy to alleviate catastrophic forgetting during fine-tuning. Extensive experiments on the VG150 and GQA benchmarks demonstrate that our method consistently achieves state-of-the-art performance across all evaluated settings.

Overall, this work demonstrates the benefits of combining large-scale relation-aware pre-training with effective generalization techniques, contributing to the advancement of scalable and open-world scene understanding.

Chapter 4

Weakly-Supervised SGG with Language Models

In Chapter 3, we discussed open-vocabulary Scene Graph Generation (SGG), where relation-aware pre-training has demonstrated significant impact. However, learning scene graphs from rich yet unannotated data remains a fundamental challenge in the weakly supervised SGG setting. To address this issue, this chapter introduces two variant approaches: *GPT4SGG*, a language-only LLM-based framework for synthesizing scene graphs from image-caption pairs, developed prior to the advent of multimodal LLMs; and *MegaSG*, a large-scale dataset constructed using a simple yet effective pipeline based on a multimodal LLM.

4.1 Challenges in Learning Scene Graphs from Caption

Conventional SGG models [135, 146, 118, 117, 20, 72] rely on datasets annotated manually, encompassing objects and their inter-relations. For an image with N objects,

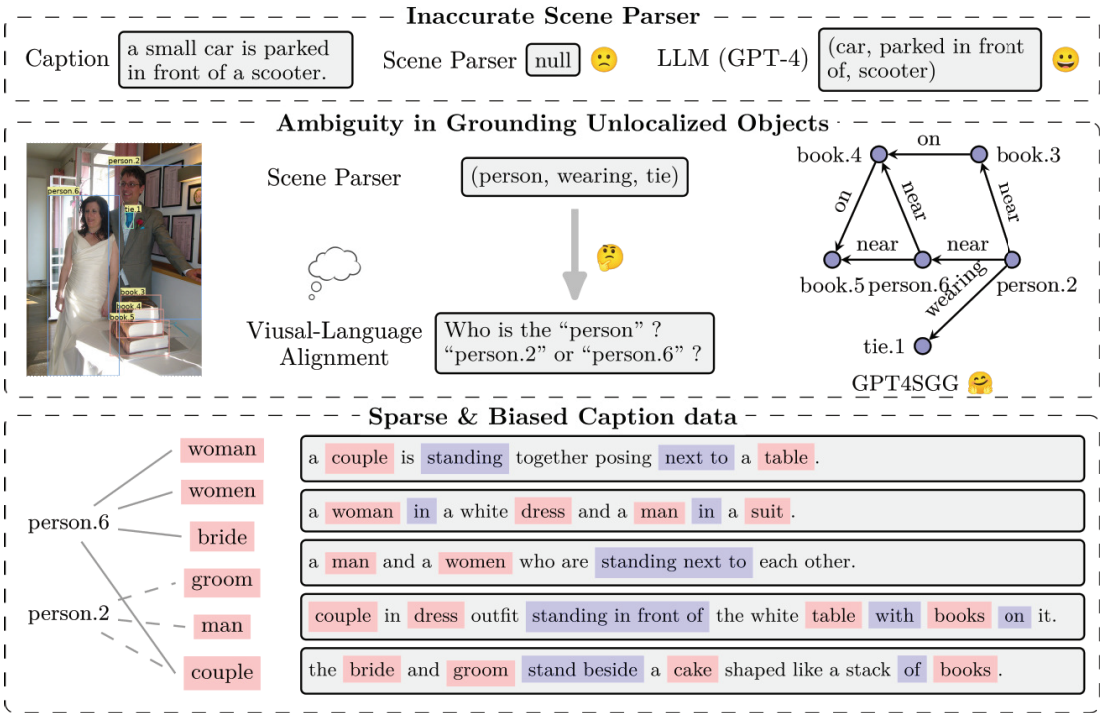


Figure 4.1: Challenges in learning scene graphs from natural language data. Best viewed in color and by zooming in.

there are $N*(N-1)$ potential pairs for annotation, making the creation of large-scale datasets both time-consuming and costly. To overcome this bottleneck, recent studies [152, 73, 150, 18] leverage image-caption data to learn scene graphs from natural language description directly, which is commonly referred to as **language-supervised SGG** or **weakly-supervised SGG**. Under this setting, previous approaches [152, 73] typically follow the pipeline: (1) *parsing relationship triplets from caption data* $\rightarrow$ (2) *grounding unlocalized objects of parsed triplets* $\rightarrow$ (3) *learning scene graphs with parsed triplets and grounded objects.* These approaches have demonstrated the potential of using natural language descriptions to generate scene graphs, providing a more scalable alternative to manually annotated datasets.

Nevertheless, **several inherent challenges hinder the effective learning of scene graphs from natural language descriptions**, as shown in Figure 4.1. **1)**

The scene parser [90] struggles to extract complete scene graphs from caption data, failing to derive relationship triplets even from simple sentences like *a small car is parked in front of a scooter*. For instance, it has been observed that there are 8% failure cases of the scene parser [90] on COCO caption [14] train set, highlighting its difficulty in accurately interpreting and structuring relationship information from natural language descriptions. By contrast, GPT-4 [97] works well for this task. However, directly replacing the scene parser with GPT-4 does not address the following issues. **2)** Ambiguity arises in grounding unlocalized objects of parsed triplets through visual-language alignment, particularly when multiple instances of the same category appear in an image, making it challenging to accurately match visual regions with language queries. For instance, given a parsed triplet $< person, on, tie >$, it is hard to choose which object is our target only based on the word "*person*" and the visual information, although the grounding model knows there are a man and a woman in this image. **3)** Caption data, even with manual annotations, exhibit sparsity and bias, often emphasizing specific observations while overlooking crucial visual cues for generating comprehensive scene graphs. These challenges are intertwined and significantly hamper learning scene graphs from natural language, leading to incomplete or biased scene graphs.

To generate more accurate and comprehensive scene graphs, an intuitive approach is to enrich the caption data by paraphrasing [57], or to use a more advanced scene parser [150]. However, this approach is sensitive to the quality of the captions and is highly affected by the parsing accuracy of the scene parser, which cannot circumvent the ambiguity issue in visual-language alignment. To reduce dependence on high-quality caption data and address the ambiguity issue, we prioritize the localization of objects (*Step 1: Grounding* in Figure 4.2) and decompose a complex scene into a set of simpler regions (*Step 2: Decomposing* in Figure 4.2). Specifically, object localization is achieved either through object detection annotations or a pre-trained object detector, ensuring accurate visual-language alignment. With these localized

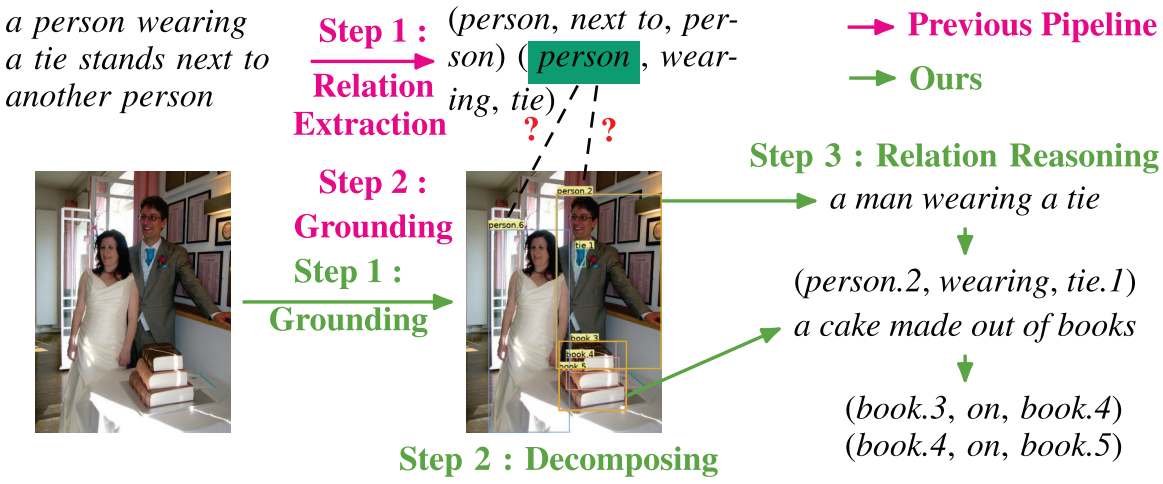


Figure 4.2: Our framework vs. previous pipeline of language-supervised SGG. Best viewed in color and by zooming in.

objects, multiple regions are sampled by the union of two objects. Further, we utilize a caption model to generate captions for the entire image and for the individual regions. The generated holistic and region-specific narratives provide rich visual information and are less biased by annotation preferences (*e.g.*, *woman*, *women*, and *bride* all refer to the same object, *person.6*, in Figure 4.1).

To synthesize a complete scene graph for an image from dense narratives, we utilize an LLM (*e.g.*, GPT-4) to perform relationship deduction and associate relationship triplets with localized objects (*Step 3: Relation Reasoning* in Figure 4.2). Compared to extracting relationship triplets one by one via a scene parser, feeding rich visual information directly to an LLM can generate more consistent and accurate scene graphs. For instance, given a caption like "*a cake made out of books*" (see Figure 4.2), the LLM can output correct relationship triplets while ignoring the fictional concept "*cake*". This process is difficult to replicate using a traditional scene parser.

Incorporating the above strategies, we develop the *GPT4SGG* framework. The strength of *GPT4SGG* lies in its ability to create a high-quality SGG dataset without human annotation and in its capacity to enhance the performance of SGG models.

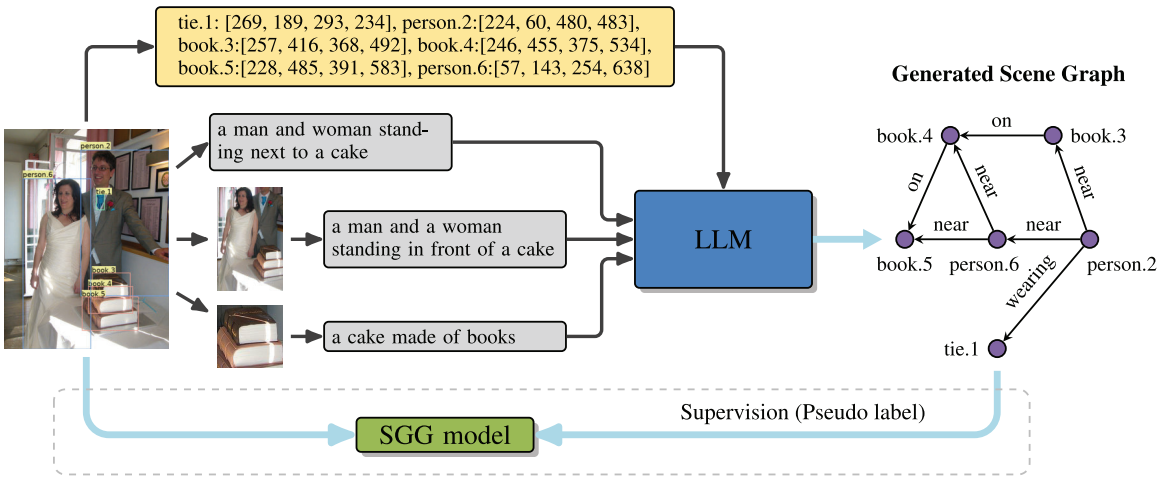


Figure 4.3: An overview of our approach *GPT4SGG*. *GPT4SGG* decomposes a complex scene into a set of region narratives and a global narrative, and utilizes an LLM to deduce relationships based on the localized objects and narratives.

To validate the proposed framework, we build two SGG-aware instruction-following datasets and conduct extensive experiments with state-of-the-art SGG models. Compared to data generated by a scene parser, the newly constructed dataset is less biased towards conjunction words (*e.g.*, "with", "in", "from") and more focused on physical interactions (*e.g.*, "wearing", "holding") between objects. With these more accurate and comprehensive scene graphs, various SGG models have demonstrated significant performance gains over previous approaches. For instance, a classical SGG model, IMP [135], has shown a 70% improvement in recall, while a newly state-of-the-art model, VS^3 [150], has achieved a 40% performance gain. These results indicate the effectiveness of *GPT4SGG* in advancing the state-of-the-art in Scene Graph Generation, offering a valuable resource for future research and real-world applications.

4.2 GPT4SGG: Synthesizing Scene Graphs from Region Captions

This section will describe the framework, *GPT4SGG*, how to generate more accurate and comprehensive scene graphs in a new way. An overview of *GPT4SGG* is illustrated in Figure 4.3.

4.2.1 Object Grounding

Object detection or grounding have been well-studied, such as [105, 103, 7, 145, 69, 69, 86]. Briefly, we can perform object grounding or detection using object detectors to obtain large-scale images with localized objects. Given an image I and a set of pre-defined object categories, object grounding involves identifying a set of instances $\{o_i\}_{i=1}^{M}$, where each instance o_i is associated with an object category and a bounding box. Alternatively, existing object detection datasets already provide large-scale images with localized objects. This preliminary step ensures that each visual object is accurately aligned with its respective category or phrase, thereby preventing ambiguity when multiple instances of the same category are present.

4.2.2 Scene Decomposing: Holistic & Region Narratives Generation

"*A picture is worth a thousand words*" is a well-known saying, as a complex scene contains rich visual clues. However, current captioning models [67, 126, 93] often produce brief descriptions that struggle to capture both global context and local details. A straightforward approach is to "look" at the image multiple times, akin to the sliding windows used in R-CNN [33]. Instead, by focusing on localized objects, we can simplify the task by selecting representative Regions of Interest (RoIs).

Algorithm 1 Generating Holistic & Region Narratives

Input: Original Image, N (maximum number of RoIs)

```
procedure GenerateNarratives(Image)
    GlobalDescription ← Blip-2(Image)
    Objects ← Detector(Image) or Annotation(Image)
    RoIList ← SelectRoIs(Objects, N)
    LocalisedDescriptions ← [ ]
    for each RoI in RoIList do
        CroppedRegion ← Crop Region with RoI
        Description ← Blip-2(CroppedRegion)
        Add Description to LocalisedDescriptions
    end for
    return GlobalDescription, LocalisedDescriptions
end procedure

function SelectRoIs(Objects, N)
    ValidPairs ← [ ]
    for pair in pairwise-combinations(Objects) do
        Add pair to ValidPairs if IoU(pair) > 0
    end for
    ValidPairs ← randomly select N samples from ValidPairs
    return Union of all pairs in ValidPairs
end function
```

As shown in Algorithm 1, we generate narratives at two levels, holistic and region-specific, to capture both the global context and detailed interactions within the scene. Specifically, the holistic narrative is generated by feeding the entire image to the captioning model, resulting in a comprehensive description of the scene's content. In contrast, region-specific narratives are produced by captioning a set of RoIs, each focusing on the content of localized regions, particularly the interactions between pairs of objects.

To select meaningful and representative RoIs for an input image, localized objects

are pairwise combined and filtered using an Intersection-over-Union (IoU) threshold greater than zero. This selection criterion is based on the premise that objects with a non-zero IoU are more likely to share a meaningful relationship, as indicated in previous works [146, 152]. This approach also reduces the computational burden of captioning and simplifies the inference process for large language models (LLMs).

The above approach of generating narratives offers a significant advantage in that it is resolution-independent. Captioning models, such as BLIP-2 [67], typically receive a relatively small input resolution like 224×224. In contrast, our approach can be adapted to any image resolution. For high-resolution images and complex scenes, we can obtain plentiful and meaningful descriptions focusing on interactions between objects. This advantage has also been demonstrated when compared to LlaVA, one of the state-of-the-art open-source multi-modal LLMs.

4.2.3 Relation Reasoning: Synthesizing Scene Graphs with LLM

Given a set of region-specific narratives and a global narrative, one might consider following the traditional pipeline (see Figure 4.2) to extract relationship triplets using a scene parser. However, this approach presents several challenges. For instance, as illustrated in Figure 4.3, the phrase *a cake made of books* could be parsed into the triplet (*cake*, *made of*, *book*), despite the fact that no *cake* exists in the image and multiple *books* are present. Another significant challenge is maintaining semantic consistency; that is, ensuring that the model correctly associates different aliases for the same entity (*e.g.*, "*man*," "*people*," "*person*" for *person.2* in Figure 4.3).

To address these issues and improve the accuracy of relation reasoning, we utilize a Large Language Model (LLM) to synthesize scene graphs from the narratives and object information. Specifically, we design a prompt template, as shown in Table A.1, which instructs the model to use object information (such as categories and

bounding boxes) alongside holistic and region-specific descriptions to establish relationships between objects. To accurately differentiate between multiple instances of the same category, we encode object information in the format "[*category*].[*index*]: [*box*]" within the prompt. This representation helps the LLM correctly match visual regions with relationship triplets, particularly in cases where multiple instances of the same category appear in an image.

Additionally, the LLM is prompted to maintain logical consistency, thereby avoiding the generation of impossible or nonsensical relationships. Using this carefully crafted prompt, the LLM synthesizes scene graphs based on the provided image information. An example of this process is demonstrated in Table A.2, where GPT-4 successfully generates a comprehensive and accurate scene graph, even providing reasonable inferences for latent relationships not explicitly mentioned in the captions.

4.2.4 Training SGG Models

Following the generation of scene graphs, standard SGG models can be trained with the localized objects and their associated relationships. To accommodate closed-set training scenarios, a mapping process is required to align the extracted objects and relationships with the target category set. For instance, the object label "person" might be mapped to "people" if "person" is not explicitly defined within the target set. For open-vocabulary scenarios, this mapping step is not needed.

4.2.5 Instruction Tuning Private LLMs

We use Low-Rank Adaptation (LoRA) [44] to fine-tune a private and local LLM, Llama 2 [120], with the instruction-following data generated by GPT-4. This process involves adapting Llama 2 to better align with specific instruction sets, enhancing its ability to synthesize scene graphs from textual image data.

4.3 Experiments of GPT4SGG

4.3.1 Experimental Setup

VG150 [135] is a widely used benchmark for SGG, comprising 108,777 images with 150 object categories and 50 predicates. It is split into 70% for training, 5,000 for validation, and the remaining for testing, with $\sim 257k$ relationship triplets.

COCO Caption [14] consists of $\sim 117k$ images, each with five manual captions. Previous works [150, 152] used a scene parser [90] to extract $\sim 181k$ relationship triplets from these captions. We refer to this derived dataset as "**COCO@Parser**".

COCO@GPT is derived from COCO with annotated bounding boxes containing $\sim 94k$ images. The model card for generating COCO@GPT is *GPT-4 Turbo*. The COCO@GPT includes $\sim 394k$ instances of relationship triplets with 80 object categories and $\sim 4.7k$ predicate categories.

VG@GPT is derived from VG150 using GPT4SGG, containing $\sim 47k$ images. This dataset includes $\sim 227k$ instances of relationship triplets with 150 object categories and $\sim 2.4k$ predicate categories.

Detailed statistics for head-10 / tail-10 categories across VG@GPT, COCO@Parser, and COCO@GPT can be found in the Figure A.1. It can be found that COCO@Parser is biased to conjunction words (*e.g.*, "with", "in", "from"), while less focus on interactions like "holding", "wearing", *etc.* This defect makes it hard for the SGG model to learn complex interactions. Conversely, COCO@GPT offers rich instances of complex relationships.

Model settings. we mainly compare two SGG models with *GPT4SGG*: 1) VS^3 [150] extends the object detector of the SGG model from closed-set to open-vocabulary. 2) OvSGTR [18] advances the SGG from closed-set to both object and relation-aware open-vocabulary setting. For a fair comparison, we follow the official implementations

Table 4.1: Comparison with state-of-the-art methods on VG150 test set. ✧ marks fully supervised methods.

SGG model	Training Data	Grounding	R@20/50/100			mR@20/50/100		
LSWS [143]		-	-	3.28	3.69	-	-	-
MOTIFS [146]		Li *et al.* [73]	5.02	6.40	7.33	-	-	-
Uniter [15]		SGNLS [152]	-	5.80	6.70	-	-	-
Uniter [15]	COCO	Li *et al.* [73]	5.42	6.74	7.62	-	-	-
$VS^3_{(Swin\text{-}L)}$ [150]	Caption	GLIP-L [69]	5.26	6.70	7.91	1.97	2.38	2.70
$OvSGTR_{(Swin\text{-}B)}$ [18]		Grounding DINO [86]	6.85	9.33	11.47	1.28	1.79	2.18
$VS^3_{(Swin\text{-}L)}$ [150]	COCO@GPT	-	**5.07**	**7.40**	**9.50**	**1.30**	**1.93**	**2.42**
$OvSGTR_{(Swin\text{-}B)}$ [18]			**7.65**	**10.10**	**11.73**	**2.92**	**3.84**	**4.69**
IMP [135] ✧			17.73	25.48	30.71	2.66	4.10	5.29
MOTIFS [146] ✧	VG150		25.40	32.34	36.80	5.80	7.38	9.04
$VS^3_{(Swin\text{-}L)}$ [150] ✧	(∼76k)	-	27.34	36.04	40.88	4.43	6.45	7.81
$OvSGTR_{(Swin\text{-}B)}$ [18] ✧			27.80	36.40	42.40	5.24	7.41	8.98
IMP [135]		GLIP-L [69]	6.51	9.54	11.86	0.88	1.41	1.89
LSWS [143]		-	-	3.85	4.04	-	-	-
MOTIFS [146]	VG Caption	Li *et al.* [73]	8.25	10.50	11.98	-	-	-
MOTIFS [146]	(∼73k)	GLIP-L [69]	13.88	18.46	21.81	3.71	4.85	5.58
$VS^3_{(Swin\text{-}L)}$ [150]		GLIP-L [69]	11.31	16.00	19.85	2.39	3.80	4.87
$OvSGTR_{(Swin\text{-}B)}$ [18]		GLIP-L [69]	16.36	22.14	26.20	3.80	5.24	6.25
IMP [135]			**12.78**$_{+6.27}$	**16.63**$_{+7.09}$	**19.39**$_{+7.53}$	**3.86**$_{+2.98}$	**4.88**$_{+3.47}$	**5.76**$_{+3.87}$
MOTIFS [146]	VG@GPT	-	**16.41**$_{+2.53}$	**20.46**$_{+2.00}$	**23.23**$_{+1.42}$	**5.86**$_{+2.15}$	**7.36**$_{+2.51}$	**8.51**$_{+2.93}$
$VS^3_{(Swin\text{-}L)}$ [150]	(∼46k)		**17.77**$_{+6.46}$	**22.42**$_{+6.42}$	**25.29**$_{+5.44}$	**4.24**$_{+1.85}$	**5.82**$_{+2.02}$	**6.97**$_{+2.10}$
$OvSGTR_{(Swin\text{-}B)}$ [18]			**20.12**$_{+3.76}$	**25.03**$_{+2.89}$	**28.84**$_{+2.64}$	**5.68**$_{+1.88}$	**7.14**$_{+1.90}$	**8.22**$_{+1.97}$

to conduct experiments.

Metric. Following standard SGG models [146, 117, 70, 135, 150, 18], we use the protocol **SGDET** [135, 117] to measure the capability of SGG models. Recall@K (R@K, K=20 / 50 / 100) and mean Recall@K (mR@K, K=20 / 50 / 100) are reported. mR@K reflects the capability of SGG models to address long-tail bias. The SGDET protocol requires the model to detect objects and predicate relationships. A relationship triplet is regarded as correct if and only if the labels of the triplet are correct and the union of bounding boxes has an IoU score more than 0.5 over the union of ground truth boxes.

4.3.2 Experimental Results

Quantitative results. Table 4.1 reports a comparison with state-of-the-art models on VG150 test set. From the result, *GPT4SGG* significantly improves the performance of SGG models. For instance, OvSGTR [18] (Swin-B) with few images outperforms itself trained on VG caption (**25.03/28.84** vs. **22.14/26.20**, R@50/100). The improvement of mR@K also indicates that *GPT4SGG* alleviates the long-tail problem. This improvement is attributed to our novel approach can generate more accurate and comprehensive scene graphs.

It is also observed that the performance gain on the COCO dataset is less pronounced compared to VG150. The reason is twofold. First, COCO only has 80 categories for *GPT4SGG* and only 66 categories can be matched into VG150, resulting in worse recall of objects. Another reason is the distribution discrepancy between the two datasets, which is well-known from previous methods.

Figure A.2 reports Recall@100 per predicate of VS^3 and OvSGTR, respectively. The quantitative results are consistent with the statistics of head-10 / tail-10 categories. From the results, *GPT4SGG* outperforms parser-based methods except for conjunction words like "on" and "of". For actions, parser-based methods obtain a better recall of "wears" due to the lemmatization in parsing triplets (*e.g.*, "wearing" would be lemmatized as "wears"). Moreover, *GPT4SGG* surpasses fully-supervised methods in predicates like "riding", "near" and "part of".

Qualitative results. We provide samples generated by *GPT4SGG* and corresponding results generated by Scene Parser [90] and GPT-4 [90], as shown in Figure 4.4. From these samples, *GPT4SGG* can generate more accurate and complete scene graphs. The qualitative comparison also presents the ambiguity issue in grounding objects and the relationship density difference between ours and the previous pipeline.

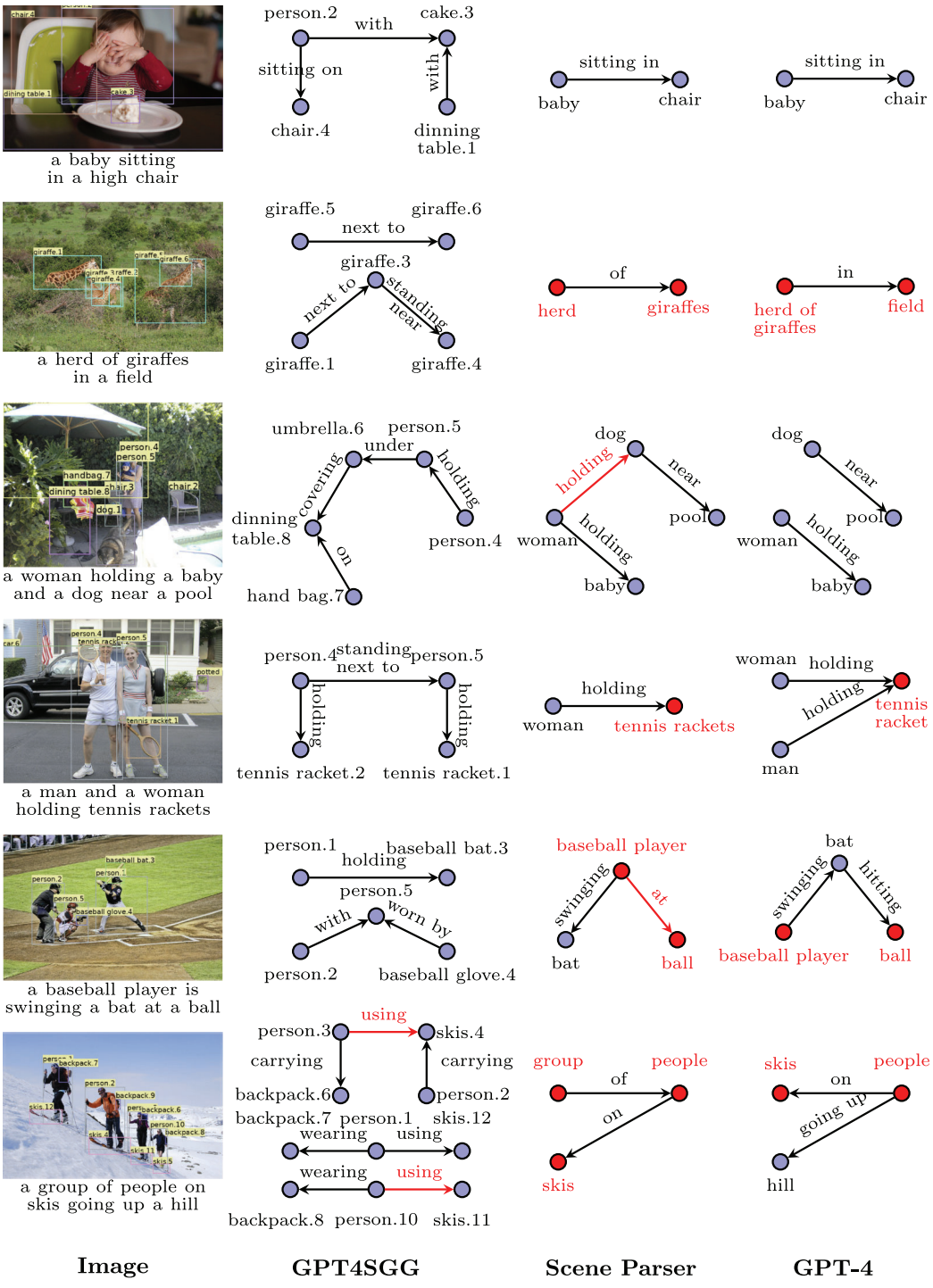


Figure 4.4: Samples from *GPT4SGG*, and corresponding triplets parsed by Scene Parser [90] and GPT-4 [97] (best view in color and zoom in). Red colors refer to incorrect edges or confusing nodes (with ambiguity), *e.g.*, "tennis rackets" in the second row. For clarity, we only list the **holistic narrative** for each image, which is the input of Scene Parser [90] (3-rd column) and GPT-4 [97] (4-th column).

Table 4.2: Ablation study of the effect of ambiguity in grounding. All models are trained on VG@Poly ($\sim 27k$ images) and tested on VG150 test set. Compared to manual annotation, *GPT4SGG* obtains a recall rate 13.4% on VG@Poly.

SGG model	Supervision	#Triplets	R@20/50/100			mR@20/50/100		
VS^3	Fully	157k	24.62	32.93	38.07	3.67	5.50	6.84
OvSGTR			27.46	35.98	41.37	5.04	6.78	7.97
VS^3	Parser + GLIP-L	386k	10.63	14.97	18.83	1.65	2.63	3.44
OvSGTR			15.63	20.58	24.20	3.33	4.48	5.38
VS^3	*GPT4SGG*	151k	**17.74**$_{+7.11}$	**22.27**$_{+7.30}$	**25.17**$_{+6.34}$	**4.07**$_{+2.42}$	**5.38**$_{+2.75}$	**6.48**$_{+3.04}$
OvSGTR			**19.19**$_{+3.56}$	**23.96**$_{+3.38}$	**27.51**$_{+3.31}$	**5.32**$_{+1.99}$	**6.77**$_{+2.29}$	**7.80**$_{+2.42}$

4.3.3 Ablation Studies

How close is GPT4SGG to manual annotation? To verify the quality of generated data by *GPT4SGG*, we make a comparison on the VG150 validation set. From the experimental result, the original *GPT4SGG* achieves a recall rate **11.43%**, and *w. lookup* (builds a lookup table by leveraging GPT-4 for semantic matching as illustrated in Table A.3) works better than *w. WordNet* (utilizes WordNet [91] synsets matching) (**12.09%** vs. **11.70%**); as there are many complicate relationships, *e.g.*, "*standing next to*" can be mapped into "*near*".

Impact of object detection/grounding. This study examines the effect of object detection accuracy on *GPT4SGG*'s performance. We compare scene graphs generated using manually annotated object bounding boxes against those generated using predictions from a state-of-the-art object detector, Grounding DINO (w. Swin-B) [86], which achieves 22.2 AP50 on the VG150 validation set. Using detector predictions instead of manual annotations leads to a significant drop in GPT4SGG's (w. GPT-3.5-turbo) recall from 7.80% to 3.11%. This drop is largely attributed to the object detector's limited recall of 56.8% (at IoU threshold 0.5), underscoring the importance of accurate object localization for *GPT4SGG*'s performance.

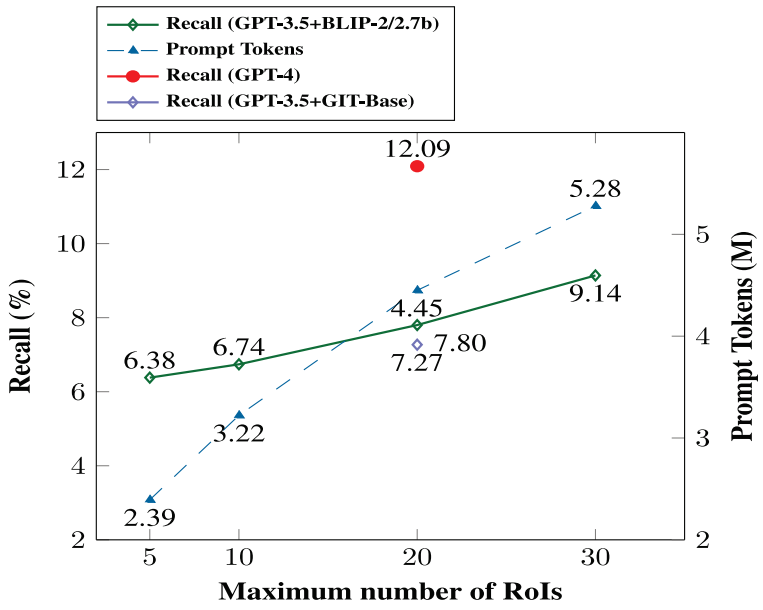


Figure 4.5: Impact of Caption. Considering the cost of ChatGPT APIs, we choose *GPT-3.5-turbo* as the main LLM to probe the performance under different settings.

Impact of Ambiguity in grounding. To highlight the challenges posed by grounding ambiguity, we select a subset of 27k images from the VG150 training set, termed "VG@Poly", each containing multiple instances of the same object category. This subset specifically targets scenarios where aligning visual objects with language descriptions is inherently ambiguous.

Table 4.2 presents the performance of SGG models trained on VG@Poly. A significant performance drop is observed when comparing models trained with ambiguous grounding (Row 2) to those trained with full supervision (Row 1), underscoring the limitations of traditional pipelines. By contrast, *GPT4SGG* significantly improving performance in the presence of grounding ambiguity. Notably, both VS^3 and OvSGTR achieve comparable or even superior mR@K scores when trained with *GPT4SGG* compared to full supervision, indicating that *GPT4SGG* effectively handles the ambiguity issue in object grounding.

Impact of Caption. This study examines the effects of using different caption-

Table 4.3: Comparison of LLMs in synthesizing scene graphs.

	Model	Fine-tuned	Recall (%)	Valid Responses
	LlaVA v1.5-13B	✗	4.74	4,004
	LlaVA v1.6-13B	✗	6.26	3,152
GPT4SGG	GPT-3.5 turbo	✗	7.80	4,537
GPT4SGG	GPT-4 turbo	✗	**12.09**	**4,760**
GPT4SGG	Llama 2-13B	✗	8.02	3,876
GPT4SGG	Llama 2-13B	✓	**11.93**	**4,718**

ing models and varying the maximum number of region descriptions. As expected (see Figure 4.5), increasing the maximum number of regions, thereby covering more ground-truth edges, leads to improved recall performance. Considering the trade-off between prompt token usage and performance, we selected a maximum number of N=20 for generating data with GPT-4. Additionally, we compared two captioning models: BLIP-2/opt-2.7b [67] and GIT-Base [126]. Using GPT-3.5, BLIP-2 achieved better results than GIT-Base (**7.80%** vs. **7.27%**).

Source of LLM. This study compares the capability of different LLMs, including GPT-3.5 [96], GPT-4 [97], and Llama 2 [120], to synthesize scene graphs. The results are summarized in the Table 4.3.

Without fine-tuning, *GPT-3.5 turbo* achieves a recall of **7.80%** on the VG150 validation set (4,808 images) with 4,537 valid responses. Llama 2-13B [120] obtains a recall rate of **8.02%** but with only 3,876 valid responses. *GPT-4 turbo* achieves the highest recall of **12.09%** with 4,760 valid responses. The results show that Llama 2 has the worst instruction-following capability among the three LLMs.

However, the fine-tuned Llama 2-13B [120] achieves a recall rate of **11.93%** on the VG150 validation set. This indicates that the instruction-tuned Llama 2 [120] can achieve performance comparable to GPT-4 [97], providing a cost-effective and private

Table 4.4: Performance of the multiple-choice QA task. "Acc." refers to the overall accuracy across all questions, while "mAcc." represents the average accuracy per image. LlaVA v1.5-13B-LoRA is fine-tuned on the VG150 training set with ground-truths, while *GPT4SGG* is fine-tuned on data generated by GPT-4 turbo.

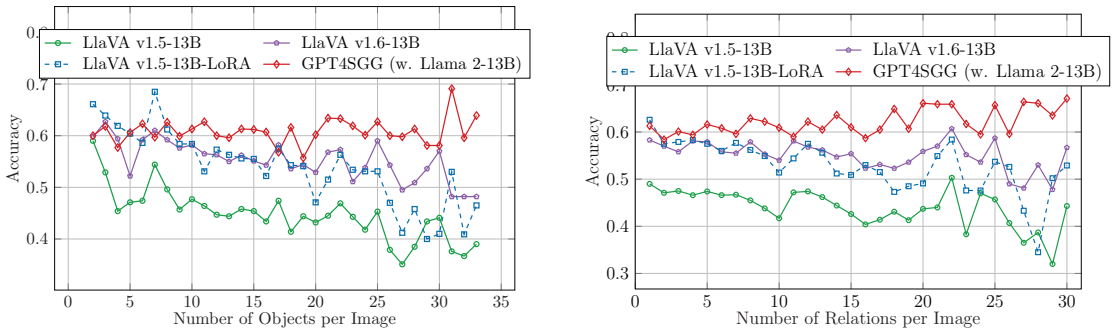
	Model	Fine-tuned	Acc. (%)	mAcc. (%)
	Random Guess	-	25.0	25.0
Closed-source	GPT-4o	✗	72.8	72.0
Open-source	LlaVA v1.5-13B	✗	44.4	46.3
	LlaVA v1.5-13B-LoRA	✓	54.2	57.0
	LlaVA v1.6-13B	✗	55.4	56.5
	GPT4SGG (w. Llama 2-13B)	✗	49.3	49.3
	GPT4SGG (w. Llama 2-13B)	✓	**61.4**	**60.7**

option for the LLM in *GPT4SGG*.

LLM vs. multi-modal LLM. While end-to-end multi-modal LLMs like LlaVA hold promise for scene understanding, our experiments reveal limitations in their direct application to SGG. As shown in Table 4.3, LlaVA v1.5-13B achieves a recall rate of only 4.74% on the VG150 validation set, significantly lower than *GPT4SGG*'s 11.93% recall rate. This gap suggests that despite access to visual information, capturing relationships within complex scenes remains challenging for current multi-modal LLMs.

To further probe this disparity, we designed a multiple-choice QA task using the VG150 validation set, consisting of 31,951 questions about relationships between localized objects. Table 4.4 compares the performance of LlaVA, GPT-4o, and GPT4SGG on this task. While the closed-source GPT-4o demonstrates a strong understanding of visual relationships, reaching 72.8% accuracy, *LlaVA struggles to grasp visual relationships, even when fine-tuned with ground-truth data (LlaVA v1.5-13B-LoRA).*

Furthermore, *GPT4SGG demonstrates superior robustness to scene complexity.* Figure 4.6a and Figure 4.6b show the accuracy trend relative to the number of objects / relations per image, respectively. *As the scene graph becomes more complex (i.e.,*



(a) Accuracy vs. Number of Objects per Image.

(b) Accuracy vs. Number of Relations per Image.

Figure 4.6: Accuracy vs. Number of Objects and Relations per Image

with more nodes or edges), LlaVA's performance declines more dramatically, while GPT4SGG *maintains consistently high accuracy.* This robustness can be attributed to *GPT4SGG*'s strategy of decoupling visual feature extraction and relationship reasoning, potentially mitigating the negative impacts of visual ambiguity and hallucination often observed in multi-modal LLMs [3]. Notably, even with improvements introduced in LlaVA v1.6 [81] like utilizing multiple image resolutions, *GPT4SGG* maintains its superior performance in handling complex scenes.

Comparison with GPT-4V. Due to the high cost of the GPT-4V API (https://openai.com/pricing), we evaluated it on a random sample of 100 images from the VG150 validation set, with results shown in Table 4.5. Each image was resized to a maximum dimension of 512 pixels, and object details, including names and bounding boxes, were provided. While GPT-4V achieves higher recall due to direct image input, it operates as a closed, non-customizable model. In contrast, *GPT4SGG* offers comparable recall performance and can be customized with local LLMs, making it a flexible, cost-effective solution for scene graph generation.

Table 4.5: Comparison of GPT4SGG and GPT-4V on 100 images from the VG150 validation set.

Model	#Triplets (GT/Generated)	Recall (original/*w.* lookup)	Token Usage (Context/Generated)	Cost (USD)
GPT4SGG	558/610	17.5/18.1	91k/17k	~1.4
GPT-4V	558/1145	17.6/19.6	56k/30k	~1.5

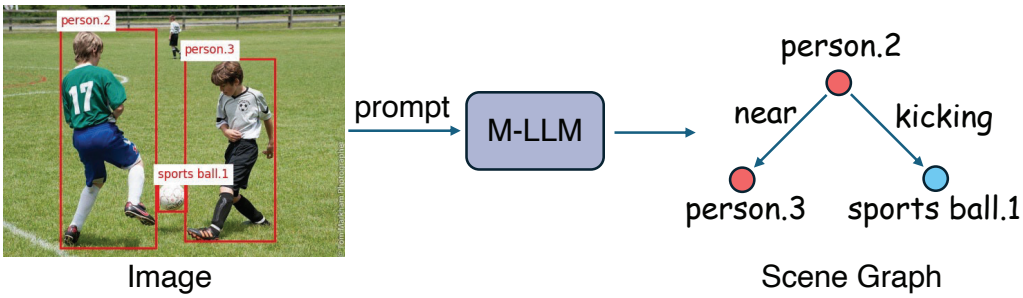
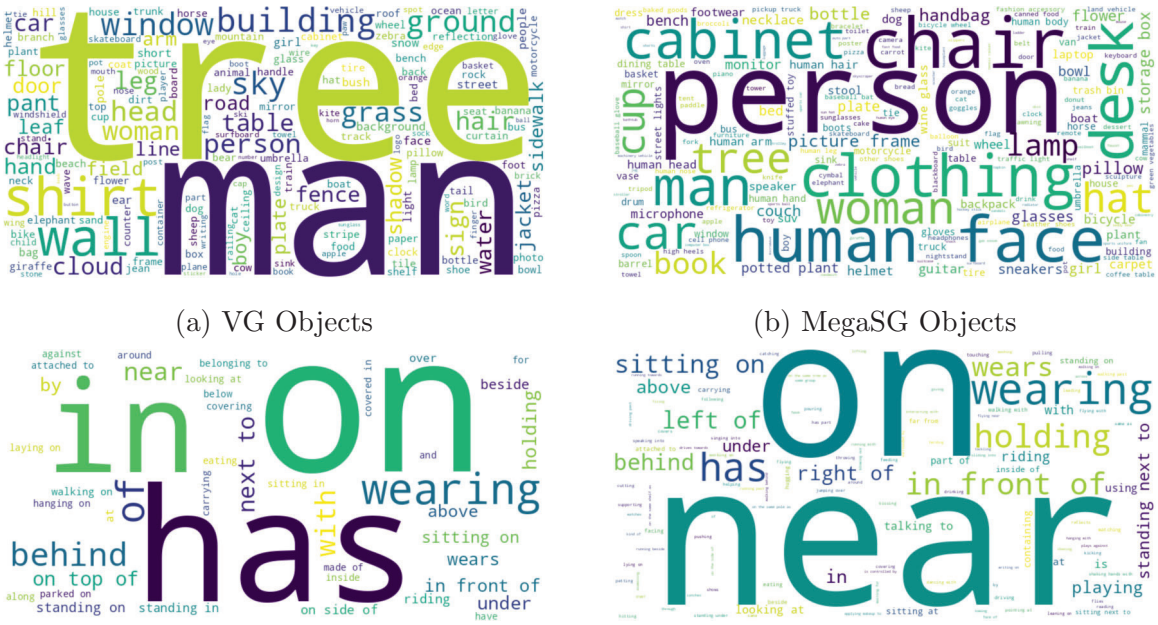


Figure 4.7: Pipeline for Constructing *MegaSG*. The multimodal LLM (M-LLM) synthesizes a scene graph from grounded objects in the image guided by a prompt.

4.4 MegaSG: A Large-scale Scene Graph Dataset

GPT4SGG demonstrates the potential of language-only LLMs in enabling weakly supervised SGG. However, their inability to directly process visual information limits the completeness and accuracy of the generated scene graphs. With the advent of multimodal LLMs, this limitation is fundamentally transformed. In this section, we introduce *MegaSG*, a large-scale scene graph dataset generated by a multimodal LLM without any fine-tuning.

The pipeline of constructing *MegaSG* is shown in Figure 4.7. We construct the *MegaSG* dataset by leveraging three widely used object detection datasets: COCO [14], Object365 [111], and Open Images v6 [60]. These resources offer diverse scenes with meticulously annotated objects. To ensure the reliability of the scene graphs, we discard images containing fewer than three objects. The annotation prompt used to



(a) VG Objects

(b) MegaSG Objects

(c) VG Relationships

(d) MegaSG Relationships

Figure 4.8: Word clouds of objects and relationships in the Visual Genome (VG) and MegaSG datasets. (a) and (b) illustrate the diversity of objects, while (c) and (d) highlight the relationships. The comparison demonstrates MegaSG's broader vocabulary and richer representation of object-relationship semantics.

generate scene graphs from images is detailed in Algorithm 2 in the Appendix.

Specifically, the multimodal large language model receives an image—along with associated object categories and bounding boxes—as input and produces a scene graph that depicts the relationships between objects. For example, as illustrated in Figure 4.7, the model identifies a "person" kicking a "sports ball" and another "person" nearby, yielding the scene graph: {*"source": "person.2", "target": "sports ball.1", "relation": "kicking"*}, {*"source": "person.2", "target": "person.3", "relation": "near"*}. This structured representation captures both the objects and their spatial as well as relational interactions.

To quantitatively assess the quality of *MegaSG*, we evaluate state-of-the-art Scene

Graph Generation (SGG) models trained on different datasets. For instance, the OvSGTR (Swin-B) [17] model trained on MegaSG achieves a zero-shot performance recall of 45.71% (R@50, PredCls mode) on the VG150 test set, outperforming models trained on smaller datasets like COCO Caption data (see Table 3.1 in Chapter 3). This improvement reflects the high-quality scene graph annotations of MegaSG, enabling further exploration of training SGG models on MegaSG or generating images from scene graphs on MegaSG.

We compare the word cloud of VG and MegaSG in Figure 4.8. From the comparison in the word clouds, both the VG and MegaSG datasets contain similar high-frequency objects like "person", "tree", and "man", as well as common relationships such as "on" and "near". However, the MegaSG dataset shows a wider variety of object types and relationship terms, suggesting it captures a wider range of visual semantics than the VG dataset.

4.5 Summary

This chapter discussed weakly-supervised Scene Graph Generation (SGG) with language models. We presents *GPT4SGG*, a novel framework for synthesizing scene graphs from holistic and region-specific textual descriptions, addressing key limitations in traditional language-supervised SGG, such as ambiguity in object grounding and sparse relational coverage. By decomposing images into regions and leveraging large language models (LLMs) for contextual reasoning, *GPT4SGG* generates accurate and diverse scene graphs without relying on manual annotations.

Building on the success of GPT4SGG, we further introduce *MegaSG*, a large-scale scene graph dataset constructed using multimodal LLMs. MegaSG enhances visual grounding by combining visual prompts with object bounding boxes, enabling dense and grounded scene graph synthesis at scale. This dataset serves as a valuable re-

source for pre-training and benchmarking open-vocabulary SGG models, offering significantly broader object and relation coverage than existing datasets.

Experimental results demonstrate that both *GPT4SGG* and *MegaSG* substantially improve SGG model performance across multiple metrics and settings. Together, they provide a scalable and generalizable pipeline for weakly-supervised scene graph generation, pushing the boundary of open-vocabulary and multimodal understanding.

Chapter 5

Reinforcement Learning for Scene Graph Generation

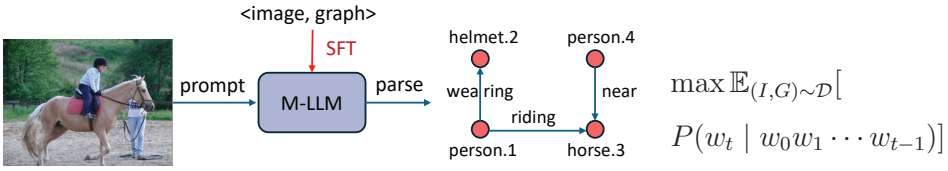
In Chapter 4, we explored how large language models (LLMs) can contribute to scene graph generation, without requiring model fine-tuning. Frameworks such as *GPT4SGG* employ a language-only LLM to infer relationships from region-level captions, while *MegaSG* leverages a multimodal LLM to directly synthesize large-scale scene graphs from grounded object inputs. Although these prompting-based approaches achieve notable gains in recall and scalability, they still rely on strong priors (*e.g.*, grounded objects) and lack mechanisms for directly optimizing graph-level structural correctness. To address these limitations, this chapter introduces reinforcement learning (RL) with rule-based rewards as a paradigm to fine-tune multimodal LLMs for structured prediction, aligning generation with relational consistency and downstream performance.

5.1 Introduction

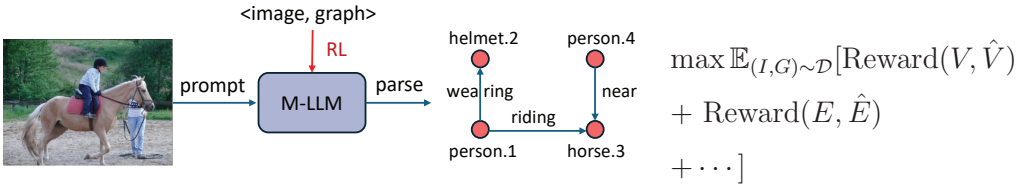
To generate scene graphs from an image, traditional Scene Graph Generation (SGG) models [52, 135, 74, 146, 118, 13, 70, 56, 23, 150, 17] decouple the task into two subtasks, *i.e.*, object detection and visual relationship recognition, and directly maximize the likelihood of the ground-truth labels given the image. Essentially, these models tend to overfit the distribution of annotated datasets; Consequently, they struggle to handle long-tail distributions and are prone to generating biased scene graphs (*e.g.*, all predicted relationships are head classes like "on" and "of").

While traditional SGG models rely on manual annotated datasets and struggle to generalize to new domains, recent advances in large languge models (LLMs) offer a new paragdim. LLM4SGG [57] utilizes an LLM to extract relationship triplets from captions using both original and paraphrased text, while GPT4SGG [19] employs an LLM to synthesize scene graphs from dense region captions. Additionally, Li [71] generates scene graphs via image-to-text generation using vision-language models (VLMs). These weakly supervised methods demonstrate potential for generating scene graphs with little or no human annotation but suffer from accuracy issues in the generated results.

Despite these advancements, existing methods typically employ text-only LLMs or rely on intermediate captions as input, which do not fully leverage the rich visual context. In contrast, multimodal large language models (M-LLMs) which integrate both visual and linguistic modalities offer the potential for more direct and holistic scene understanding. By processing visual information alongside natural language prompts, M-LLMs can generate scene graphs in an end-to-end manner. However, in practice, M-LLMs suffer from instruction following (*e.g.*, the output does not contain "objects" or "relationships"), repeated response (*e.g.*, {*"objects":[*$\cdots$ {*"id": "desk.9", "bbox": [214, 326, 499, 389]*}, {*"id": "desk.10", "bbox": [214, 326, 499, 389]*}, {*"id": "desk.11", "bbox": [214, 326, 499, 389]*}, $\cdots$ *]*$\cdots$}), inaccurate location, *etc.* These



(a) M-LLM with SFT is optimized token by token (here, w_i refers to a token).



(b) M-LLM with RL is optimized using rule-based rewards. Here, $G = (V, E)$ and $\hat{G} = (\hat{V}, \hat{E})$ refer to the ground-truth and predicted scene graphs, respectively.

Figure 5.1: Comparison of multimodal LLMs (M-LLMs) fine-tuned via Supervised Fine-tuning (SFT) and Reinforcement Learning (RL) for Scene Graph Generation (SGG).

challenges highlight the need for better alignment between visual understanding and structured representation within the M-LLM framework.

To improve instruction-following and structured output generation in M-LLMs, one intuitive solution is to perform Supervised Fine-tuning (SFT) on scene graph datasets (see Figure 5.1-(a)). In the context of SGG, SFT aligns the model's outputs with expected formats (*e.g.*, structured lists of objects and relationships) by training it on high-quality scene graph annotations. This process encourages the model not only to recognize entities and relations from the image but also to organize them into a coherent and valid graph structure. Nevertheless, SFT alone still be insufficient as all output tokens are weighted equally in the loss. For example, the experimental results on the VG150 dataset [135] reveal that even with SFT, M-LLM still has a high failure rate to generate a valid and high-quality scene graph. The drawback of SFT in SGG lies in the lack of effective signals to correct the output (*e.g.*, the model cannot directly utilize the Intersection over Union (IoU) between the predicted box

and the ground truth to refine its output).

To advance M-LLMs for effective Scene Graph Generation (SGG), we propose *R1-SGG*, a novel framework leveraging visual instruction tuning enhanced by reinforcement learning (RL). The visual instruction tuning stage follows a conventional supervised fine-tuning (SFT) paradigm, *i.e.*, fine-tuning the model using prompt-response pairs with a cross-entropy loss. For the RL stage, we adopt GRPO, an online policy optimization algorithm introduced in DeepSeekMath [112].

To enable effective reinforcement learning for Scene Graph Generation, we introduce a set of rule-based, graph-centric rewards that reflect the structural characteristics of scene graphs. Given an image and a prompt, a multimodal large language model (M-LLM) generates a set of objects and relational triplets. To evaluate and optimize these predictions, we formulate reward functions aligned with standard SGDET metrics [135] and structured reasoning objectives. Specifically, we define three reward variants: **Hard Recall**, which counts a triplet as correct only if the subject, predicate, and object labels exactly match the ground truth and both bounding boxes achieve IoU ≥ 0.5; **Hard Recall+Relax**, which relaxes the exact match constraint by incorporating embedding similarity between predicted and ground-truth labels; and **Soft Recall**, which further densifies reward signals via bipartite matching, combining object label similarity, IoU, and bounding box distance into a unified cost function. These scene graph rewards are computed over matched object and edge pairs, and are complemented by a format reward that enforces structural adherence in output formatting. This reward design enables stable and fine-grained policy optimization using GRPO, guiding the M-LLM toward generating accurate, complete, and structurally valid scene graphs.

Our contributions can be summarized as follows:

- We explore how to develop a multimodal LLM for Scene Graph Generation (SGG), by leveraging visual instruction tuning with reinforcement learning

(RL). To the best of our knowledge, this is among the earliest studies to develop an advanced multimodal LLM for end-to-end scene graph generation.

- Graph-centric, rule-based rewards are designed to guide policy optimization in a manner aligned with standard evaluation metrics in SGG, such as the recall of relationship triplets—metrics that cannot be directly optimized through SFT.
- Experimental results demonstrate that the proposed framework improves the ability to understand and reason about scene graphs for multimodal LLMs.

5.2 R1-SGG: Multimodal LLM with GRPO

5.2.1 Preliminary

Reinforcement Learning with GRPO. Group Relative Policy Optimization (GRPO) is a online reinforcement learning algorithm introduced by DeepSeekMath [112]. Unlike traditional methods such as PPO [110], which require an explicit critic network, GRPO instead compares groups of candiates to update the policy π_θ. Specifically, for each input query q, a set of candidate outputs $\{o_i\}_{i=1}^{G}$ is drawn from the previous policy $\pi^{\text{old}}(O|q)$, and the advantage of each candidate is computed relative to the group's average reward:

$$A_i = \frac{r_i - \text{mean}(\{r_1, \ldots, r_G\})}{\text{std}(\{r_1, \ldots, r_G\})}. \tag{5.1}$$

The policy parameters θ are updated by maximizing the following GRPO objective:

$$\begin{aligned} J_{\text{GRPO}}(\theta) = \mathbb{E}_{q\sim P(Q),\,\{o_i\}_{i=1}^{G}\sim\pi^{\text{old}}(O|q)} \Bigg[\frac{1}{G}\sum_{i=1}^{G} \Bigg(\min\Bigg(\frac{\pi_\theta(o_i|q)}{\pi^{\text{old}}(o_i|q)} A_i, \\ \text{clip}\Big(\frac{\pi_\theta(o_i|q)}{\pi^{\text{old}}(o_i|q)},\, 1-\epsilon,\, 1+\epsilon \Big) A_i \Bigg) - \beta\, D_{\text{KL}}\big(\pi_\theta \parallel \pi_{\text{ref}}\big) \Bigg], \end{aligned} \tag{5.2}$$

Here, ϵ and β are hyper-parameters. The first term uses a clipped probability ratio (as in PPO) to control the update magnitude, while the KL divergence regularizer

$D_{\mathrm{KL}}(\pi_\theta \parallel \pi_{\mathrm{ref}})$ constrains the new policy π_θ to not deviate too much from a reference policy π_{ref}. This formulation, which combines a group-relative advantage, a clipping mechanism, and a KL divergence regularizer, stabilizes policy updates and improves training efficiency, demonstrating remarkable potential for enhancing the reasoning performance of LLMs such as DeepSeek R1 [39].

5.2.2 Overview of R1-SGG

R1-SGG is a reinforcement learning framework that enhances scene graph generation (SGG) in multimodal large language models (M-LLMs). It builds on a supervised fine-tuning (SFT) stage using prompt-response pairs, followed by reinforcement learning (RL) with structured, graph-centric rewards.

Given an input image and prompt, the M-LLM generates a scene graph $\mathcal{G}_{\mathrm{pred}} = (\mathcal{V}_{\mathrm{pred}}, \mathcal{E}_{\mathrm{pred}})$, comprising objects (nodes) and their relationships (edges). We primarily optimize using *Hard Recall*, which aligns with SGDET metrics by rewarding exact triplet matches. To study the sparsity and design of the rewards, we also evaluated relaxed alternatives based on bipartite matching between $\mathcal{G}_{\mathrm{pred}}$ and the ground truth graph $\mathcal{G}_{\mathrm{gt}}$, allowing fine-grained node and edge rewards. Our RL pipeline employs Group Relative Policy Optimization (GRPO) [112], which compares sampled outputs and promotes higher-reward candidates. By integrating SFT, GRPO, and graph-aware rewards, R1-SGG enables M-LLMs to generate accurate, diverse, and structurally valid scene graphs.

5.2.3 Rewards Definition

Format Reward. Following DeepSeek R1 [39], we employ a format reward to ensure the model's response adheres to the expected structure, specifically *<think>*$\cdots$ *</think>* *<answer>*$\cdots$ *</answer>*. A reward of 1 is assigned if the response follows

this format and the segment enclosed by `<answer>···</answer>` contains both the keywords `object` and `relationships`; otherwise, the reward is 0.

Scene Graph Rewards. Standard evaluation protocols for Scene Graph Generation (SGG), such as SGDET [135], formulate the task as a recall-oriented problem, emphasizing the model's ability to retrieve correct relationship triplets from an image. To investigate the impact of different reward formulations, we introduce three variants: *Hard Recall*, *Hard Recall+Relax*, and *Soft Recall*.

Hard Recall. To align policy optimization with standard SGDET metrics, we define *Hard Recall*, where a predicted triplet $\langle \texttt{subject}, \texttt{predicate}, \texttt{object} \rangle$ is counted as a true positive when *both* of the following hold: 1) *Triplet accuracy:* the subject, predicate, and object labels exactly match the ground truth. 2) *Localization accuracy:* the IoU between predicted and ground-truth bounding boxes exceeds 0.5. This reward is aligned with standard metrics but suffers from sparsity due to its strict criteria.

Hard Recall + Relax. We relax the triplet accuracy requirement by computing cosine similarity between the entity embeddings of predicted and ground-truth triplets. This softens the discrete matching constraint to provide more gradient signal.

Soft Recall. We further propose a dense matching reward by formulating it as a bipartite matching problem, similar to DETR [8], where predicted nodes $\{v_i = (c_i, b_i)\}_{i=1}^{M}$ (each node v_i is comprising an object class c_i and a bounding box b_i) are matched to ground-truth nodes $\{\tilde{v}_j = (\tilde{c}_j, \tilde{b}_j)\}_{j=1}^{N}$ with the following cost:

$$\begin{aligned} \text{cost}(v_i, \tilde{v}_j) =& \lambda_1 \cdot (1.0 - \langle \texttt{Embedding}(c_i), \texttt{Embedding}(\tilde{c}_j) \rangle) \\ &+ \lambda_2 \cdot (1.0 - \text{IoU}(b_i, \tilde{b}_j)) + \lambda_3 \cdot ||b_i - \tilde{b}_j||_1, \end{aligned} \tag{5.3}$$

where $\langle \cdot, \cdot \rangle$ denotes cosine similarity, λ_1, λ_2 are weight factors, and `Embedding` is obtained via the NLP tool SpaCy. By solving the bipartite matching problem, we establish a one-to-one node matching between the predicted graph $\mathcal{G}_{\text{pred}}$ and the ground-truth graph $\mathcal{G}$.

For a predicted node v_i, the reward is defined as

$$R(v_i) = \begin{cases} \lambda_1 \cdot \langle \texttt{Embedding}(c_i), \texttt{Embedding}(\tilde{c}_j) \rangle & \\ \quad +\lambda_2 \cdot \text{IoU}(b_i, \tilde{b}_j) & \\ \quad +\lambda_3 \cdot \exp(-||b_i - \tilde{b}_j||_1), & \text{if } v_i \text{ and } \tilde{v}_j \text{ are matched,} \\ 0, & \text{otherwise.} \end{cases} \tag{5.4}$$

which is the linear combination of object category similarity and the IoU of bounding boxes. The total rewards of an image's prediction set $\{v_i\}_{i=1}^{M}$ is computed as

$$R(\{v_i\}_{i=1}^{M}) = \frac{1}{|\mathcal{V}_{\text{gt}}|} \sum_{i=1}^{M} R(v_i). \tag{5.5}$$

For a predicted triplet $e_{ij} :=< v_i, p_{ij}, v_j >$, the reward is defined as

$$R(e_{ij}) = \begin{cases} \langle \texttt{Embedding}(v_i), \texttt{Embedding}(\tilde{v}_k) \rangle \cdot & \\ \langle \texttt{Embedding}(v_j), \texttt{Embedding}(\tilde{v}_l) \rangle \cdot & \\ \langle \texttt{Embedding}(p_{ij}), \texttt{Embedding}(p_{kl}) \rangle, & \text{if } v_i \text{ matches } \tilde{v}_k \\ & \quad \text{and } v_j \text{ matches } \tilde{v}_l, \\ 0, & \text{otherwise.} \end{cases} \tag{5.6}$$

Thereby, the reward of an image's predicted edge set is computed as

$$R(\{e_{ij}\}) = \frac{1}{|\mathcal{E}_{\text{gt}}|} \sum R(e_{ij}). \tag{5.7}$$

5.3 Experimental Results

5.3.1 Experimental Setup

Dataset. The widely-used scene graph dataset VG150 [135] consists of 150 object categories and 50 relation categories. Following prior works [150, 17], the training set

used in this work contains 56,224 image-graph pairs, while the validation set includes 5,000 pairs. To prompt the M-LLM, we transform each image-graph pair using the template described in Table B.2.

The Panoptic Scene Graph (PSG) dataset [138] is built on the COCO dataset [77], consisting of 80 *thing* object categories, 53 *stuff* object categories, and 56 relation categories. It contains 46,563 image-graph pairs for training and 2,186 pairs for testing.

Evaluation. Following the standard evaluation pipeline in SGG, we adopt the SGDET protocol [135, 117] to measure the model's ability to generate scene graphs. SGDET requires the model to generate scene graphs directly from the image without any predefined object boxes. Performance is evaluated using Recall and mean Recall (mRecall). Recall is computed for each image-graph pair, where a predicted triplet is considered correct if both the subject and object bounding boxes have an Intersection over Union (IoU) of at least 0.5 with the corresponding ground-truth boxes, and the subject category, object category, and relationship label all match the ground truth. Mean Recall (mRecall) is obtained by averaging the Recall across all relation categories. We additionally report AP@50 to assess object detection performance and Failure Rate to evaluate format consistency.

Implementation Details. Our code is based on the `trl` library [124] and utilizes vLLM [61] to speed up sampling during reinforcement learning. For SFT, the model is trained for 3 epochs with a batch size of 128 on 4 NVIDIA A100 (80GB) GPUs, using the AdamW optimizer [88] with a maximum learning rate of 1e-5. For RL, the model is trained for 1 epoch with a batch size of 32 and 8 generations per sample on 16 NVIDIA GH200 (120GB) GPUs, also using AdamW with a maximum learning rate of 6e-7. Our code is available at https://github.com/gpt4vision/R1-SGG/.

Table 5.1: Prompt template used for the visual relationship QA task.

Prompt = "Analyze the relationship between the object {sub_name} at {sub_box} and the object {obj_name} at {obj_box} in an image of size ({width} x {height}). The bounding boxes are in [x1, y1, x2, y2] format. Choose the most appropriate relationship from the following options: A) {choices[0]}; B) {choices[1]}; C) {choices[2]}; D) {choices[3]}.''

5.3.2 How Well Do M-LLMs Reason About Visual Relationships?

We evaluate the visual relationship reasoning capabilities of open-source multimodal LLMs using a four-to-one Visual Question Answering (VQA) task. Each model is prompted with an image and a corresponding question. The used prompt template is shown in Table 5.1. We report Acc (accuracy over all questions) and mAcc (mean accuracy per image) in Table 5.2. The results reveal that many multimodal LLMs struggle with visual relationship reasoning. Moreover, the task exhibits a noticeable text bias, and the presence of bounding boxes can sometimes mislead the model's attention. As a simpler task compared to SGG, the poor performance suggests that directly applying multimodal LLMs to SGG may yield suboptimal results.

5.3.3 How Well Do M-LLMs Generate Scene Graphs?

Benchmark on VG150

We report the performance under various settings in Table 5.3, which includes: 1) *Specific Models*: Methods built on specific detectors such as Faster R-CNN [105] (e.g., IMP [135]) or DETR [8] (e.g., OvSGTR [17]) for scene graph generation. 2) *Commercial M-LLMs*: Advanced multimodal large language models such as GPT-

Table 5.2: Comparison of VQA on the VG150 validation set across various models and settings. Gains compared to the *Original Image* (1st row) are indicated in red. *"mask img."* refers to masking the entire image with random noise, *"mask obj."* refers to masking object regions with black pixels, *"w/o cats."* refers to not providing object categories in the prompt, and *"w/o box."* refers to not providing bounding boxes in the prompt.

	InstructBLIP 7B		**LLaVA v1.5 7B**		**LLaVA v1.6 7B**		**Qwen2VL 7B**	
	Acc	**mAcc**	**Acc**	**mAcc**	**Acc**	**mAcc**	**Acc**	**mAcc**
org. img.	2.3	1.9	45.8	45.6	28.7	29.2	53.7	53.4
mask img.	1.0 (-1.3)	1.0 (-0.9)	21.8 (-24.0)	21.6 (-24.0)	3.9 (-24.8)	4.0 (-25.2)	0.0 (-53.7)	0.0 (-53.4)
mask obj.	1.9 (-0.4)	1.9 (-0.1)	37.2 (-8.6)	37.2 (-8.4)	12.8 (-15.9)	13.2 (-16.0)	16.2 (-37.5)	16.8 (-36.5)
w/o cats.	2.5 (+0.2)	2.4 (+0.4)	32.8 (-12.9)	32.7 (-12.9)	9.5 (-19.2)	10.1 (-19.1)	16.8 (-36.9)	18.1 (-35.3)
+ mask img.	1.0 (-1.3)	1.0 (-0.9)	15.4 (-30.3)	15.3 (-30.3)	0.0 (-28.7)	0.0 (-29.2)	0.2 (-53.6)	0.2 (-53.1)
+ mask obj.	1.8 (-0.5)	1.7 (-0.3)	27.9 (-17.8)	28.4 (-17.2)	3.3 (-25.4)	3.8 (-25.4)	4.7 (-49.1)	5.5 (-47.8)
w/o box.	26.0 (+23.7)	25.9 (+24.0)	61.9 (+16.2)	61.3 (+15.7)	53.5 (+24.8)	52.1 (+22.9)	78.1 (+24.4)	77.1 (+23.8)
+ mask img.	10.1 (+7.9)	10.2 (+8.2)	36.3 (-9.5)	35.2 (-10.4)	11.5 (-17.2)	11.4 (-17.7)	0.0 (-53.7)	0.0 (-53.4)
+ mask obj.	19.3 (+17.0)	19.1 (+17.1)	54.2 (+8.5)	53.8 (+8.2)	33.5 (+4.8)	33.2 (+4.1)	40.3 (-13.4)	39.3 (-14.1)

4o [47] and Gemini 1.5 Flash [104]. 3) *Open-source M-LLMs*: Publicly available models such as LLaVA v1.5 [80], Qwen2-VL [129], and our proposed *R1-SGG-Zero* (based on `Qwen2-VL-2B/7B-Instruct`, trained with GRPO but without supervised fine-tuning) and *R1-SGG* (built on the same backbone, fine-tuned with GRPO and initialized from SFT checkpoints).

The results in Table 5.3 reveal several key observations.

Zero-shot Performance of M-LLMs. Either commercial or open-source multimodal LLMs struggle to generate accurate scene graphs, and this can be attributed to several factors. First, the internal processing of private models such as GPT-4o remains opaque to users, resulting in suboptimal object detection performance. Second, models like LLaVA v1.5 align visual and textual features only at the image level, typically using a fixed resolution of 336×336, which restricts spatial understanding.

Table 5.3: SGDET performance on the VG150 validation set. For M-LLMs, predefined object classes and relation categories are included in the input prompts.

Method	Params	Failure Rate (%)	AP@50	Recall	mRecall
Specific Models					
IMP [135]	-	-	20.91	17.85	2.66
MOTIFS [146]			29.56	27.21	7.84
VCTree [118]			28.13	24.87	8.47
OvSGTR [17]			33.39	26.74	5.83
Commercial M-LLMs					
GPT-4o [47]	-	2.94	0.00	0.00	0.00
Gemini 1.5 Flash [104]	-	1.10	0.51	0.10	0.08
Gemini 2.0 Flash [36]	-	1.06	0.54	0.07	0.03
Open-sourced M-LLMs					
LLaVA v1.5 [80]	7B	82.70	0.00	0.00	0.00
Qwen2-VL-2B-Instruct [129]	2B	59.96	2.18	0.07	0.18
+SFT	2B	72.42	8.10	5.47	1.46
Qwen2-VL-7B-Instruct [129]	7B	54.46	6.07	0.69	0.80
+SFT	7B	39.54	14.18	9.62	3.30
R1-SGG-Zero	2B	0.34	12.30	11.89	5.70
R1-SGG	2B	0.10	17.87	21.09	7.48
R1-SGG-Zero	7B	0.04	15.59	18.34	8.32
R1-SGG	7B	**0.08**	**19.47**	**23.75**	**11.43**

Third, although models such as Gemini 2.0 and Qwen2-VL demonstrate a degree of spatial understanding, the task of scene graph generation is much complex than pure object detection or visual grounding. Consequently, their zero-shot performance drops significantly.

SFT vs. RL. 1) RL substantially improves performance across all metrics compared to SFT alone. Specifically, RL dramatically reduces the failure rate (*e.g.*, from 72.42% to 0.10% for 2B models) and yields significant gains in AP@50, Recall, and mRecall. This highlights the effectiveness of GRPO in enhancing the model's ability to generate accurate and complete scene graphs. **2)** SFT achieves moderate improvements in

AP@50 and Recall over the baseline but struggles with a relatively high failure rate. This suggests that SFT primarily improves relation prediction while being less effective at correcting structural errors, such as missing objects, relationships, or format inconsistencies. **3)** applying RL on top of SFT (*i.e.*, R1-SGG) further boosts performance over both SFT and R1-SGG-Zero in most cases. This indicates that combining SFT and RL benefits from better initialization, leading to stronger relation recognition and higher recall. **4)** larger models (*e.g.*, 7B) consistently outperform smaller models (*e.g.*, 2B) across AP@50, Recall, and mRecall, demonstrating the benefits of scaling model capacity for scene graph generation.

Compared to Specific Models. The gap between AP@50 and Recall highlights the advantage of dense predictions. However, our models, such as *R1-SGG*, achieve a notable mean Recall (mRecall) of 11.43%, suggesting that multimodal LLMs are more effective at generating less biased scene graphs. Moreover, specific models are typically restricted to a limited vocabulary and struggle to generalize across domains, whereas multimodal LLMs exhibit greater adaptability and broader generalization capabilities.

Overall, the results demonstrate that reinforcement learning (RL) significantly reduces the failure rate and enhances both object detection and relationship recognition. In contrast, supervised fine-tuning (SFT) alone results in a relatively high failure rate and limited improvements. As shown in Figure 5.2, the failure rate quickly drops to near-zero with RL, whereas SFT continues to suffer from frequent structural errors.

Benchmark on PSG

As shown in Table 5.4, our R1-SGG approach achieves strong performance on the PSG dataset. Compared to baselines, SFT significantly improves AP@50, Recall, and mean Recall (mRecall), while reinforcement learning further enhances relationship recognition, achieving the highest Recall (43.48% for 7B model) and mRecall

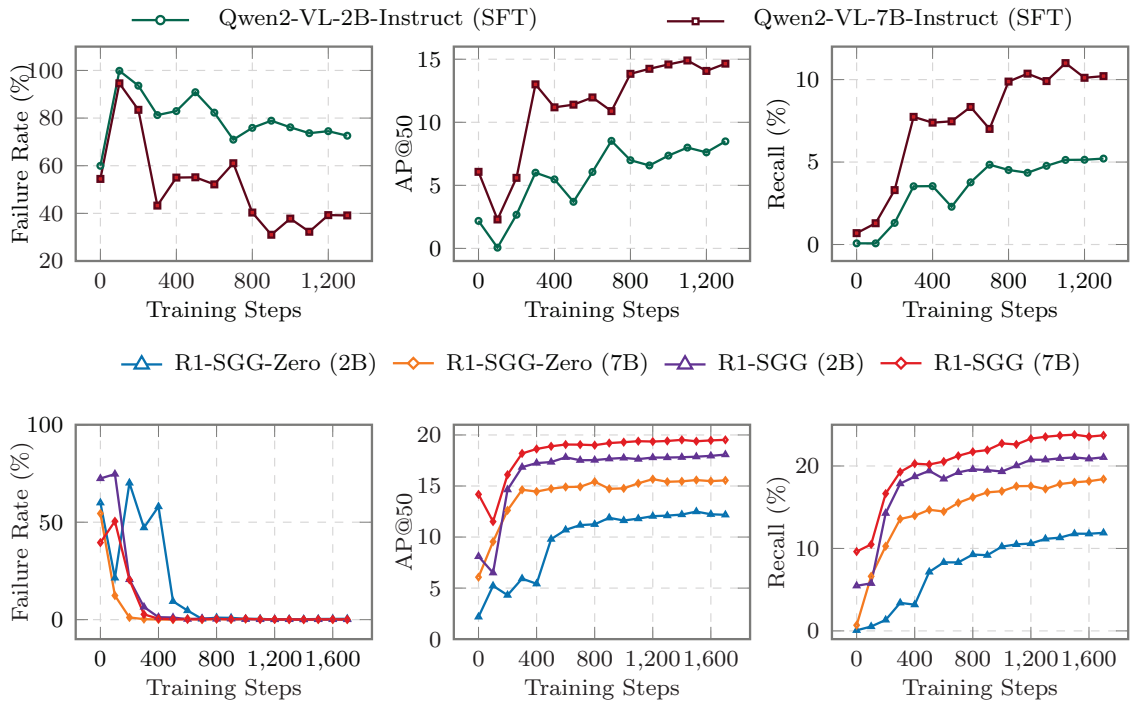
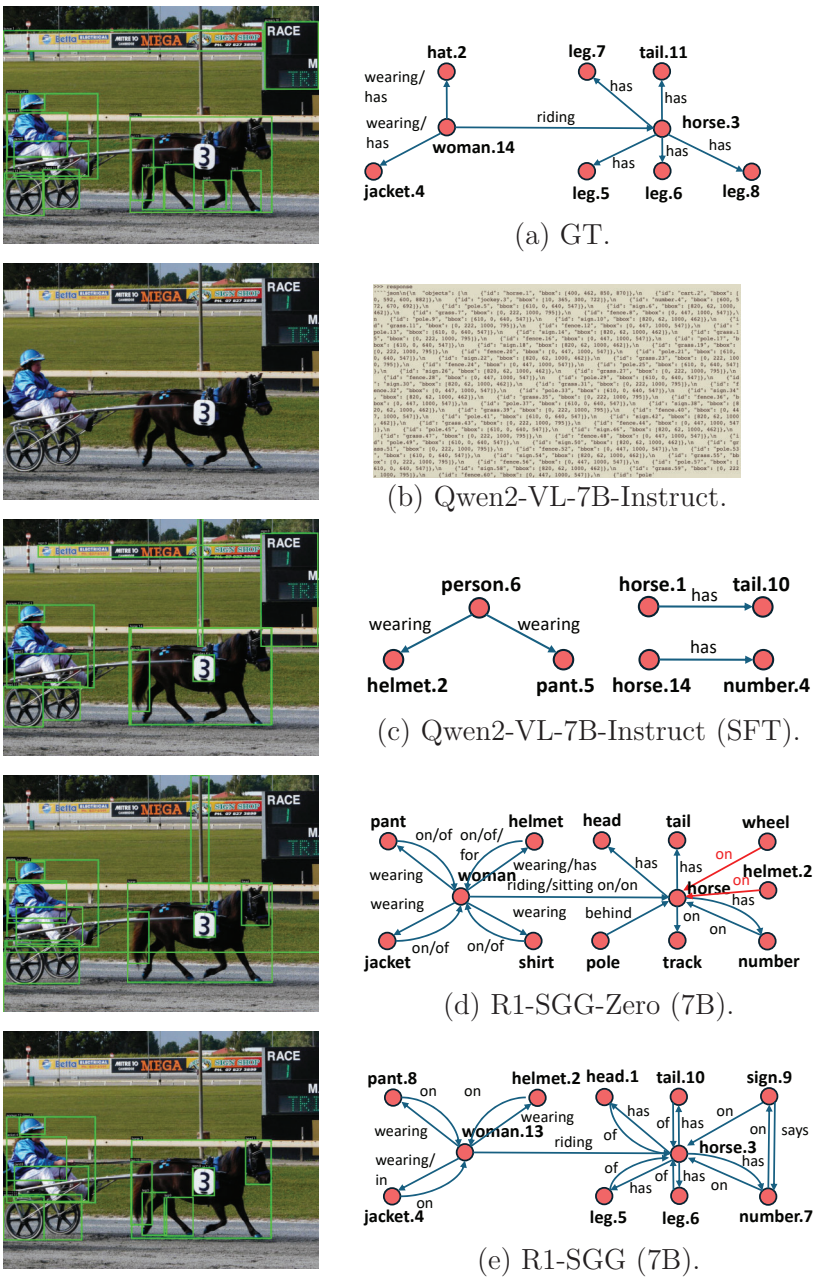


Figure 5.2: Comparison of R1-SGG-Zero and R1-SGG models against SFT baselines (Qwen2-VL-2B/7B-Instruct) across training steps on the VG150 validation set in terms of Failure Rate (%), AP@50, and Recall (%).

(33.71%). Notably, our method also drives the failure rate to zero, demonstrating the effectiveness of reinforcement learning in promoting structured, accurate scene graph generation even without predefined object categories.

5.3.4 Qualitative Results

We present qualitative results in Figure 5.3 and Figure 5.4. As shown in Figure 5.3, the ground-truth scene graph (Figure 5.3-(a)) captures key objects and their relationships but is biased toward the predicate "*has*". Conversely, the zero-shot Qwen2-VL-7B-Instruct (Figure 5.3-(b)) fails to generate a valid JSON output, indicating poor instruction-following ability. With supervised fine-tuning, the model produces



(a) GT.

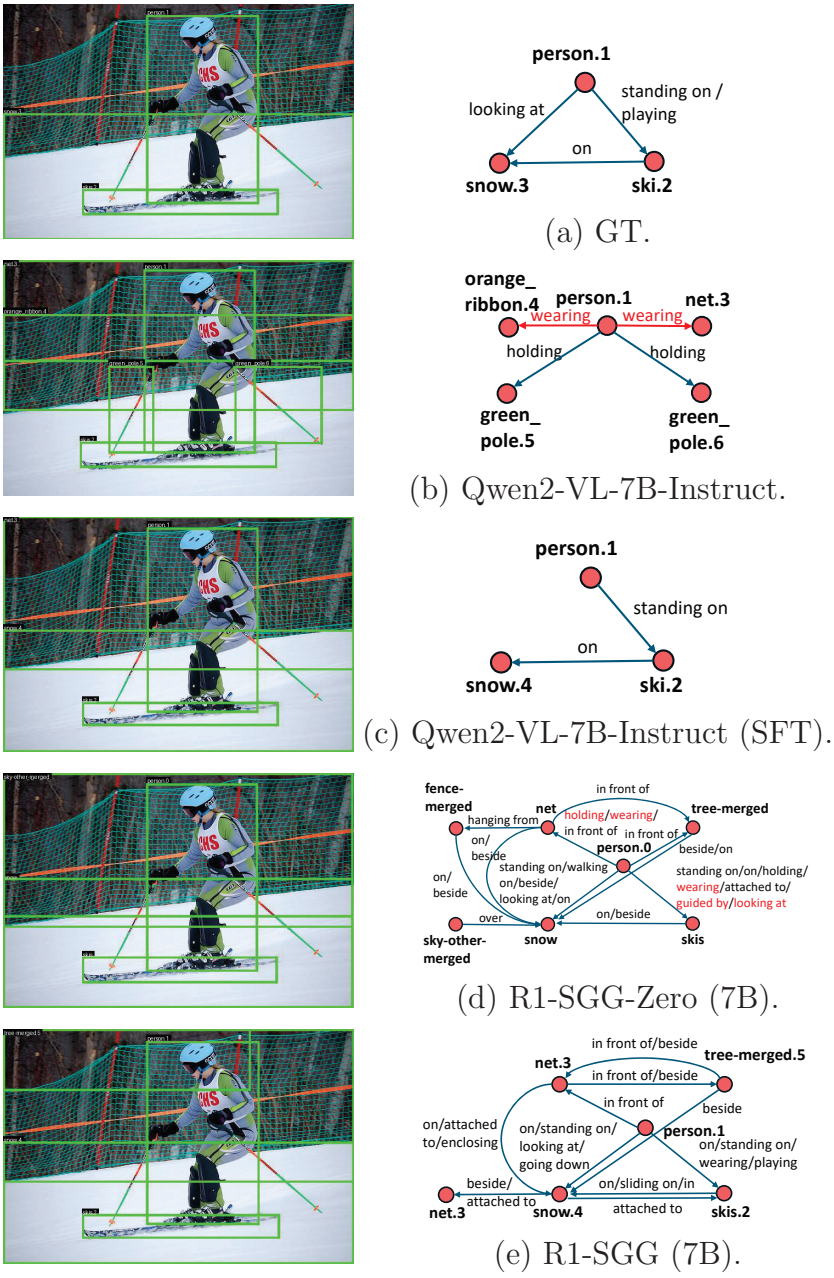
(b) Qwen2-VL-7B-Instruct.

(c) Qwen2-VL-7B-Instruct (SFT).

(d) R1-SGG-Zero (7B).

(e) R1-SGG (7B).

Figure 5.3: Qualitative comparison of generated scene graphs (from VG150). (a) Ground-truth scene graph annotated by humans. (b) Zero-shot Qwen2-VL-7B-Instruct produces an invalid JSON (failure to follow format). (c) Qwen2-VL-7B-Instruct (SFT) outputs a valid graph but omits many relations. (d) R1-SGG-Zero (7B) recovers most objects and relations but still hallucinates incorrect triplets (*e.g.*, < *wheel, on, horse*> and < *helmet.2, on, horse* >). (e) R1-SGG (7B) yields a complete, structurally correct scene graph with higher recall.



(a) GT.

(b) Qwen2-VL-7B-Instruct.

(c) Qwen2-VL-7B-Instruct (SFT).

(d) R1-SGG-Zero (7B).

(e) R1-SGG (7B).

Figure 5.4: Qualitative comparison of generated scene graphs (from PSG). (a) Ground-truth scene graph annotated by humans. (b) Zero-shot Qwen2-VL-7B-Instruct generates a valid graph but includes incorrect triplets (*e.g.*, <*person.1, wearing, net.3*>). (c) Qwen2-VL-7B-Instruct (SFT) produces a valid graph but omits some relationships. (d) R1-SGG-Zero (7B) recovers most objects and relations but still hallucinates errors (*e.g.*, <*person.0, wearing, net*>). (e) R1-SGG (7B) generates a complete and accurate scene graph with higher recall.

Table 5.4: Performance on the PSG dataset [138]. For M-LLMs, predefined object classes and relation categories are included in the input prompts.

Model	Params	Failure Rate (%)	AP@50	Recall	mRecall
Specific Models					
IMP [135]				16.50	6.50
MOTIFS [146]				20.00	9.10
VCTree [118]	-	-	-	20.60	9.70
GPSNet [78]				17.80	7.00
PSGFormer [138]				18.60	16.70
Open-sourced M-LLMs					
LLaVA v1.5 [80]	7B	81.97	0.07	0.00	0.00
TextPSG [151]	-	-	-	4.80	-
ASMv2 [130]	13B	0.87	21.45	14.77	11.82
LLaVA-SpaceSGG [136]	13B	-	-	15.43	13.23
Qwen2-VL-2B-Instruct	2B	67.20	4.89	0.39	0.26
+SFT	2B	6.54	36.05	22.06	14.92
Qwen2-VL-7B-Instruct	7B	37.97	12.75	3.18	4.33
+SFT	7B	0.96	40.79	24.73	17.11
R1-SGG-Zero	2B	0.23	25.61	25.06	18.15
R1-SGG	2B	2.70	39.28	38.49	31.21
R1-SGG-Zero	7B	0.00	32.92	37.00	32.04
R1-SGG	7B	**0.00**	**42.05**	**43.48**	**33.71**

structurally valid graphs (Figure 5.3-(c)) but frequently omits important relationships, resulting in a sparse scene graph. R1-SGG-Zero (7B), trained with RL only, improves relational recall and structure (Figure 5.3-(d)), yet still outputs inaccurate triplets such as <*wheel, on, horse*> and <*helmet.2, on, horse*>. Finally, R1-SGG (7B), trained with both SFT and RL, produces a complete and consistent scene graph (Figure 5.3-(e)), with results that even surpass the ground truth in relational richness.

5.3.5 Discussion

Through the exploration of applying GRPO to the SGG task, we make several observations.

KL Regularization. We compare models trained with and without KL divergence regularization in Figure 5.5. From the result, removing KL regularization leads to improved performance, particularly with a significant reduction in failure rate.

Sampling Length. In our experiments, the default sampling length is set to 1,024, which sufficiently covers most corrected answers. As shown in Figure 5.5, increasing the sampling length to 2,048 does not yield further performance improvements, suggesting that longer sampling might enlarge the search space and introduce additional optimization difficulties without clear benefits. This observation aligns with prior findings on test-time scaling, where increasing Chain-of-Thought (CoT) length can degrade performance [147].

Group Size. As shown in Figure 5.5, increasing the group size from 8 to 16 stabilizes training performance, consistent with the intuition that more candidates reduce variance in group statistics estimation. To balance computational cost and performance, we adopt a group size of 8 as the default in this work.

To Think or Not Think? We adopt the *<think>*$\cdots$*</think>* *<answer>*$\cdots$ *</answer>* format in the system prompt, following DeepSeek R1 [39]. However, models such as `Qwen2-VL-2B / 7B-Instruct` often fail to produce outputs with the `<think>` tag after fine-tuning, indicating difficulty in adhering to the intended structure. This suggests that rule-based rewards alone are insufficient to trigger abstract reasoning patterns like CoT, and highlights the need for additional SFT on CoT-specific datasets to incentivize coherent intermediate reasoning.

Generalization Across Datasets. We report performance comparisons across datasets in Table 5.5. The results highlight several key insights: 1) **VG150 poses a**

Table 5.5: Generalization across datasets using `Qwen2-VL-7B-Instruct` as the baseline. Columns under *Pre-training* indicate whether the weights were initialized from specific checkpoints, while the *Training* column specifies the dataset(s) used during the fine-tuning stage. "*w/o* cats." denotes prompts without predefined object classes or relation categories.

Model	Pre-Training	Training	VG150				PSG			
			Failure Rate	AP@50	Recall	mRecall	Failure Rate	AP@50	Recall	mRecall
baseline	-	-	54.46	6.07	0.69	0.80	37.97	12.75	3.18	4.33
baseline (*w/o* cats.)	-	-	44.58	6.83	0.61	0.37	30.28	13.79	1.96	2.30
SFT	-	VG150	39.54	14.18	9.62	3.30	22.10	11.05	3.03	1.36
SFT (*w/o* cats.)	-	VG150	42.98	13.03	8.94	2.47	19.81	12.15	3.87	1.81
R1-SGG-Zero	-	VG150	0.04	15.59	18.34	8.32	**0.18**	**24.92**	**13.83**	**8.90**
R1-SGG-Zero (*w/o* cats.)	-	VG150	0.06	15.30	16.33	6.94	0.18	18.10	6.16	3.38
R1-SGG	SFT	VG150	**0.08**	**19.47**	**23.75**	**11.43**	0.23	18.12	9.10	5.13
R1-SGG (*w/o* cats.)	SFT (*w/o* cats.)	VG150	0.30	18.09	22.73	9.62	0.64	14.64	7.51	3.88
SFT	-	PSG	36.98	5.79	1.42	0.77	0.91	40.58	24.75	17.31
SFT (*w/o* cats.)	-	PSG	2.54	7.94	1.77	1.25	1.01	39.02	23.70	17.17
R1-SGG-Zero	-	PSG	**0.12**	**14.22**	**8.90**	**5.34**	0.00	32.92	37.00	32.04
R1-SGG-Zero (*w/o* cats.)	-	PSG	0.02	9.08	2.80	1.78	0.05	24.26	19.94	18.04
R1-SGG	SFT	PSG	0.94	10.38	4.40	2.69	**0.00**	**42.05**	**43.48**	**33.71**
R1-SGG (*w/o* cats.)	SFT (*w/o* cats.)	PSG	0.14	9.38	2.16	1.55	0.14	41.15	41.44	31.51

significantly greater challenge than PSG. For instance, SFT trained solely on PSG achieves a high AP@50 of 40.58 and Recall of 24.75%, with a low failure rate of 0.91%. In contrast, SFT trained only on VG150 results in a much higher failure rate of 39.54%, with notably lower AP@50 (14.18) and Recall (9.62%). 2) **SFT has a strong domain-specific effect.** SFT models trained on one dataset (*e.g.*, VG150) exhibit substantial performance drops when evaluated on another (*e.g.*, PSG), reflecting limited transferability. For example, VG150-trained SFT only achieves 3.03% Recall and 1.36% mRecall on PSG. 3) **Predefined categories in the prompt.** Models trained and evaluated without categories (denoted as "*w/o* cats.") generally exhibit a slight drop in performance, while those with category information demonstrate better generalization under open-set settings. 4) **Initialization of RL matters.** R1-SGG initialized with SFT checkpoints consistently outperforms R1-SGG-Zero. On VG150, R1-SGG (7B) achieves 23.75% Recall and 11.43% mRecall versus 18.34% and 8.32%

Table 5.6: Ablation of reward formulations on VG150 validation set using R1-SGG (7B).

Setting	Sparsity	Metric Aligned	Failure Rate (%)	AP@50	Recall (%)	mRecall (%)
Hard Recall	sparse	✓	0.08	19.47	23.75	11.43
Hard Recall + Relax	medium	✗	0.02	19.93	24.05	9.61
Soft Recall	dense	✗	0.06	18.73	21.92	5.61

for R1-SGG-Zero. A similar trend is observed on PSG. This highlights the importance of using SFT as a warm-start for reinforcement learning, which leads to improved sample efficiency and stronger downstream performance. 5) **R1-SGG-Zero exhibits stronger cross-dataset generalization.** This aligns with the domain-specific nature of SFT—models trained via SFT tend to overfit to the source domain, resulting in degraded performance on unseen datasets. In contrast, R1-SGG-Zero, trained without SFT, generalizes more robustly across domains.

Hard Recall vs. Soft Recall. As shown in Table 5.6, *Hard Recall* outperforms other variants despite providing sparser reward signals. This highlights the importance of aligning reward functions with evaluation metrics, rather than prioritizing reward smoothness alone.

5.4 Summary

In this chapter, we present a reinforcement learning framework for enhancing end-to-end Scene Graph Generation (SGG) using multimodal large language models (MLLMs). To align training with the structured nature of scene graphs, we design a set of rule-based reward functions, including three variants—*Hard Recall*, *Hard Recall+Relax*, and *Soft Recall*—along with a format consistency reward. These components enable fine-grained and stable policy optimization through Graph-based Reinforced Preference Optimization (GRPO).

Our approach significantly improves the structural correctness, relational diversity, and semantic accuracy of generated scene graphs. By directly optimizing graph-level outputs via reinforcement learning, rather than relying solely on supervised learning, our method demonstrates the effectiveness of preference-based training for structured vision-language tasks. We release our code, models, and benchmarks to support future research on multimodal structured understanding with large language models.

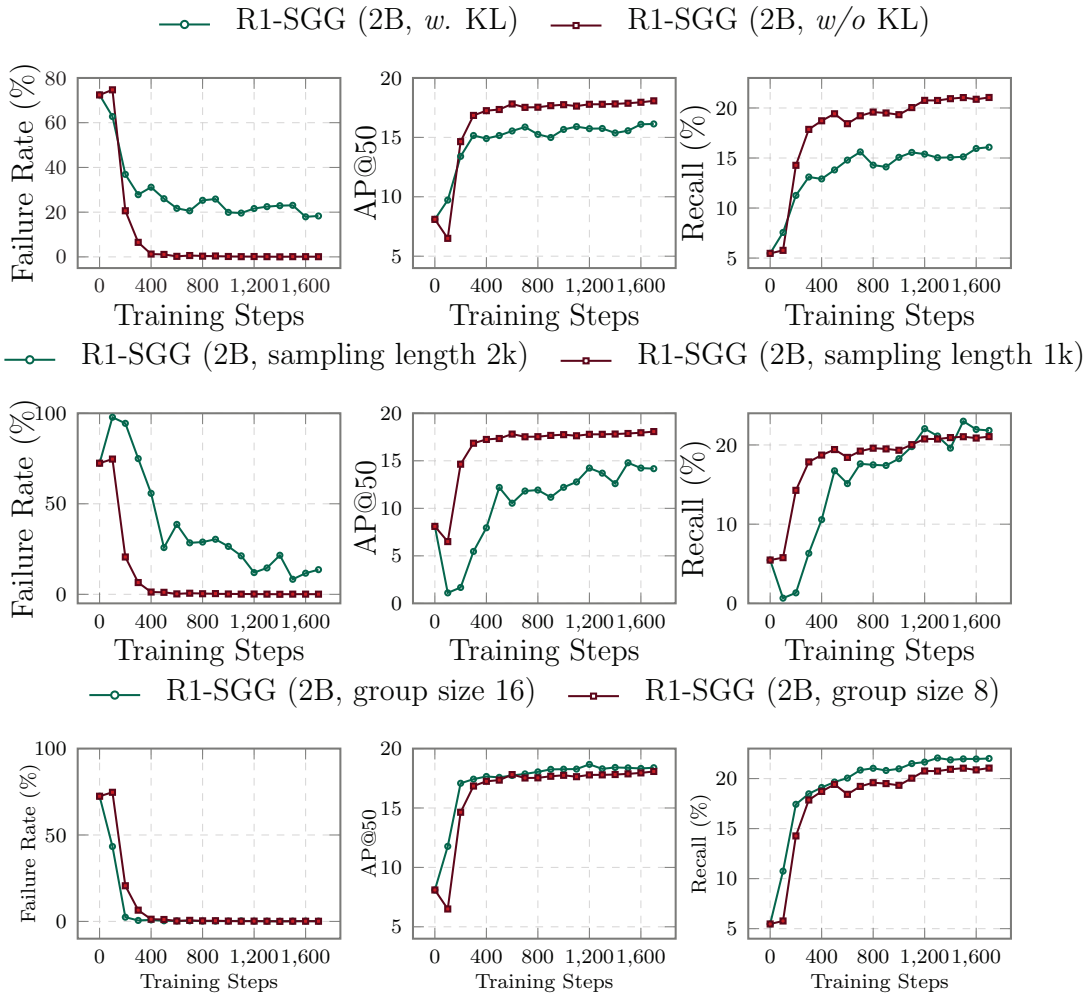


Figure 5.5: Performance comparison of R1-SGG (2B) across training steps on the VG150 validation set. Each row evaluates a different setting: (Top) KL divergence regularization (β=0.04 vs. β=0), (Middle) sampling length, and (Bottom) group size. Metrics reported include Failure Rate (%), AP@50, and Recall (%).

Chapter 6

Image Generation with Scene Graphs

"If you can't measure it, you can't improve it."

– Peter Drucker

While previous chapters focus on Scene Graph Generation—deriving structured representations from visual inputs—this chapter explores the inverse problem: generating realistic images conditioned on scene graphs. By leveraging the rich semantic and structural information encoded in scene graphs, we aim to guide image synthesis in a controllable and interpretable manner.

6.1 Introduction

Generating realistic images coherent with natural scenes is important in numerous applications such as photo editing [48, 155, 54], content creation [53, 4, 107], *etc.*. Early generative models like Variational Autoencoders (VAE) [58] produced blurry

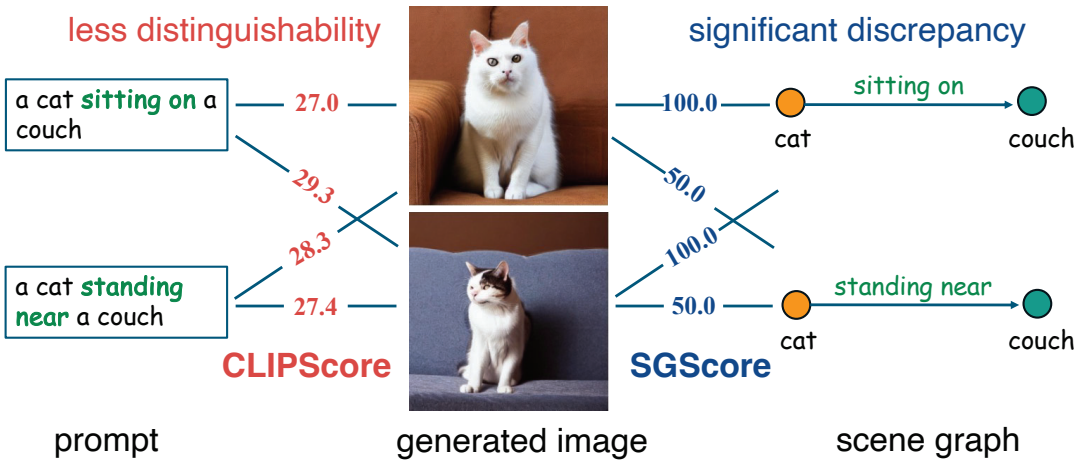


Figure 6.1: A comparison of CLIPScore [41] and the proposed SGScore for evaluating factual consistency. SGScore can distinguish such relationship discrepancies, while CLIPScore often overlooks them.

images due to limitations in modeling complex data distributions. Generative Adversarial Networks (GAN) [34] improved image quality but faced issues like training instability and mode collapse. Recently, diffusion models like Stable Diffusion [107] have proven effective for generating visually appealing images with realistic objects and high-resolution details [43, 94, 100, 27]. Although diffusion models have achieved significant success, they still face challenges in generating complex scenes involving multiple objects [140, 84], particularly in ensuring factual consistency, such as the accurate presence of multiple objects and the correct relationship between objects within a natural image.

Recent efforts have aimed to address these limitations, focusing on compositional objects [85, 30, 89], improving text-image alignment [62, 28], and enhancing spatial consistency [9]. For instance, methods like Composable Diffusion [85] compose multiple concepts by explicitly optimizing the defined energy functions, and Structured Diffusion [30] combines multiple objects by manipulating cross-attention layers. These methods improve the accuracy of multiple objects occurring in a single scene. Addi-

tionally, Chatterjee [9] proposed a benchmark to evaluate and enhance the capability of modeling spatial relationships.

Despite this progress, *how to evaluate the factual consistency between the condition (e.g., text, image, etc.) and the generated image* remains challenging. The difficulty lies in standard metrics like Fréchet Inception Distance (FID) [42] and CLIPScore [41] primarily evaluate image quality but fall short in capturing factual consistency in complex scenes. FID, widely used to assess the visual fidelity of generated images, focuses on feature distribution matching between real and generated datasets. However, it overlooks spatial relationships and object interactions. For instance, images depicting a dog "*under a table*" and "*on a table*" may receive similar FID scores, despite their vastly different relationships. Similarly, CLIPScore measures semantic alignment between images and text by emphasizing global themes but cannot assess specific object relationships. CLIPScore may assign high scores to images that include all relevant objects but incorrectly depict their relationships, such as confusing "*a cat sitting on a couch*" with "*a cat standing near a couch*" (see Figure 6.1). The drawback of FID and CLIPScore highlights the need for more specialized evaluation metrics to assess objects' presence and precise interactions between them.

To evaluate the factual consistency, we employ a structured representation known as a *Scene Graph*, which has been demonstrated to outperform pure text in image retrieval [52, 59]. A scene graph encodes objects as nodes and their relationships as edges. For textual conditions, scene graphs can be parsed using natural language processing tools such as Scene Parser [90] or large language models (LLMs). For image conditions, scene graphs are generated via Scene Graph Generation (SGG) models [135, 146, 118, 17] or multimodal LLMs. Leveraging this structured representation, we introduce a novel evaluation metric, *SGScore*, which quantifies the factual consistency between generated images and their corresponding scene graphs. SGScore evaluates *Object Recall* by verifying the presence of nodes and *Relation Recall* by assessing the accuracy of edges within the scene graph. To adapt different domains and

handle the extensive vocabulary inherent in generated images, we utilize a multimodal LLM to perform these evaluations instead of relying on a pre-trained SGG model to convert images into scene graphs. Thanks to the reasoning and zero-shot capabilities of the multimodal LLM, SGScore can effectively distinguish between images depicting subtle differences, as shown in Figure 6.1.

When applying the new metric to benchmark different generative models, a critical bottleneck is the lack of large-scale datasets annotated with scene graphs, which are essential for fair and comprehensive comparisons. Existing datasets, such as Visual Genome (VG) [59] and COCO-Stuff [6], are relatively small (*e.g.*, only 5k and 2k images for testing, respectively), and their inherent long-tail distributions lead to biased evaluations. As a result, we develop *MegaSG* (see Section 4.4 of Chapter 4), a large-scale dataset comprising one million images richly annotated with scene graphs that capture a wide range of objects and their complex relationships. MegaSG enables models to be trained and evaluated on diverse scenarios, from simple to highly intricate scenes, thus overcoming the limitations of previous datasets that were constrained by small-scale and biased distributions.

By combining the proposed *SGScore* and *MegaSG*, we introduce a novel benchmark, *Scene-Bench*. To provide a comprehensive and fair benchmark, we sample images from MegaSG based on Scene Diversity and Scene Complexity. Scene Diversity sampling aims to evaluate model performance across diverse scene scenarios, while Scene Complexity sampling aims to evaluate model performance at different complexity levels. To our knowledge, *Scene-Bench* is the first benchmark to evaluate generative models on a large-scale natural scene dataset using scene graphs.

Building upon this scene graph-based evaluation, we design a scene graph feedback pipeline that leverages multimodal LLMs for iterative refinement. The process begins with generating an initial image from a scene graph, followed by assessing factual consistency using the Scene-Bench metrics. When discrepancies are detected, such as missing objects or incorrect relationships, a missing graph is created to highlight

these errors. A reference image is generated based on this missing graph to address the identified issues. By integrating this new image with the initial one, we refine the output, resulting in a final image that more accurately matches the intended scene described by the original scene graph.

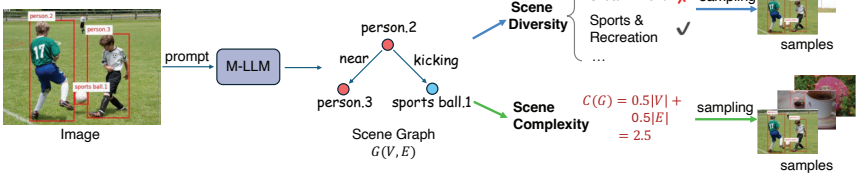
In short, our contribution can be summarized as

- We introduce *Scene-Bench*, a comprehensive and large-scale benchmark for evaluating factual consistency in scene graph-to-image generation. *Scene-Bench* includes *MegaSG*, a dataset with one million images annotated with scene graphs, and a novel evaluation metric, *SGScore*, which explicitly measures factual consistency by assessing the accuracy of object presence and relationships in generated images.

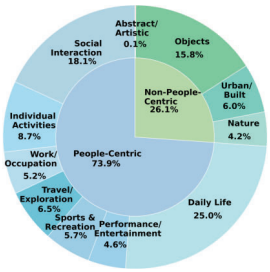
- We propose a scene graph feedback strategy that iteratively refines generated images by detecting and correcting discrepancies in object presence and relationship accuracy, thereby enhancing the factual consistency between the generated image and the intended scene.

- Extensive experiments demonstrate that *Scene-Bench* provides a more comprehensive and effective evaluation benchmark for factual consistency in natural scenes. Furthermore, our proposed feedback pipeline significantly improves the factual consistency of generated images, particularly in complex scene scenarios.

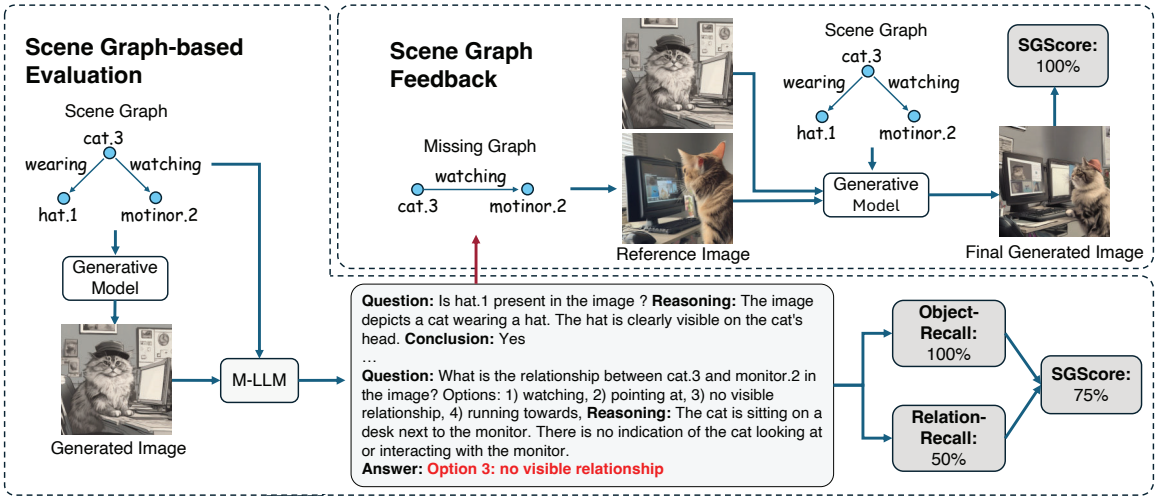
Scene-Bench is a comprehensive benchmark that evaluates and enhances the factual consistency of natural scene generation from scene graphs by rigorously verifying both object presence and inter-object relationships. Specifically, Scene-Bench consists of a large-scale dataset of scene graphs, and an autonomous evaluation pipeline. The overview of Scene-Bench is shown in Figure 6.2.



(a) Creation of the MegaSG.



(b) Scene distribution.



(c) Scene Graph-based Evaluation and Scene Graph Feedback.

Figure 6.2: Overview of the Scene-Bench. (a) Scene graphs are generated from images using a multimodal LLM (M-LLM), capturing object relationships and interactions. Scene Diversity and Scene Complexity guide sampling to ensure dataset balance. (b) Scene distribution across categories highlights the diversity in People-Centric and Non-People-Centric themes. (c) Scene graph-based evaluation and feedback leverages the M-LLM to calculate object and relationship recall, generating an SGScore metric that quantifies factual consistency between the generated image and the intended scene. The feedback identifies and corrects discrepancies, iteratively refining the generated image.

6.1.1 MegaSG: a large-scale dataset of scene graphs

Creation of the Dataset. Due to the complexity and high cost of manual annotation, existing scene graph datasets, such as Visual Genome [59], are relatively small in scale (*e.g.*, only 5k images are prepared for the test set [51]). The limited size and inherent long-tail distribution make these datasets inadequate for studying diffusion models across diverse scene scenarios. To address this limitation and build a large-scale scene graph dataset, we leverage the reasoning capabilities of multimodal large language models (LLMs) in combination with pre-existing object detection datasets. Specifically, we collect 1 million images from COCO [14], Object365 [111], and Open Images v6 [60], which offer rich object categories and bounding boxes. These datasets are ideal for generating large-scale scene graphs efficiently. For additional details, please see Section 4.4 of Chapter 4.

Scene Diversity. To better understand the behavior of diffusion models in diverse scene scenarios, we utilized an LLM (*e.g.*, Gemini 1.5 Flash [104]) to classify the MegaSG dataset into two main themes: *People-Centric* and *Non-People-Centric*. The *People-Centric* theme includes fine-grained categories such as *Social Interaction*, *Individual Activities*, *Work / Occupation*, *Travel / Exploration*, *Sports & Recreation*, *Performance / Entertainment*, *Daily Life*. For the *Non-people-Centric* theme, we identified categories like *Nature*, *Urban / Built*, *Objects*, and *Abstract / Artistic*. The hierarchical distribution of these categories is illustrated in Figure 6.2 (b). And samples are shown in the Figure 6.3, showcasing the diversity and range of scenarios covered in the dataset.

Human Evaluation In addition to categorizing natural scenes, measuring scene complexity is crucial for evaluating the performance of diffusion models. While simple scenes are generally easier for these models to handle, complex scenes pose greater challenges. This raises an important question: *How can we quantitatively define the complexity of a natural scene?*

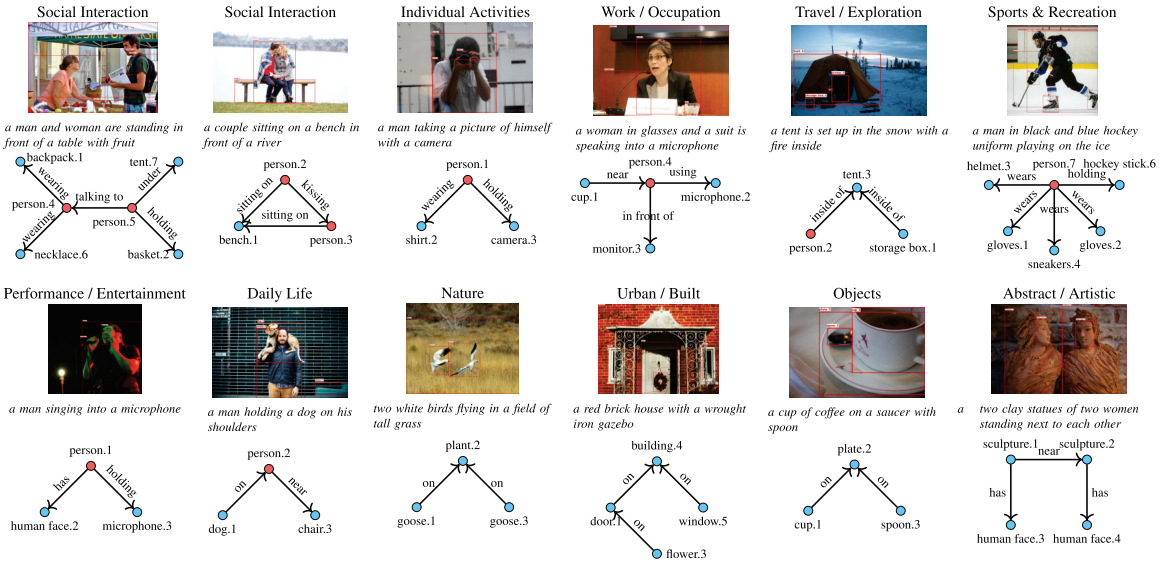


Figure 6.3: Illustration of scene categories in the MegaSG dataset. The image shows various themes, such as People-Centric (*e.g.*, social interaction, individual activities) and Non-People-Centric (*e.g.*, nature, urban environments). The caption is provided for illustrative purposes and generated using BLIP-2 [66], and the scene graph is constructed as described in Section 4.4 of Chapter 4.

In this work, we define the complexity of a scene based on its scene graph $G = (V, E)$, where V represents the set of nodes (objects) and E represents the set of edges (relationships). The complexity is calculated as

$$C(G) = \gamma \cdot |V| + (1.0 - \gamma) \cdot |E|, \tag{6.1}$$

where $\gamma \in [0, 1]$ is a weighting factor that balances the influence of the number of nodes and edges.

The scene graph representation provides a straightforward way to quantify complexity, making it possible to analyze the performance of diffusion models across a range of difficulty levels, from simple to highly complex scenes. In contrast, defining the complexity of a text prompt is inherently more challenging due to the lack of explicit structural information. By leveraging this graph-based approach, we can better understand how diffusion models respond to varying levels of scene complexity, offering

Table 6.1: Comparison of MegaSG with widely-used scene graph datasets. "Obj./Rel." denotes the number of object and relationship categories, while "Balanced" indicates whether the dataset is balanced with respect to scene complexity. The VG statistics follow the pioneering work [51] in SG2IM, and almost all subsequent works adopt this setting.

Dataset	Images	Obj./Rel.	Test		
			Samples	Triplets	Balanced
VG [59]	108k	179 / 49	5,096	12.8k	✗
COCO-Stuff [6]	4.5k	171 / 6	2,048	22.7k	✗
MegaSG	1M	775 / 122	50,000	275k	✓

insights into their strengths and limitations across different scenarios.

Dataset Comparison. We compare the MegaSG with existing scene graph datasets, specifically Visual Genome (VG) [59] and COCO-Stuff [6], as summarized in Table 6.1. MegaSG significantly outperforms VG and COCO-Stuff in terms of scale, encompassing 1 million images compared to VG's 108k and COCO-Stuff's 4.5k. Additionally, MegaSG offers a substantially richer vocabulary with 775 object categories and 122 relations, enhancing the diversity and complexity of scene annotations. Importantly, MegaSG's test set is *complexity balanced*, ensuring an even distribution of simple, medium, and hard scenes, whereas VG and COCO-Stuff lack this balanced composition.

6.1.2 Evaluation Strategy

To quantify the factual consistency, we utilize a multimodal LLM (M-LLM) to assess the recall of objects and relationships, as shown in Figure 6.2 (c).

Recall of Objects. Given a generated image I and its intended scene graph

$G = (V, E)$, where V represents the set of objects (nodes) and E represents the relationships (edges), we prompt the M-LLM with specific queries about the existence of each object. For example, for a scene graph containing the relationships: {*"source": "person.2", "target": "sports ball.1", "relation": "kicking"*}, {*"source": "person.2", "target": "person.3", "relation": "near"*}, we would prompt the M-LLM with questions such as "*Is there a sports ball in the image?*". The M-LLM, based on its multimodal capabilities, examines the generated image and responds with a binary answer (Yes / No) to indicate whether the specified object is present. We define the object recall as the fraction of correctly identified objects in the generated image:

$$\text{ObjectRecall}(G, I) = \frac{|V\text{pred} \cap V_{\text{gt}}|}{|V_{\text{gt}}|}, \tag{6.2}$$

where V_{pred} is the set of objects the LLM identifies as present in the image, and V_{gt} is the set of ground-truth objects from the original scene graph.

Recall of Relationships. To further assess the quality of the generated scene, we evaluate the recall of relationships between objects in the image. For each relationship $r \in E$, we check whether the predicted relationship r_{pred} exists between the corresponding objects in the generated image. The relationship recall is defined as:

$$\text{RelationRecall}(G, I) = \frac{|E_{\text{pred}} \cap E_{\text{gt}}|}{|E_{\text{gt}}|}, \tag{6.3}$$

where E_{pred} represents the predicted relationships between objects in the generated scene, and E_{gt} represents the ground-truth relationships from the original scene graph. To obtain E_{pred}, we prompt the LLM with multiple-choice questions such as: *"What is the relationship between the person and the sports ball in the image? A) kicking; B) throwing; C) holding; D) no visible relationship."*

SGScore. In addition to individual recalls of objects and relationships, we introduce a comprehensive metric, *SGScore*, which evaluates the overall quality of the scene graph in terms of both objects and relationships. SGScore is computed as a weighted

combination of object recall and relationship recall:

$$\begin{aligned} \text{SGScore}(G, I) = \alpha \cdot \text{ObjectRecall}(G, I) + \\ (1.0 - \alpha) \cdot \text{RelationRecall}(G, I), \end{aligned} \tag{6.4}$$

where $\alpha \in [0, 1]$ is a hyperparameter that controls the relative importance of object recall versus relationship recall. By adjusting this weight, we can tune the evaluation to place more emphasis on either the objects or the relationships, depending on the task requirements. SGScore provides a holistic evaluation of how well the generated scene aligns with the scene graph, offering a balanced measure that reflects both object accuracy and relationship consistency.

6.2 Scene Graph Feedback

Building on the scene graph-based evaluation, we propose a scene graph feedback to iteratively refine the generated image based on identified discrepancies between the image and the input scene graph. This process leverages multimodal LLMs to analyze the generated scene and provide targeted feedback for refinement.

Specifically, given a scene graph $G = (V, E)$, we first perform scene composition using an LLM (see Table C.3 in the Appendix), in which nodes and edges are seamlessly integrated into a *prompt* for an exact scene. This *prompt* results in an initial image I_0 through the diffusion model f_D. With the input scene graph G and the generated image I_0, a multimodal LLM f_M has been applied to evaluate the presence of objects and relationships. The missing objects and relationships are constructed as a missing graph G_{miss}. If discrepancies exists, *e.g.*, $G_{\text{miss}} \neq (\emptyset, \emptyset)$, we will generate a reference image I_1 conditioned on the G_{miss} as does in generating I_0 conditioned on G. To generate the final output image, we use IP-Adapter [142] to integrate *prompt*, I_0, and

I_1, in which the cross attention process can be formulated as

$$\begin{aligned} Z =& \text{Attention}(Q, K_{prompt}, V_{prompt}) + \\ & \lambda_0 \cdot \text{Attention}(Q, K_{I_0}, V_{I_0}) + \\ & \lambda_1 \cdot \text{Attention}(Q, K_{I_1}, V_{I_1}), \end{aligned} \tag{6.5}$$

where Q is the query features of the latent variable, K_{prompt} / V_{prompt}, K_{I_0} / V_{I_0}, K_{I_1} / V_{I_1}, are the projected features of *prompt*, I_0, I_1, respectively. The Attention is defined as $\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d}})V$ as does in [122]. λ_0, λ_1 are weight factors.

6.3 Experimental Results

6.3.1 Experimental Setup

Models. We evaluate several popular open-source diffusion models, including Composable [85], Structured [30], SD v1.5 [107] (checkpoint: *runwayml/stable-diffusion-v1-5*), SD v2.1 [107] (checkpoint: *stabilityai/stable-diffusion-2-1*), PixArt-α [10] (checkpoint: *PixArt-alpha/PixArt-XL-2-1024-MS*), SD3 [27] (checkpoint: *stabilityai/stable-diffusion-3-medium-diffusers*), SD3.5 [27] (checkpoint: *stabilityai/stable-diffusion-3.5-large*), SDXL [100] (checkpoint: *stabilityai/stable-diffusion-xl-base-1.0*), and LLM-based methods such as RPG [140]. We use `diffusers` [123] or official code to benchmark these models.

Datasets. We benchmark models on the widely used Visual Genome (VG) and the proposed MegaSG dataset.

- VG consists of 108k images annotated by human. Following SG2Im [51], it has been split into training set (62,565), validation set (5,506), and test set (5,088[1]) images for scene graph-based image generation.

[1]we use official code to obtain 5,096 images for test.

- MegaSG comprises 1 million images annotated using Gemini 1.5 Flash. Relationships with a frequency below 100 are filtered out, and synonyms are merged by a large language model (LLM).

Metrics. We employ common metrics and the proposed SGScore.

- Inception Score (IS) [108]: Measures the realism of generated images using a pre-trained Inception-V3 [116] network.
- Fréchet Inception Distance (FID) [42]: Assesses the similarity between generated and real images by measuring the distance between the distributions of their feature representations.
- CLIPScore [41]: Evaluates the semantic alignment between generated images and corresponding text using the CLIP model [101].
- SGScore measure the factual consistency in terms of object recall and relation recall. We use $\alpha = 0.5$ in Eq. 6.4 to give a balanced measurement.

Scene Graph Representation. For text-to-image (T2I) models that condition on a sentence, we encode scene graphs in the format "`{subject} {predicate} {object}`" (*e.g.*, `cat sitting on desk, dog near chair`). For the prompt that converts a scene graph into a consistent description (*i.e.*, scene composition in Section 6.2), please refer to Table C.3 of the Appendix.

Scene Complexity. Based on Scene complexity defined as Eq. 6.1, we categorize the scene complexity into three levels: *simple*, *medium*, and *hard*. The three levels of complexity are defined as follows:

- **Simple**: $1 \leq C(G) \leq 3$, typically involving 2–3 objects and no more than 3 relationships in the scene.

Table 6.2: Model Comparison on the VG test set. Models including SGDiff and SceneGenie are trained on VG train set. Since SceneGenie [29] does not release the code, we only present the reported IS and FID.

Method	Resolution	IS ↑	FID ↓	CLIPScore ↑	SGScore ↑			
					Overall	Simple (# 3993)	Medium (# 930)	Hard (# 173)
SGDiff [139]	256x256	16.0	29.6	-	64.5	64.2	66.3	61.5
SceneGenie [29]	256x256	20.2	42.2	-	-	-	-	-
Composable [85]	512x512	20.5	47.5	22.0	48.0	48.9	45.0	44.5
Structured [30]	512x512	23.0	42.2	22.0	52.5	51.8	54.6	56.1
SD v1.5 [107]	512x512	23.1	42.8	22.0	52.5	51.9	54.7	53.9
SD v2.1 [107]	768x768	20.8	46.6	22.1	54.4	53.5	57.8	57.9
PixArt-α [10]	1024x1024	20.8	52.9	22.1	59.8	58.5	64.1	67.0
SD3.5 [27]	1024x1024	21.5	45.6	22.1	60.5	59.4	64.1	65.9
SD3 [27]	1024x1024	23.4	44.5	22.1	62.1	60.9	66.3	66.7
SDXL [100]	1024x1024	23.1	43.4	22.1	60.7	59.6	64.0	69.6
RPG [140] (SDXL)	1024x1024	22.9	44.2	19.3	69.3 (+14.2%)	69.4 (+16.4%)	68.7 (+7.3%)	70.5 (+1.3%)
Ours (SD v1.5)	512x512	20.7	41.6	19.1	65.1 (+24.0%)	65.1 (+25.4%)	65.1 (+19.0%)	66.8 (+23.9%)
Ours (SDXL)	1024x1024	21.0	42.7	19.3	74.1 (+22.1%)	74.2 (+24.5%)	73.3 (+14.5%)	75.3 (+8.2%)

- **Medium**: $4 \leq C(G) \leq 7$, characterized by a denser arrangement of objects and relationships.
- **Hard**: $C(G) \geq 8$, representing the most challenging cases with highly dense objects and intricate relationships.

LLM. In addition to utilizing Gemini 1.5 Flash [104] (cutoff November 2024), we also present results using GPT-4o [97], Qwen-VL-Max [128], and LLaVA 1.5 [81] to evaluate the robustness of the proposed *SGScore*.

IP-Adapter. We use the official implementation in `diffusers` [123], with λ_0 and λ_1 (in Eq. 6.5) empirically set to 0.5.

6.3.2 Evaluation of Scene-Bench

Performance on Visual Genome. Table 6.2 presents the results on the VG test set. The first finding is that *SGScore* provides much more distinguishability than other

Table 6.3: Model comparison on a 50,000-image subset of the MegaSG dataset, sampled with a Scene Complexity of $\gamma = 0$. Scene graph-based methods such as SGDiff [139], which are limited by the vocabulary of the VG dataset, were excluded from testing.

Method	**Resolution**	**IS** ↑	**FID** ↓	**CLIPScore** ↑	**SGScore** ↑			
					Overall	Simple (# 15k)	Medium (# 20k)	Hard (# 15k)
Composable [85]	512x512	20.3	41.0	22.9	42.0	61.0	39.0	28.3
Structured [30]	512x512	28.6	26.2	23.0	53.9	65.1	53.9	46.0
SD v1.5 [107]	512x512	27.0	29.1	22.8	54.2	64.9	53.4	44.7
SD v2.1 [107]	768x768	24.9	34.0	22.9	57.8	68.2	56.4	49.2
PixArt-α [10]	1024x1024	24.3	43.9	23.0	59.5	68.4	58.2	52.7
SD3.5 [27]	1024x1024	25.9	34.5	23.0	63.4	73.1	61.9	55.7
SD3 [27]	1024x1024	27.2	35.5	23.0	65.2	74.2	63.8	58.0
SDXL [100]	1024x1024	25.3	31.6	23.0	65.6	72.9	64.6	59.6
RPG [140] (SDXL)	1024x1024	23.4	37.5	20.0	71.0 (+8.2%)	76.5 (+4.9%)	70.5 (+9.1%)	66.1 (+10.9%)
Ours (SD v1.5)	512x512	23.1	28.9	19.9	62.0 (+14.4%)	71.0 (+9.4%)	61.3 (+14.8%)	53.9 (+20.6%)
Ours (SDXL)	1024x1024	21.6	34.1	20.0	77.1 (+17.5%)	81.8 (+12.2%)	76.6 (+18.6%)	73.1 (+22.7%)

metrics like FID and CLIPScore. For instance, SD v1.5 has a better FID score than SD v2.1 (42.8 vs. 46.6), yet its SGScore is lower than that of SD v2.1 (52.5 vs. 54.4), indicating there are more missed objects and relationships in the images generated by SD v1.5. Another finding is that due to the VG test set being biased towards simple scenes, the performance on medium and hard scenes is counterintuitive: the performance should decrease as scene complexity increases, but this trend is not consistently observed, largely because of the limited number of complex scenes in the test set. This counterexample justifies why we need a new large-scale benchmark to evaluate models comprehensively.

Performance on MegaSG. To fairly assess models' abilities to handle more complex scenes, we evaluated them on a large-scale subset of the MegaSG dataset, sampled by Scene Complexity ($\gamma = 0$). As shown in Table 6.3, we observe a general decline in performance across all models. For example, SD v1.5's SGScore drops from 64.9 (simple scenes) to 44.7 (hard scenes), indicating they struggle more with accurately modeling the relationships in complex scenes.

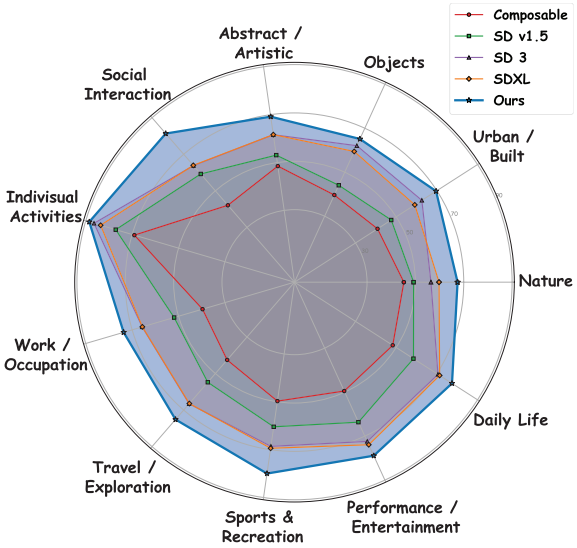
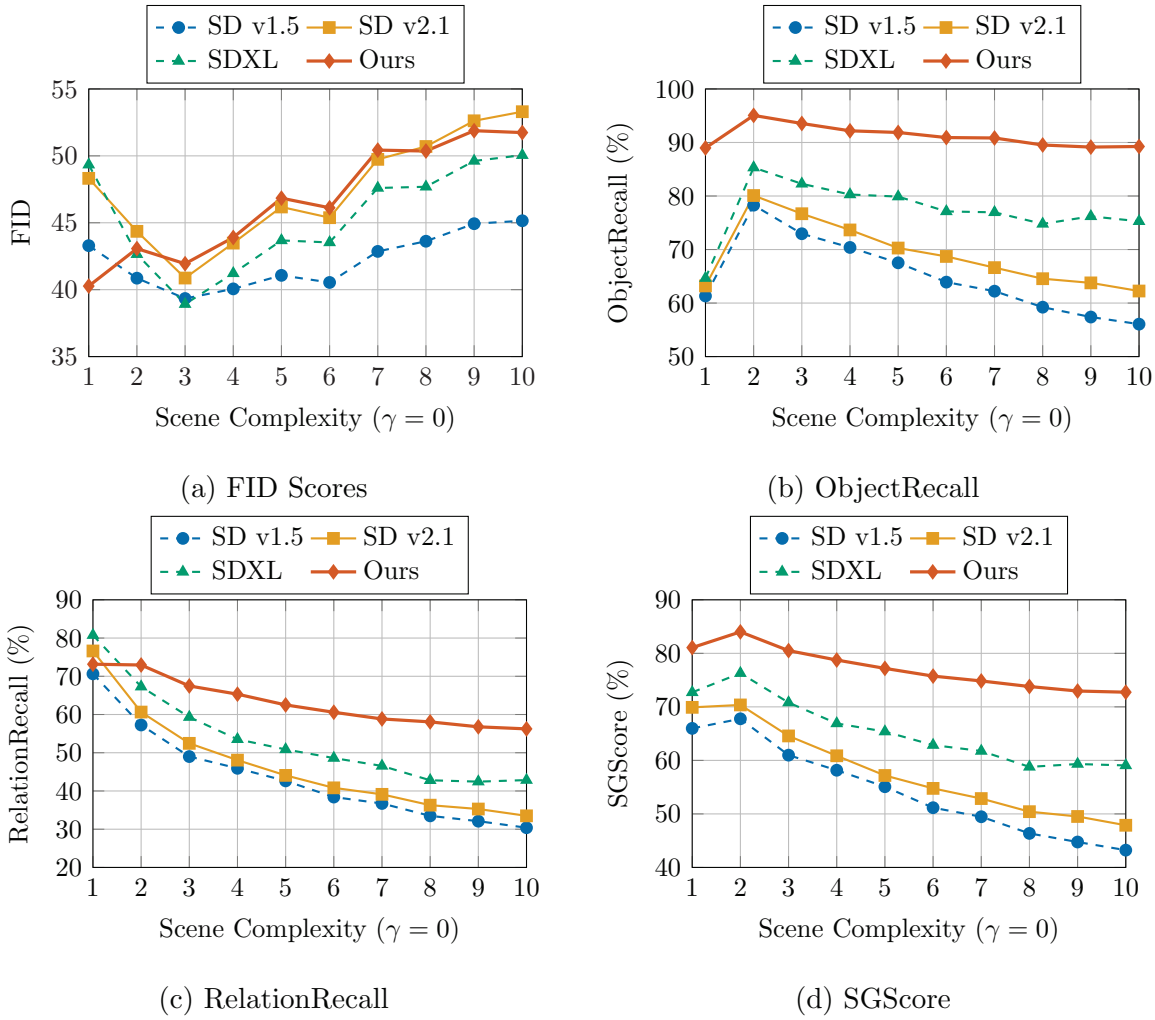


Figure 6.4: Comparison of model performances using SGScore across various scene categories.

Performance Across Scene Diversity. We evaluated model performance across diverse scene categories identified in our Scene Diversity analysis (see Section 6.1.1). 6.4 presents results for categories like *Social Interaction*, *Nature*, and *Urban Environments*. Models generally perform better in categories like *Individual Activities*, *Performance / Entertainment*, and *Daily Life*, but face challenges in *Social Interaction* (where scenes often include multiple people) and *Abstract / Artistic* (due to style discrepancies), *etc.*. This variation underscores the importance of evaluating models across a broad range of scenarios to comprehensively assess their strengths and limitations.

Impact of Scene Complexity. To examine how scene complexity affects model performance, we analyzed FID and SGScore for SD v1.5, SD v2.1, SDXL, and our model across various complexity levels (see Figure 6.5).

As scene complexity increases (*i.e.*, with more objects and relationships), we observe a consistent decline in SGScore across all models. This suggests that, with greater complexity, the models struggle to accurately represent the expected scene graphs.



(a) FID Scores

(b) ObjectRecall

(c) RelationRecall

(d) SGScore

Figure 6.5: Comparison of FID, ObjectRecall, RelationRecall, and SGScore for models SD v1.5, SD v2.1, SDXL, and Ours across different scene complexity levels. (a) FID scores show relatively stable image quality, while (b) ObjectRecall and (c) RelationRecall indicate a consistent decline in factual consistency with increasing scene complexity. (d) SGScore demonstrates the overall advantage of our approach in maintaining higher factual consistency, particularly in complex scenes.

The decreasing SGScore highlights the challenge of maintaining factual consistency in complex scenes. However, our model demonstrates a notable improvement over the other models by consistently achieving a higher SGScore across all complexity levels, particularly through maintaining stable and high object recall. This suggests that our model is more effective at preserving factual consistency even in complex scenes.

Interestingly, FID scores remain stable across complexity levels, indicating that image quality does not degrade significantly with complexity. This stability implies that while models retain visual fidelity, they encounter difficulties modeling intricate object relationships and interactions in complex scenes. Therefore, even as images appear visually coherent, the factual accuracy, as measured by SGScore, declines with increased scene complexity.

Our findings reveal that, although image quality (measured by FID) remains stable, increasing scene complexity significantly degrades the factual consistency of scene representations (measured by SGScore). Notably, our model consistently outperforms competitors by maintaining higher object and relationship recall across all complexity levels.

6.3.3 Evaluation of Scene Graph Feedback

We evaluated our scene graph feedback pipeline using two diffusion models: SD v1.5 [107] and SDXL [100]. For each model, we compared three settings: baseline (without scene composition or feedback), with scene composition only, and with both scene composition and feedback.

Results and Analysis. Table 6.4 summarizes the results. For SD v1.5, the baseline achieved an ObjectRecall of 64.93% and a RelationRecall of 44.19% (SGScore 54.56%). Incorporating scene composition improved these metrics to 75.45% and 48.84% (SGScore 62.14%), demonstrating that detailed prompts help the model better capture specified objects and relationships. Applying our feedback strategy further

Table 6.4: Effectiveness of the scene graph feedback on 5,000 images sampled from MegaSG.

Model	ObjectRecall ↑	RelationRecall ↑	SGScore ↑
SD v1.5 [107]			
Baseline	64.93 ± 0.31	44.19 ± 0.09	54.56 ± 0.12
+ Scene Composition	75.45 ± 0.19	48.84 ± 0.39	62.14 ± 0.25
+ Feedback	$\mathbf{79.93} \pm 0.34$	$\mathbf{53.97} \pm 0.20$	$\mathbf{66.95} \pm 0.23$
SDXL [100]			
Baseline	77.22 ± 0.22	53.78 ± 0.36	65.50 ± 0.19
+ Scene Composition	88.07 ± 0.17	60.37 ± 0.14	74.22 ± 0.14
+ Feedback	$\mathbf{91.30} \pm 0.24$	$\mathbf{63.20} \pm 0.10$	$\mathbf{77.25} \pm 0.21$

increased ObjectRecall to 79.93% and RelationRecall to 53.97% (SGScore 66.95%), indicating effective correction of discrepancies. A similar trend was observed in SDXL, where improvements after applying scene composition and feedback increased the SGScore from 65.50% to 77.25%.

Compared with LLM-based methods like RPG [140], the performance gain on VG or MegaSG is significant. RPG [140] utilizes an LLM as an agent to perform re-captioning, region planning and merging, while it lacks a feedback for ensuring factual consistency.

These results demonstrate that our scene graph feedback effectively enhances factual consistency by identifying and correcting discrepancies between generated images and the intended scene graphs.

Ablation Study of IP-Adapter. To evaluate the effectiveness of the scene graph feedback, particularly considering the additional parameters introduced by the IP-Adapter, we conducted another ablation study. In this experiment, we set $\lambda_1 = 0$ in Eq. 6.5, meaning the IP-Adapter processes only the initial generated image, without

Table 6.5: Comparison of ObjectRecall, RelationRecall, and SGScore with and without the reference image in the IP-Adapter setup. "Ref. Img." denotes the reference image.

IP-Adapter	Ref. Img.	ObjectRecall	RelationRecall	SGScore
✗	✗	75.45	48.84	62.14
✓	✗	70.91	49.83	60.37
✓	✓	**79.93**	**53.97**	**66.95**

incorporating the reference image derived from the missing graph.

As shown in Table 6.5, introducing the IP-Adapter alone (row 2 vs. row 1) does not improve the factual consistency of generated images. However, incorporating the reference image (row 3) significantly enhances ObjectRecall, RelationRecall, and SGScore, demonstrating the importance of scene graph feedback.

6.3.4 Qualitative Evaluation

We present qualitative results to demonstrate the effectiveness of Scene-Bench and the proposed scene graph feedback. Fig. 6.6 compares images generated by various models using the same scene graphs. Most of models often struggle with complex scenes, leading to images with missing objects or incorrectly depicted relationships. For example, when generating a scene from the scene graph *<person.2, holding, baseball glove.1>, <person.3, wearing, helmet.4>*, previous models may omit the *helmet* or fail to represent *person wearing helmet*.

With the proposed scene graph feedback, the generated image more faithfully represents the intended scene graph. The feedback process identifies missing elements and corrects relational inaccuracies, resulting in the image where one person is wearing a helmet and another is holding a baseball glove. This demonstrates the model's

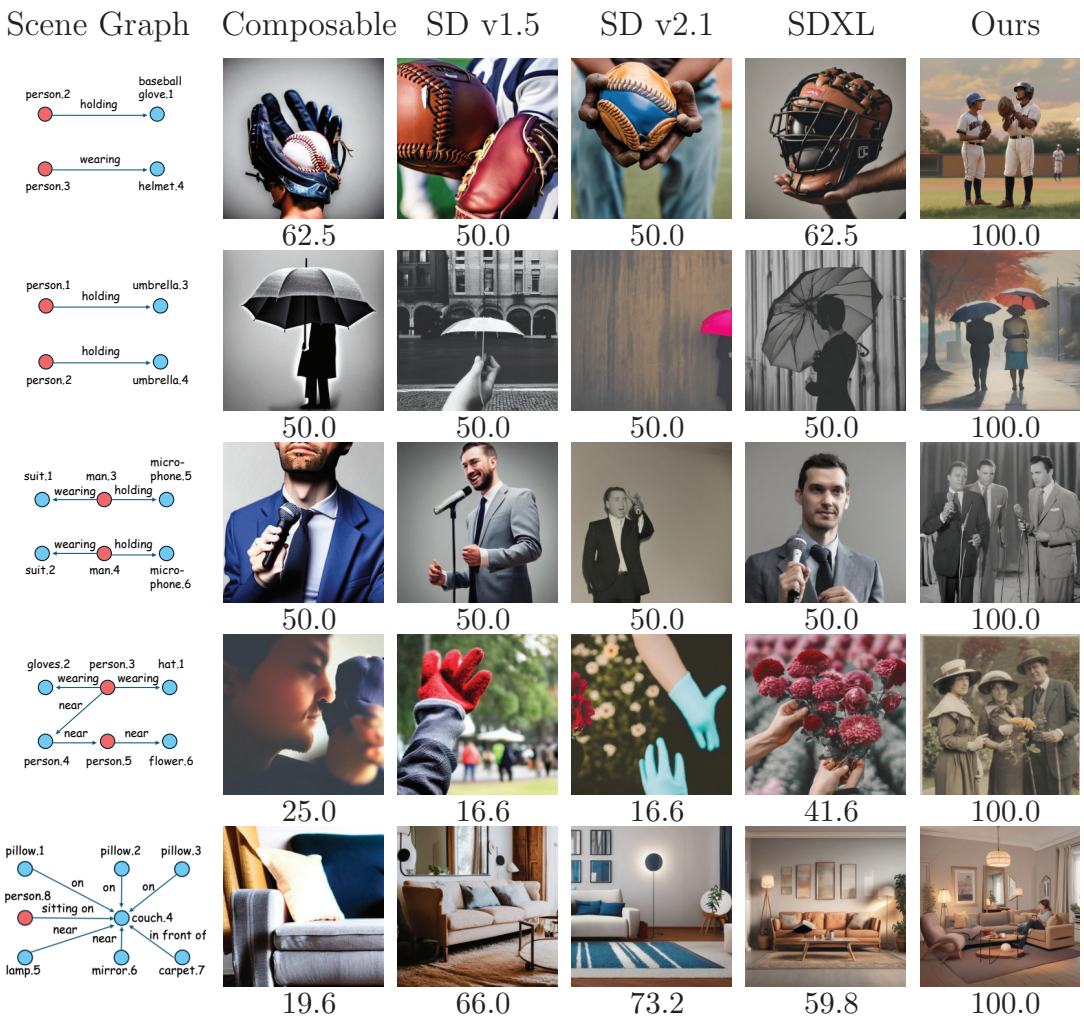


Figure 6.6: Comparison of Scene Graph-based Image Generation across Different Models. Each row displays a unique scene graph used as input for image generation. We present the **SGScore** below each generated image to quantify the consistency between the scene graph and the generated output.

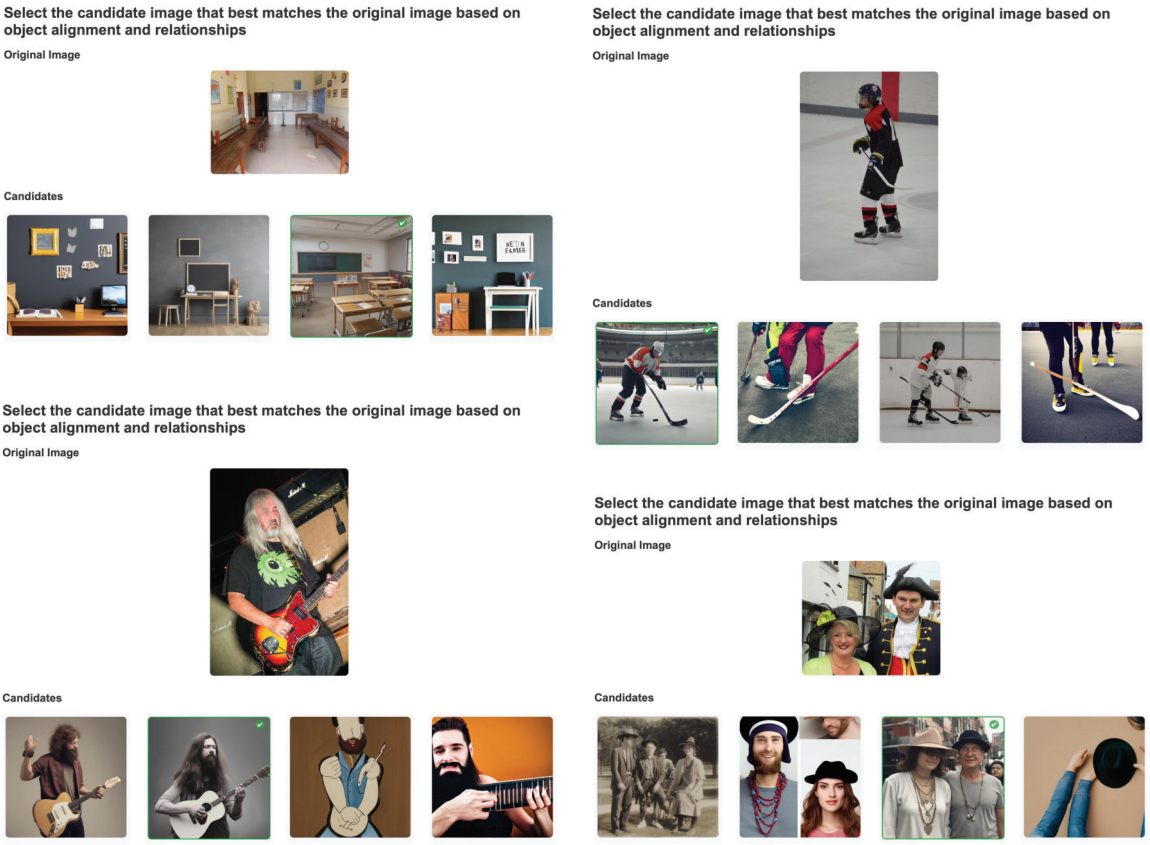


Figure 6.7: Example questions presented to human annotators.

improved ability to handle complex object interactions and spatial arrangements, highlighting the benefits of our approach.

6.3.5 Human Evaluation

To validate the efficacy of *SGScore* in improving the verification of factual consistency and assess the impact of the proposed feedback pipeline, we conducted a human evaluation. Specifically, three human annotators were instructed to select the candidate image that best aligns with the original image regarding object presence and relationship accuracy. We randomly sampled 1,000 images and selected corresponding generated images from four models: SD v1.5, SD v2.1, SDXL, and Ours (SDXL). Model identities were hidden from the annotators to avoid bias. Figure 6.7 illustrates

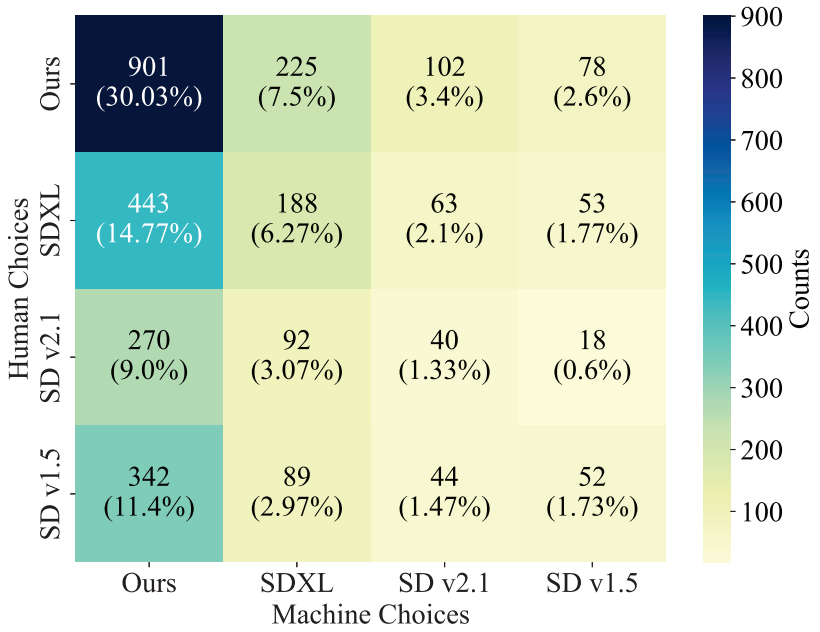


Figure 6.8: Confusion matrix showing the comparison of human choices against machine choices based on SGScore.

the annotation interface.

Each query displayed an original image alongside four generated images produced by different models. Annotators were instructed to select the image that most accurately preserved the object presence and relationships of the original. As illustrated in Figure 6.8, both human judgments and machine-based selections consistently favored our model, thereby confirming its superior factual consistency as measured by *SGScore*.

6.4 Discussion

In the field of SG2IM, most existing approaches [51, 2, 139, 29, 113, 83, 131] focus primarily on network architecture design, train on widely used datasets (*e.g.*, Visual Genome and COCO-stuff), and report conventional metrics such as IS, FID, and CLIPScore. However, *assessing factual consistency and leveraging identified inconsistency as feedback remain underexplored*. To address this gap, we introduce a

large-scale scene graph dataset, a novel metric *SGScore*, and an automatic evaluation pipeline to systematically measure the factual consistency in SG2IM. Based on the evaluation pipeline, we incorporate a training-free feedback pipeline to enhance factual consistency.

6.4.1 Sensitivity Analysis of SGScore

To test the scalability and sensitivity of SGScore, we randomly sample subsets (*e.g.*, with size 500, 1k, 2k, 4k, $\cdots$, 32k, *etc.*) from the MegaSG to compute the SGScore on images generated by SD v1.5. As shown in Figure 6.9, the mean SGScore remains relatively stable across sample sizes, with only slight variations observed. Additionally, the standard deviation decreases as the sample size increases, demonstrating that the metric becomes more reliable and less sensitive to random fluctuations with larger subsets. This indicates that SGScore is both scalable and robust, providing consistent evaluations of factual consistency regardless of the dataset size.

6.4.2 Binding Attributes or Not?

One potential concern for *MegaSG* is its lack of attribute annotations, as users might expect attributes to play a role in scene generation—for example, distinguishing between `a red apple is above the black car` and `a black apple is above the red car`. Does omitting attributes in *MegaSG* adversely affect the modeling of such distinctions?

Recent work [22] demonstrates that diffusion models can effectively bind attributes (*e.g.*, color, texture, shape) under certain conditions. However, most existing SG2IM approaches [51, 139, 29, 113, 83] do not explicitly model attributes, despite their availability in the Visual Genome dataset. To evaluate the impact of attribute modeling, we augment 775 object categories in *MegaSG* with attribute labels (*e.g.*, $\{apple \rightarrow$

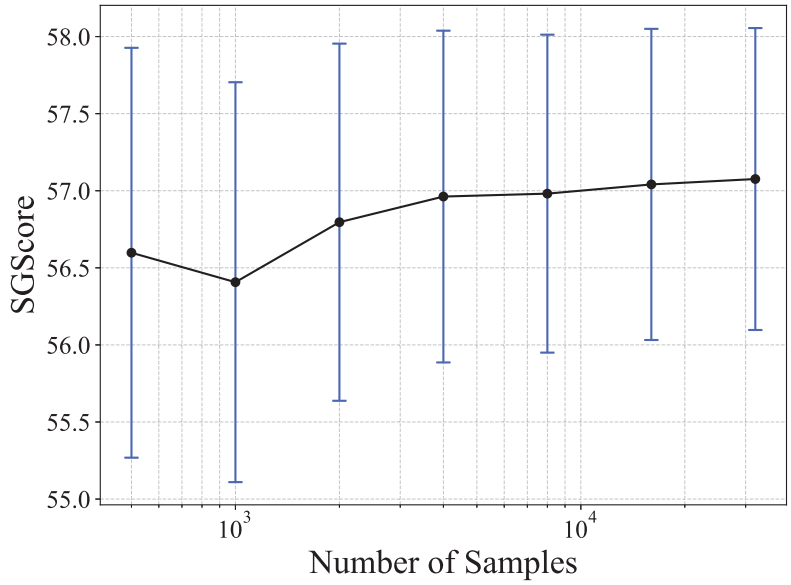


Figure 6.9: Mean and standard deviation (std.) of SGScore across varying numbers of samples. For each sample size, the image generation process is repeated with different random seeds using SD v1.5 to compute the mean and std. of SGScore.

red apple, *green apple*, *ripe apple*}, {*door* → *wooden door*, *red door*, *glass door*}, *etc.*). Each object category is augmented three times by a multimodal LLM (the prompt can be found in Table C.1 in the Appendix), yielding a total of 2,325 text prompts for diffusion models. As shown in Table 6.6, diffusion models consistently handle both single-object and attribute-bound object generation with minimal performance differences, suggesting that explicit attribute modeling offers only marginal benefits in this setting.

More importantly, our experiments (*e.g.*, Table 6.4 and Figure 6.5) reveal that current diffusion models struggle to capture spatial and interactive relationships, rather than merely modeling individual objects. This shortcoming can be attributed to two factors. First, T2I models typically employ a CLIP text encoder to process the textual input; however, due to dataset biases (*e.g.*, LAION 5b [109]) during training, the encoder learns a visual–text alignment that favors object presence over the relationships

Table 6.6: Recall comparison of single-object vs. attribute-bound generation on 775 object categories. The recall evaluation is performed by Gemini 1.5 Flash.

	SD v1.5	SDXL
w.o. attributes	93.90 ± 0.29	92.40 ± 0.36
w. attributes	90.12 ± 0.26	93.34 ± 0.15

between objects. Similarly, the training data for Stable Diffusion models predominantly consists of single-object images, providing limited exposure to the alignment of multiple objects and their interactions.

Considering the trade-off between annotation cost and the minimal gains from introducing attributes, we opted to omit attributes in the construction of *MegaSG*. Our focus is on effectively evaluating scene graph-to-image generation (SG2IM), thus providing a promising direction for optimizing diffusion models.

6.4.3 Compared to Existing Benchmarks

Although our work focuses on Scene Graph-to-Image (SG2IM) generation, textual inputs can be directly parsed into structured scene graphs. To validate the effectiveness of our proposed *SGScore* in enhancing factual consistency, we compare *SGScore* with two text-to-image (T2I) benchmarks, namely TIFA [45] and DSG [21].

TIFA [45] examines faithfulness in Visual Question Answering (VQA) by leveraging GPT-3 [5] for question generation and using question answering modules such as mPLUG [64] and BLIP-2 [68]. With these text prompts, we synthesize images using different diffusion models to compare *SGScore* with the TIFA score, evaluated using the `mPLUG-large` model. To compute *SGScore*, we utilize an LLM (*e.g.*, Gemini 1.5 Flash) with the prompt detailed in Table C.2 to extract a textual scene graph from a caption. From Figure 6.10, we observe a consistent trend between the TIFA

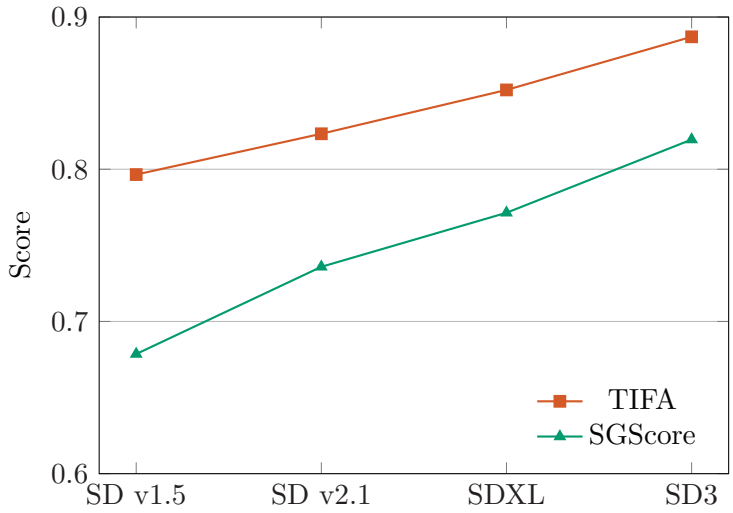


Figure 6.10: Comparison of TIFA and SGScore across different Stable Diffusion models on TIFA v1.0 benchmark.

score and *SGScore* across different diffusion models on the TIFA v1.0 benchmark, indicating that both metrics reliably reflect factual consistency. However, while TIFA is designed for text-to-image (T2I) generation, our *SGScore* focuses on scene graph-to-image (SG2IM) generation.

Similary, DSG [21] decomposes the text prompt into a series of tuples (*e.g.*, entities, attributes, relations) and generate corresponding questions for verifying the faithfulness of text-to-image generation. However, due to the high computational cost (see Section 6.4.5), we opted not to report DSG scores against TIFA and *SGScore*.

6.4.4 Multimodal LLMs for SGScore

We evaluate the performance of different multimodal LLMs on SGScore, including Gemini 1.5 Flash [104], GPT-4o [97], Qwen-VL-Max [128], and LLaVA 1.5 [81], as shown in Figure 6.11. The results show minimal discrepancies across models, indicating that SGScore is insensitive to the specific choice of state-of-the-art multimodal LLMs for the same evaluation. However, the visual reasoning capability of these

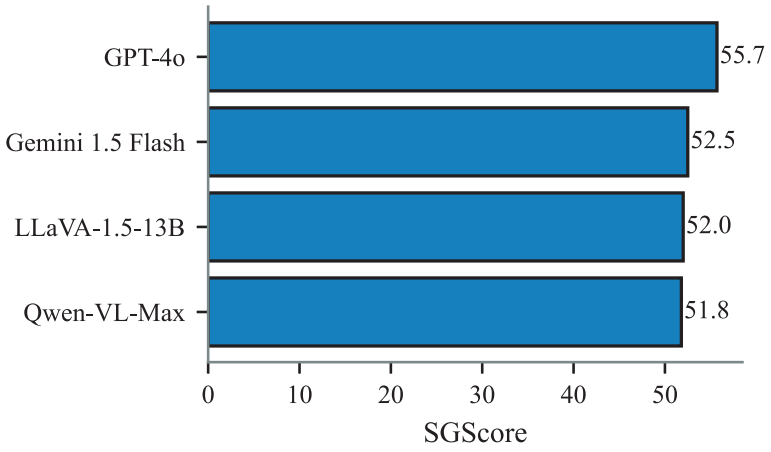


Figure 6.11: Performance comparison of M-LLMs on VG test set (images are generated by SD v1.5).

models remains important. Considering cost-effectiveness, we recommend Gemini 1.5 Flash, which offers excellent multimodal reasoning performance at a significantly lower price [1], as indicated in Section 6.4.5.

6.4.5 Computation Cost

Computation cost for evaluation pipeline. We benchmark 5,000 samples using Gemini 1.5 Flash to assess computation cost. On average, input and output token consumption are ∼4.7M and ∼3.9M, respectively, with a cost of ∼1.5 USD for 5k samples (as of March 2025)[2]. In contrast, DSG [21] requires ∼45M input tokens and ∼0.7M output tokens, resulting in a cost of ∼7 USD for benchmarking 5k samples—even when using the considerably cheaper and more powerful LLM model GPT-4o mini [98] compared to the originally employed `gpt-3.5-turbo -16k-0613` in the paper.

Computation cost for scene graph feedback. While scene graph feedback can

[2]Gemini API pricing [1]

be iteratively applied to refine generated results, we perform a single refinement step once discrepancies are detected by our evaluation pipeline. As shown in Table 6.3 and Table 6.2, this approach yields a significant performance gain despite the additional computational overhead. A related work in T2I generation [134] also employs an iterative correction pipeline, integrating an object detector with an LLM, which requires extra computational cost.

6.5 Summary

In this chapter, we present *Scene-Bench*, a comprehensive benchmark for evaluating the factual consistency of image generation conditioned on scene graphs. Central to this benchmark is the large-scale dataset *MegaSG* and our proposed metric *SGScore*, which measures the presence and relational correctness of objects through the reasoning capabilities of multimodal large language models (M-LLMs).

To further enhance image fidelity, we introduced a scene graph feedback mechanism that detects and corrects inconsistencies between the input scene graph and the generated image. This iterative refinement process leads to substantial improvements in factual consistency, especially in complex scenes where traditional metrics fail to provide reliable guidance.

Extensive experiments across various models and datasets demonstrate that *Scene-Bench* enables rigorous and interpretable evaluation, facilitating progress in controllable image generation. We believe that *Scene-Bench* will establish a new standard and inspire future research in high-fidelity, controllable generation.

Chapter 7

Conclusion and Future Work

7.1 Conclusion

This thesis presents a comprehensive study on advancing Scene Graph Generation (SGG) towards more generalized, scalable, and practical applications. We began by identifying key limitations in traditional SGG models, including reliance on closed vocabularies, high annotation costs, and a lack of alignment between training objectives and the structured nature of outputs. To address these limitations, we explored four major research directions:

First, we proposed a unified transformer-based architecture, *OvSGTR*, for fully open-vocabulary SGG, capable of recognizing unseen objects and relations by leveraging visual-language alignment and retention mechanisms. This approach enables a flexible and extensible vocabulary, improving generalization to real-world scenarios.

Second, we introduced a weakly-supervised paradigm, *GPT4SGG*, which reverses the conventional SGG pipeline by generating scene graphs from holistic and region-specific image narratives. This reduces the dependence on expensive manual annotations and demonstrates the feasibility of language-supervised SGG.

Third, we developed a reinforcement learning framework with rule-based reward functions to optimize structured outputs of LLMs directly. By designing recall-based and format-consistency rewards, we align model training more closely with scene graph evaluation metrics, leading to improved structural accuracy and relational completeness.

Finally, we extended our work to the inverse task of image generation from scene graphs. We proposed a new benchmark, *Scene-Bench*, and a metric, *SGScore*, to quantitatively evaluate object presence and relationship fidelity in generated images using multimodal large language models (M-LLMs). This structured feedback enables iterative refinement of generative outputs and highlights the potential of scene graphs as controllable and interpretable interfaces for image synthesis.

Collectively, the methods and contributions of this thesis offer a new perspective on scalable scene understanding. Through innovations in modeling, supervision, training objectives, and evaluation, this work advances the field toward robust, open-world visual reasoning.

7.2 Future Work

While this thesis provides a significant step forward in generalized SGG, several promising directions remain open for future research.

- **Scaling to 3D and temporal data.** Most current work focuses on 2D static images. Extending scene graph generation to 3D point clouds, video sequences, or embodied environments would enable richer and more dynamic scene representations, with potential applications in robotics and AR/VR.

- **Interactive and multi-agent SGG.** Future systems may involve collaborative agents that communicate via structured representations. Investigating how

scene graphs can serve as a representation for agent interaction and reasoning is a fruitful direction for embodied AI.

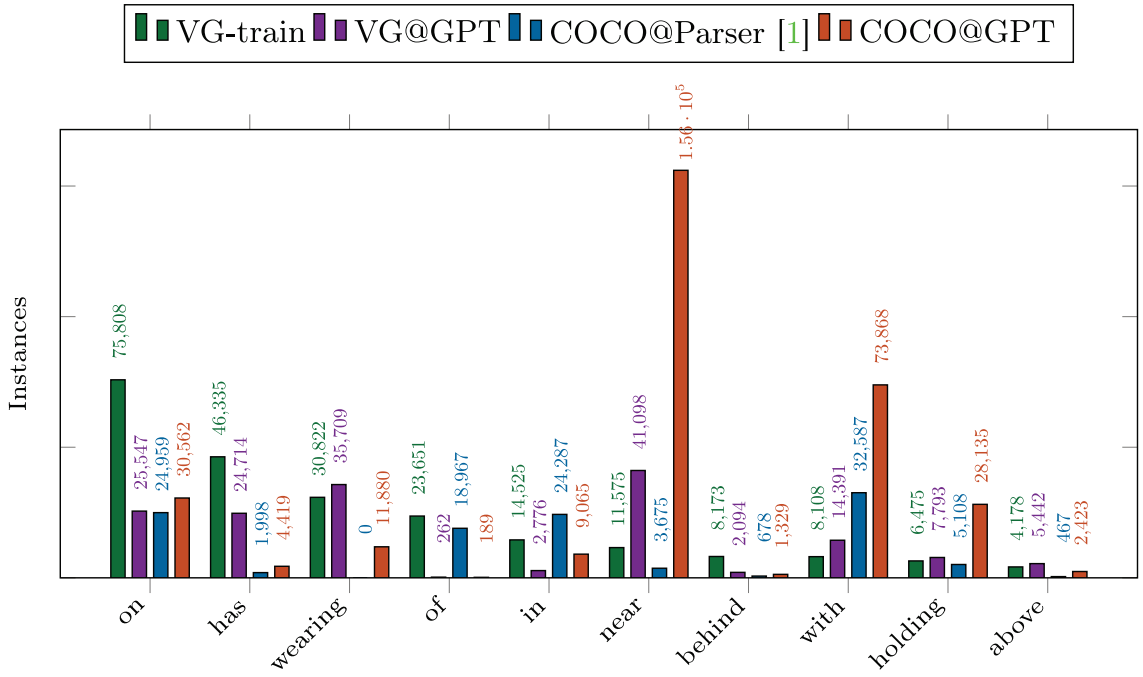
- **Joint learning with other modalities.** Integrating audio, text, and tactile information into the scene graph generation pipeline could lead to more holistic scene understanding, particularly in real-world multimodal contexts.
- **Unifying SGG with large vision-language models.** As M-LLMs become increasingly powerful, future research can explore tighter integration between explicit scene graphs and implicit knowledge representations, enabling joint reasoning and generation with structured and unstructured data.

In conclusion, this thesis lays a solid foundation for open-world, language-guided, and feedback-driven scene graph generation, paving the way for its continued evolution as a key enabler of interpretable and controllable visual intelligence.

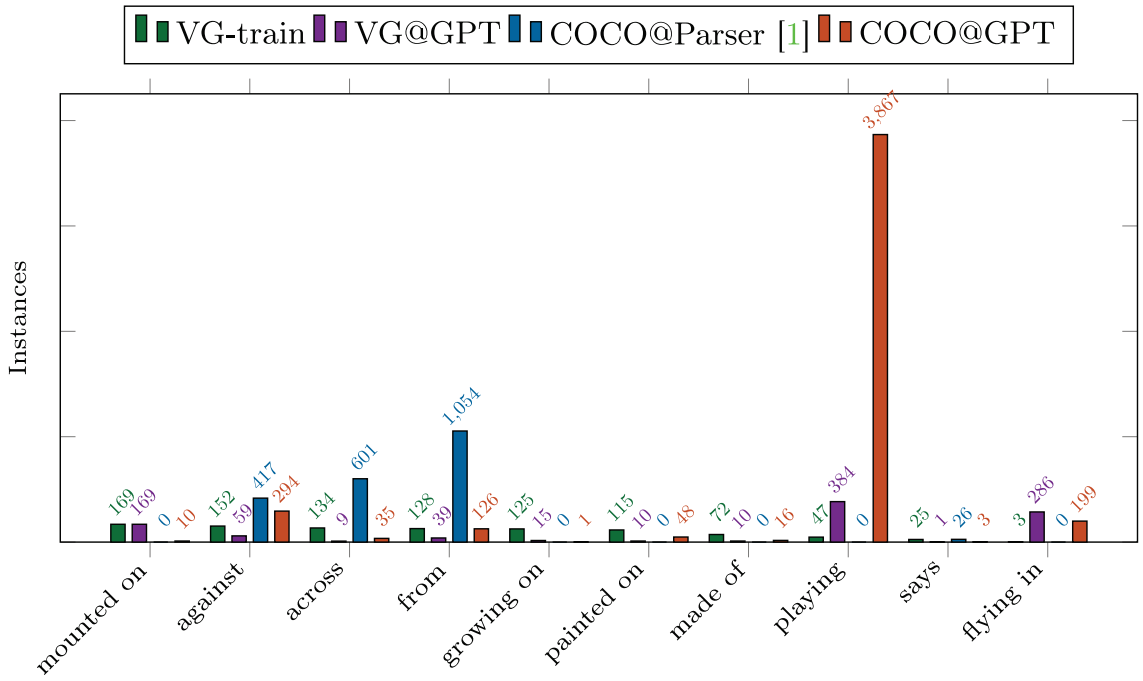
Appendix A

Weakly-Supervised SGG with Language Models

Detailed statistics for the head-10 and tail-10 categories in VG@GPT, COCO@Parser, and COCO@GPT can be found in Figure A.1. It can be found that COCO@Parser is biased to conjunction words (*e.g.*, "with", "in", "from"), while less focus on interactions like "holding", "wearing", *etc.* This defect makes it hard for the SGG model to learn complex interactions. Conversely, COCO@GPT offers rich instances of complex relationships.

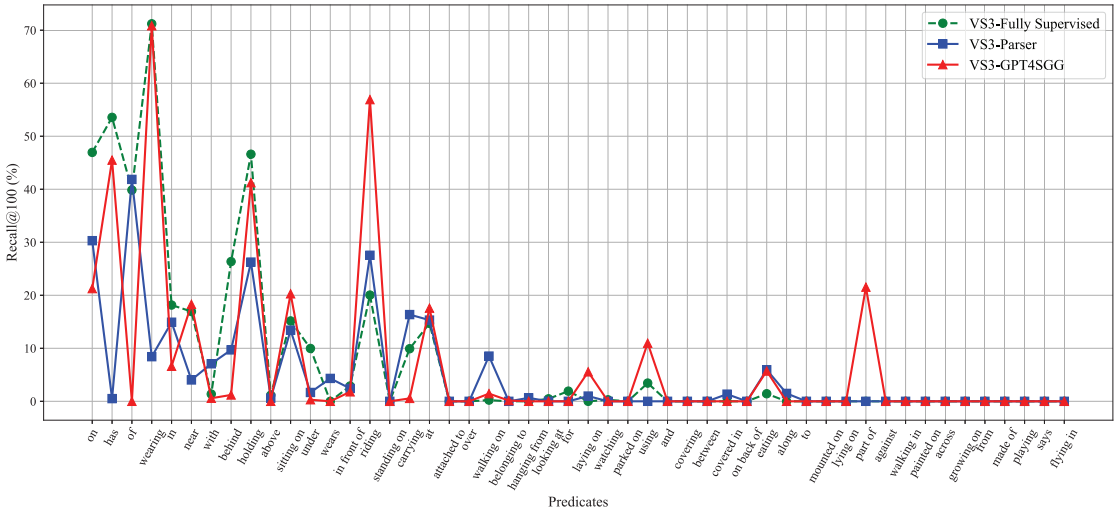


(a) Frequency of head-10 predicates.

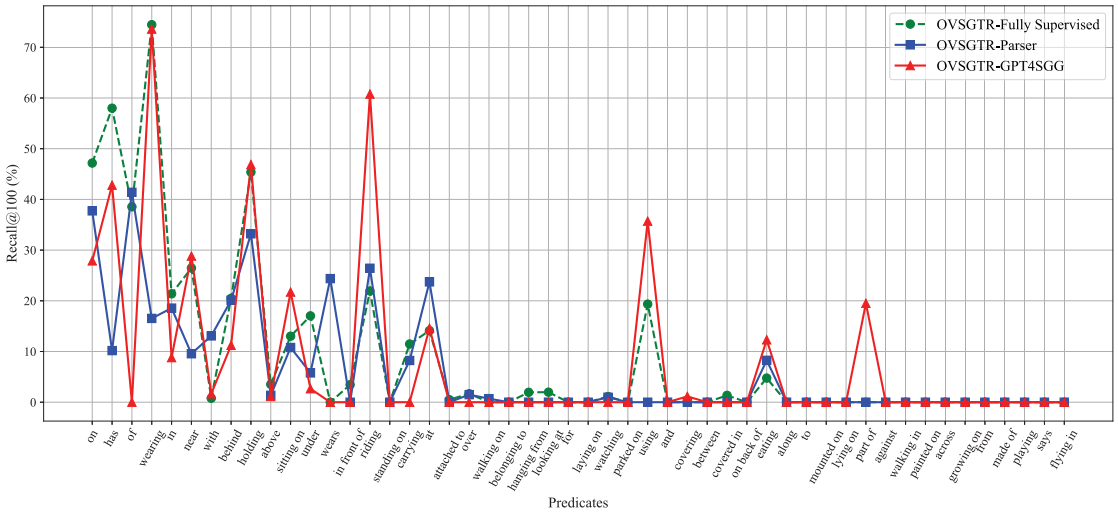


(b) Frequencey of tail-10 predicates.

Figure A.1: Frequency of head-10 (a) / tail-10 (b) predicates from VG-train, compared across VG@GPT, COCO@Parser[90], and COCO@GPT datasets.



(a) Recall@100 per predicate of VS3.



(b) Recall@100 per predicate of OvSGTR.

Figure A.2: Comparison of three methods (best view via zoom in): fully supervised SGG, parser-based language supervised SGG, and *GPT4SGG*. Models are tested on VG150 test set , and predicates of x-axis are sorted by their frequencies.

Table A.1: Prompt for synthesizing scene graphs. We only provide an output example in the instruction, while the input example is excluded for no bias introduction.

messages = [{ **"role"**: "system", **"content"**: "You are a helpful AI visual assistant. Now, you are seeing image data. Each image provides a set of objects, and a set of captions for global and localized descriptions." },

{ **"role"**: "user", **"content"**: f"""Extract relationship triplets from image data, each characterized by a unique "image_id", image dimensions, a set of objects consisting of categories (formatted as "[category].[number]") and bounding boxes (in "xyxy" format). Each image data includes a global description for the entire image and localized descriptions for specific regions (notated as "Union(name1:box1, name2:box2)", keys with ";" in captions like "Union(name1:box1, name2:box2); Union(name3:box3, name4:box4)" refer to multiple union regions share the same caption).

Here are the requirements for the task: 1. Process each image individually: Focus on one image at a time and give a comprehensive output for that specific image before moving to the next. 2. Infer interactions and spatial relationships: Utilize objects' information and both global and localized descriptions to determine relationships between objects(*e.g.*, "next to", "holding", "held by", etc.). 3. Maintain logical consistency: Avoid impossible or nonsensical relationships (*e.g.*, a person cannot be riding two different objects simultaneously, a tie cannot be worn by two persons, etc.). 4. Eliminate duplicate entries: Each triplet in the output must be unique and non-repetitive. 5. Output should be formatted as a list of dicts in JSON format, containing "image_id" and "relationships" for each image.

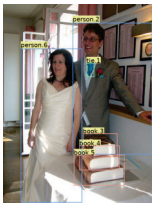
Example output: ' ["image_id": "123456", "relationships": ["source": "person.1", "target": "skateboard.2", "relation": "riding", "source": "person.4", "target": "shirt.3", "relation": "wearing", "source": "person.2", "target": "bottle.5", "relation": "holding", "source": "person.4", "target": "bus.1", "relation": "near",] , "image_id": "23455", "relationships": ["source": "man.1", "target": "car.1", "relation": "driving"]] '

Ensure that each image's data is processed and outputted separately to maintain clarity and accuracy in the relationship analysis.

Input: " {Input} " ### Output: } """]

example_input = {"image_id": "227884", "width": 444, "height": 640, "objects": ["tie.1:[217, 409, 233, 436]", "tie.2:[212, 409, 233, 507]", "person.3:[119, 289, 300, 523]"], "captions": {"global": "a man wearing a suit", "Union(tie.1:[217, 409, 233, 436], tie.2:[212, 409, 233, 507])": "a purple and black cat sitting on a window ledge", "Union(tie.2:[212, 409, 233, 507], person.3:[119, 289, 300, 523]) ; Union(tie.1:[217, 409, 233, 436], person.3:[119, 289, 300, 523])": "a man in a suit and tie sitting at a table with a laptop"} }

Table A.2: Example of synthesising scene graphs with GPT-4.



Example Input : {"image_id": "395890", "width": 480, "height": 640, "objects": ["tie.1:[269, 189, 293, 234]", "person.2:[224, 60, 480, 483]", "book.3:[257, 416, 368, 492]", "book.4:[246, 455, 375, 534]", "book.5:[228, 485, 391, 583]", "person.6:[57, 143, 254, 638]"], "captions": { "Union(person.2:[224, 60, 480, 483], book.3:[257, 416, 368, 492])": "a man and a woman standing next to a cake", "Union(book.3:[257, 416, 368, 492], book.4:[246, 455, 375, 534])": "a cake made of books", "Union(book.3:[257, 416, 368, 492], book.5:[228, 485, 391, 583])": "a man standing next to a cake that is made of books", "Union(book.4:[246, 455, 375, 534], book.5:[228, 485, 391, 583])": "a cake made out of books", "Union(book.5:[228, 485, 391, 583], person.6:[57, 143, 254, 638])": "a man and a woman", "Union(book.4:[246, 455, 375, 534], person.6:[57, 143, 254, 638])": "a man and a woman standing in front of a cake", "global ; Union(person.2:[224, 60, 480, 483], person.6:[57, 143, 254, 638]) ; Union(tie.1:[269, 189, 293, 234], person.2:[224, 60, 480, 483]) ; Union(person.2:[224, 60, 480, 483], book.4:[246, 455, 375, 534])": "a man and woman standing next to a cake"}}

GPT-4 Response : { "image_id": "395890", "relationships": [{"source": "person.2", "target": "book.3", "relation": "near"}, {"source": "person.2", "target": "person.6", "relation": "near"}, {"source": "person.2", "target": "tie.1", "relation": "wearing"}, {"source": "person.6", "target": "book.4", "relation": "near"}, {"source": "person.6", "target": "book.5", "relation": "near"}, {"source": "book.3", "target": "book.4", "relation": "on"}, {"source": "book.4", "target": "book.5", "relation": "on"}] }

User: How would you infer that "person.2" rather than "person.6" is "wearing" the "tie.1" ?

GPT-4: To infer that "person.2" rather than "person.6" is wearing "tie.1", the following observations from the image data can be considered:
Spatial Proximity: The bounding box of "tie.1" ([269, 189, 293, 234]) is entirely within the bounding box of "person.2" ([224, 60, 480, 483]). This indicates that the tie is located on or very near to person.2. In contrast, the bounding box of "person.6" ([57, 143, 254, 638]) does not overlap with that of the tie, suggesting that person.6 is not wearing it. Global and Localized Descriptions: The global description mentions a "man and woman standing next to a cake," but does not specify who is wearing the tie. However, the union of "tie.1" and "person.2" in the captions suggests a relationship between them. Since ties are typically worn as part of clothing, the logical inference is that person.2 is wearing the tie. Logical Consistency: In typical scenarios, a tie is worn by one person at a time. Given that the tie's location coincides with the location of person.2 and not with person.6, it is logical to conclude that person.2 is the one wearing the tie. In conclusion, the spatial arrangement of the objects and the given descriptions lead to the inference that "person.2" is wearing "tie.1".

Table A.3: Prompt for building a lookup table with GPT-4. Set A is the set to be matched with Set B (VG150 relationships).

messages = [{ **"role"**: "user", **"content"**:
"""Given the relationships in Set A: ‘ {Set A} ’, and Set B: ‘ [”above”, ”across”, ”against”, ”along”, ”and”, ”at”, ”attached to”, ”behind”, ”belonging to”, ”between”, ”carrying”, ”covered in”, ”covering”, ”eating”, ”flying in”, ”for”, ”from”, ”growing on”, ”hanging from”, ”has”, ”holding”, ”in”, ”in front of”, ”laying on”, ”looking at”, ”lying on”, ”made of”, ”mounted on”, ”near”, ”of”, ”on”, ”on back of”, ”over”, ”painted on”, ”parked on”, ”part of”, ”playing”, ”riding”, ”says”, ”sitting on”, ”standing on”, ”to”, ”under”, ”using”, ”walking in”, ”walking on”, ”watching”, ”wearing”, ”wears”, ”with”] ’, Please associate the two sets based on semantic comparison.
1) We should visit each item of set A only once and find the associated item in set B.
2) Use a direction flag where 1 indicates the same meaning, 2 indicates a weakly similar meaning, -1 indicates an antonym, and 0 refers to no associated item.
3) Format the result as a dictionary with keys ”source”, ”target”, ”direction” . ”source” must be from set A , and ”target” must be from set B or None if there is no association.
4) Provide the complete associations as a list of dicts in JSON format.
Example output:
‘‘‘
[{”source”: ”on”, ”target”: ”on”, ”direction”: 1},
{”source”: ”next to”, ”target”: ”near”, ”direction”: 1},
{”source”: ”ridden by”, ”target”: ”riding”, ”direction”: -1}]
‘‘‘
Output for Set A: """ }]

Algorithm 2 Generate Scene Graph

```
import google.generativeai as genai

generation_config = {"temperature": 0.7, "top_p": 0.95, "top_k": 64,
    "max_output_tokens": 8192,"response_mime_type": "application/
      json",
}
# Load the generative model
model = genai.GenerativeModel("gemini-1.5-flash", generation_config=
  generation_config)
prompt_template = """Given a set of detected objects in an image,
  each object is characterized by a name, a bounding box in "(xmin,
  ymin, xmax, ymax)" format. Please generate a scene graph to
  describe this image. The scene graph should describe relationships
  in the format "source -> relation -> target". Example Output:\n{"
  relationships": [{"source": "object_id1", "target": "object_id2", "
  relation":\n"relation_type"}, ... ]}\n Now, objects are {OBJECTS}.
  The original width and height of the provided image are {IMG_WH}.
  Please output the scene graph in JSON style without any comments.
  """
def annotate(image_name):
    # Load image and get its dimensions
    image = Image.open(image_name)
    image_wh = (image.width, image.height)
    # Load objects in the image,
    # e.g., {["sports ball.1:[312, 360, 370, 417]", "person.2:[116,
      49, 309, 491]", "person.3:[367, 108, 550, 477]"]}
    image_objects = load_objects(image_name)
    # Construct text prompt
    text_prompt = prompt_template.replace("OBJECTS", str(
      image_objects)).replace(
        "IMG_WH", str(image_wh))
    # Generate scene graph using generative model
    response = model.generate([image, text_prompt])
    return response
```

Appendix B

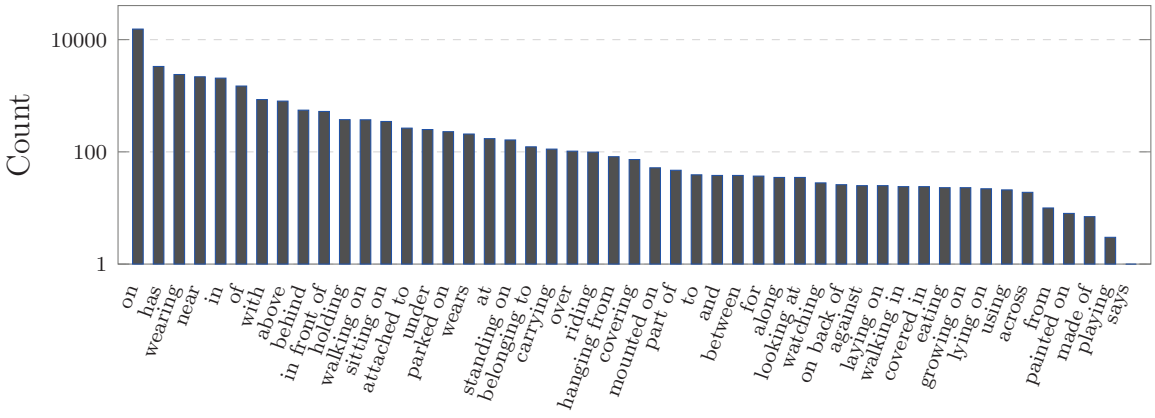
Reinforcement Learning for Scene Graph Generation

B.1 Prompt Templates for SGG

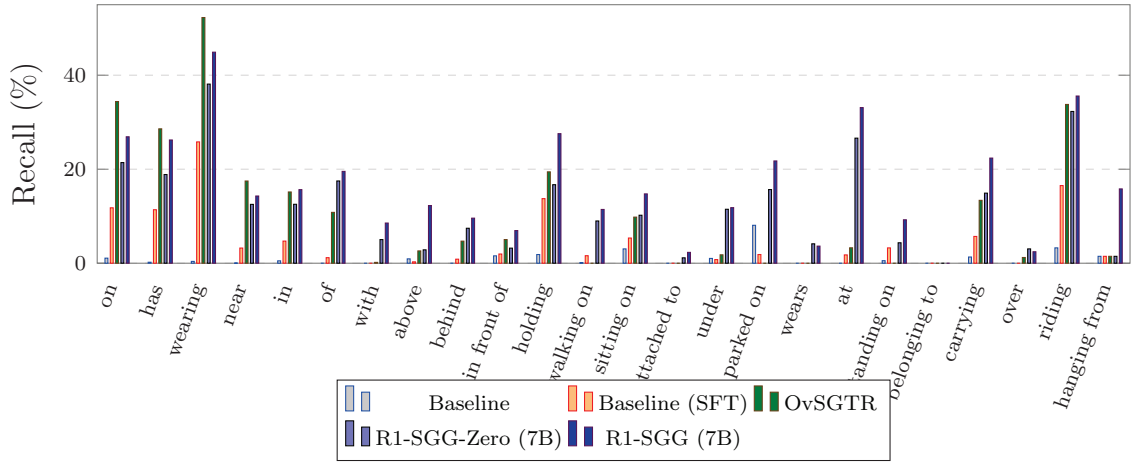
In this work, we adopt two prompt templates for scene graph generation, as illustrated in Table B.2 and Table B.1. The difference lies in whether predefined object classes and relation categories are provided.

B.2 Qualitative Results

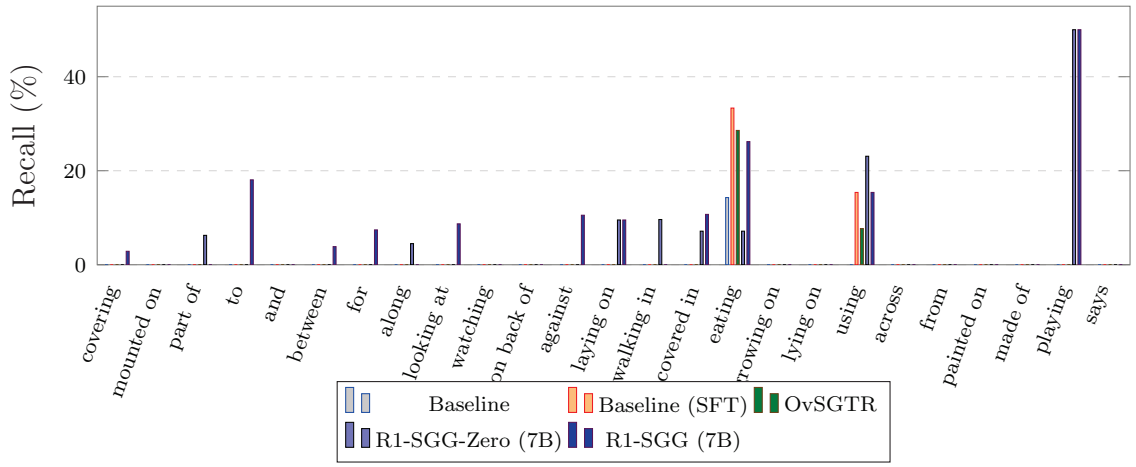
We present qualitative results in Fig. 5.3 and Fig. 5.4, and analyze head and tail predicate performance in Fig. B.1 and Fig. B.2 to assess long-tail bias. As shown in Fig. B.1, both specific models such as OvSGTR and M-LLMs like Qwen2-VL-7B-Instruct (with or without SFT) tend to be biased toward head classes, whereas R1-SGG achieves significantly higher recall on tail predicates. This trend is also confirmed on the PSG dataset in Fig. B.2. These results demonstrate that R1-SGG



(a) Histogram of predicate frequency in the VG150 validation set.

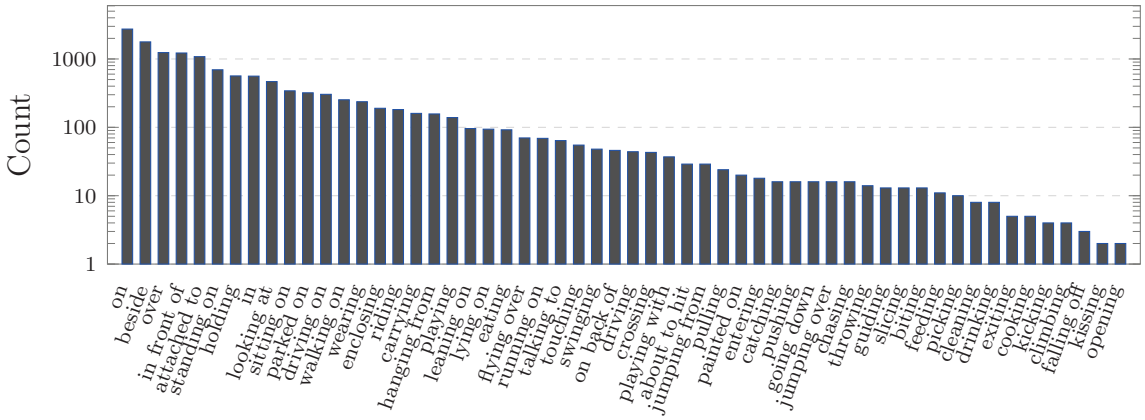


(b) Recall scores of top-24 predicates of the VG150 validation set.

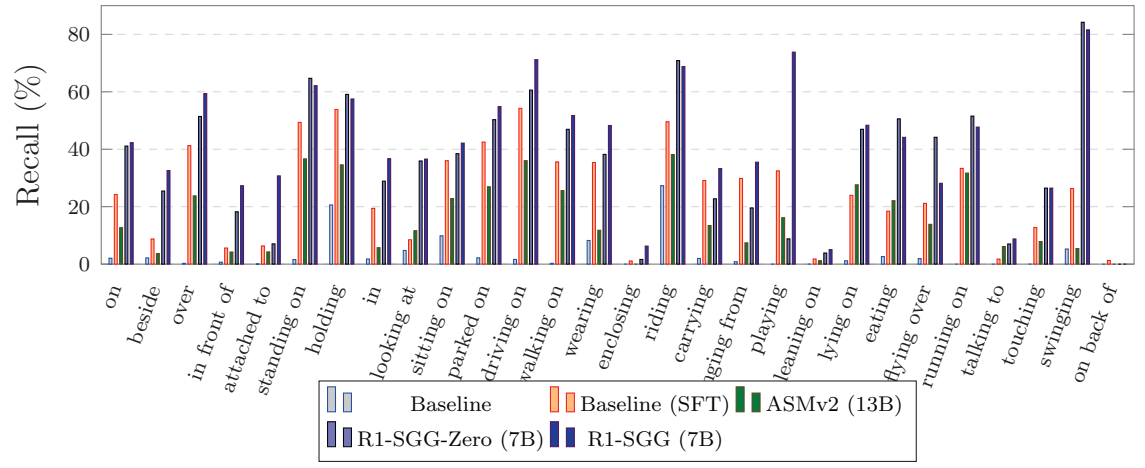


(c) Recall scores of tail-25 predicates of the VG150 validation set.

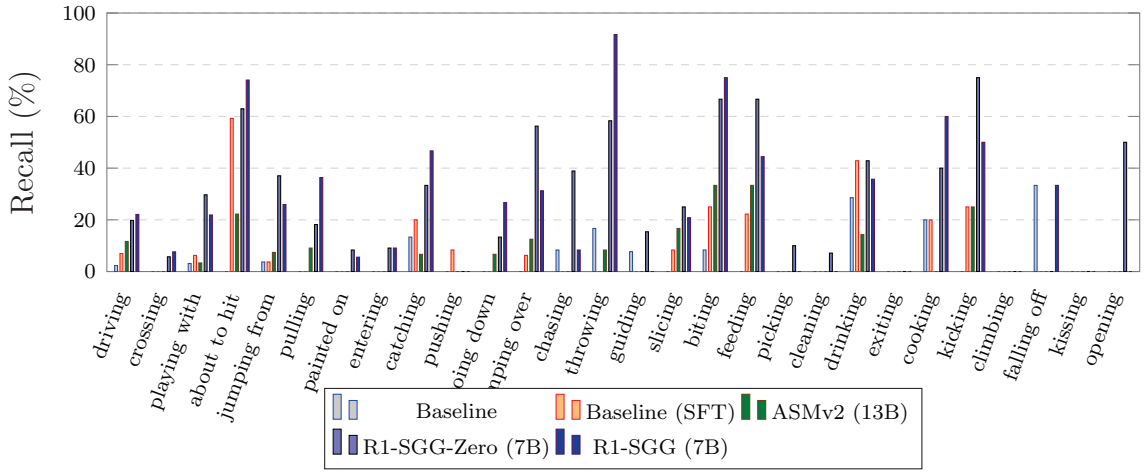
Figure B.1: Comparison of predicate frequency and predicate-wise recall on the VG150 validation set. Here, *Baseline* refers to `Qwen2-VL-7B-Instruct`.



(a) Histogram of predicate frequency in the PSG test set.



(b) Recall scores of top-28 predicates of the PSG test set.



(c) Recall scores of tail-28 predicates of the PSG test set.

Figure B.2: Comparison of predicate frequency and predicate-wise recall on the PSG test set. Here, *Baseline* refers to `Qwen2-VL-7B-Instruct`.

Table B.1: Prompting an M-LLM to generate scene graphs without providing predefined object classes or predicate types.

messages = [{ **"role"**: "system", **"content"**: " {system_prompt}" }, { **"role"**: "user", **"content"**: f"""Generate a structured scene graph for an image using the following format: “‘json { ”objects”: [{”id”: ”object_name.number”, ”bbox”: [x1, y1, x2, y2]}, ...], ”relationships”: [{”subject”: ”object_name.number”, ”predicate”: ”relationship_type”, ”object”: ”object_name.number”}, ...] }“‘. ### **Guidelines:** - **Objects:** - Assign a unique ID for each object using the format ”object_name.number” (e.g., ”person.1”, ”bike.2”). - Provide its bounding box ‘[x1, y1, x2, y2]’ in integer pixel format. - Include all visible objects, even if they have no relationships.
- **Relationships:** - Represent interactions accurately using ”subject”, ”predicate”, and ”object”. - Omit relationships for orphan objects.
Example Output: “‘json { ”objects”: [{”id”: ”person.1”, ”bbox”: [120, 200, 350, 700]}, {”id”: ”bike.2”, ”bbox”: [100, 600, 400, 800]}, {”id”: ”helmet.3”, ”bbox”: [150, 150, 280, 240]}, {”id”: ”tree.4”, ”bbox”: [500, 100, 750, 700]}], ”relationships”: [{”subject”: ”person.1”, ”predicate”: ”riding”, ”object”: ”bike.2”}, {”subject”: ”person.1”, ”predicate”: ”wearing”, ”object”: ”helmet.3”}] } “‘ Now, generate the complete scene graph for the provided image: """ }]

is more effective at generating unbiased scene graphs.

Table B.2: Prompting an M-LLM to generate scene graphs with predefined object classes and predicate types. Here, *OBJ_CLS* and *REL_CLS* refer to the predefined object classes and relation categories respectively.

messages = [{ **"role"**: "system", **"content"**: " {system_prompt}" }, { **"role"**: "user", **"content"**: f"""Generate a structured scene graph for an image using the following format: “‘json { ”objects”: [{”id”: ”object_name.number”, ”bbox”: [x1, y1, x2, y2]}, ...], ”relationships”: [{”subject”: ”object_name.number”, ”predicate”: ”relationship_type”, ”object”: ”object_name.number”}, ...] }“‘. ### **Guidelines:** - **Objects:** - Assign a unique ID for each object using the format ”object_name.number” (e.g., ”person.1”, ”bike.2”). The **object_name** must belong to the predefined object set: ‘{OBJ_CLS}’. - Provide its bounding box ‘[x1, y1, x2, y2]’ in integer pixel format. - Include all visible objects, even if they have no relationships.
- **Relationships:** - Represent interactions accurately using ”subject”, ”predicate”, and ”object”. - Omit relationships for orphan objects. - The **predicate** must belong to the predefined relationship set: ‘{REL_CLS}’. ### **Example Output:** “‘json { ”objects”: [{”id”: ”person.1”, ”bbox”: [120, 200, 350, 700]}, {”id”: ”bike.2”, ”bbox”: [100, 600, 400, 800]}, {”id”: ”helmet.3”, ”bbox”: [150, 150, 280, 240]}, {”id”: ”tree.4”, ”bbox”: [500, 100, 750, 700]}], ”relationships”: [{”subject”: ”person.1”, ”predicate”: ”riding”, ”object”: ”bike.2”}, {”subject”: ”person.1”, ”predicate”: ”wearing”, ”object”: ”helmet.3”}] } “‘ Now, generate the complete scene graph for the provided image: """ }]

Appendix C

Image Generation with Scene Graphs

Scene Diversity. To classify the scene categories, we utilize an LLM (*e.g.*, Gemini 1.5 Flash [104]) to perform such classification. The prompt used here is *Now, we have a list of image information like {IMAGE_INFO} , where each image information contains "xyxy" bounding boxes and "relationships" depicting the relation between the "source" object and the "target" object. Please classify the scene in **each image** using the following hierarchy: Level 1: - People-Centric, - Non-People Centric. Level 2: If People-Centric: [Choose one: Social Interaction, Individual Activities, Work/Occupation, Travel/Exploration, Sports & Recreation, Performance/Entertainment, Daily Life]; If Non-People Centric: [Choose one: Nature, Urban/Built, Objects, Abstract/Artistic]. Please provide the classification for each image in the list, and present your answer as a **JSON-formatted** list of dictionaries, where each dictionary corresponds to an image and contains the following keys: "image_id", "file_name", "level 1", "level 2".*

Figure 6.3 illustrates examples of categorized scenes in MegaSG, showcasing the diversity and range of scenarios covered in the dataset.

Table C.1: Prompt for refining object categories with attributes. The LLM receives a list of object categories as input and outputs corresponding fine-grained object descriptions.

messages = [{ `"role": "user", "content":` "Given a file listing object categories, generate 3 realistic instances for each category by adding common-sense visual attributes such as color, size, material, or texture. Ensure that the attributes are contextually appropriate. For example, transform 'apple' into 'red apple', 'green apple', and 'ripe apple'; or 'car' into 'white car', 'small car', and 'luxury car'. The generated instances should be diverse and plausible. formulate the output as a dict in JSON like {"apple": ["red apple", "green apple", "ripe apple"], "car":["white car", "black car", "luxury car"]} }];

Scene Graph Representation. For text-to-image (T2I) models that condition on a sentence, we encode scene graphs in the format `{subject} {predicate} {object}` (e.g., `cat sitting on desk, dog near chair`). The prompt used to convert a scene graph into a consistent description (*i.e.*, the scene composition described in Section 6.2) is provided in Table C.3.

Table C.2: Prompt for extracting a textual scene graph from a caption.

messages = [{ `"role": "user", "content":` "Extract a structured scene graph from the following sentence. In this scene graph, include all physical objects even if they do not participate in any relationships (i.e., allow orphan nodes). The output should be a dictionary with two keys: - "objects": a list of all physical objects mentioned in the sentence. - "relationships": a list of dictionaries, each containing: - "source": the subject (a physical object) of the relationship. - "target": the object (a physical object) of the relationship. - "relation": the predicate describing the relationship between the source and target. Ensure that both "source" and "target" are physical objects (not abstract concepts).
Sentence: "'{input_sentence}"'
Output format: { "objects": [...], "relationships": ["source": "...", "target": "...", "relation": "...", ...] }
];

Table C.3: Prompt for scene composition: translating a concise scene graph into a coherent and descriptive text.

messages = [{ `"role": "user", "content":` "You are an AI assistant tasked with converting scene graphs into descriptive text prompts for image generation using diffusion models. Given the input scene graph, generate a detailed text description adhering to the following instructions:

Instructions:

1. **Scene Setup:** Begin by describing the overall setting or background of the scene. If no explicit background object exists, infer a suitable one from the present objects and relationships.

2. **Object Placement:** Introduce each object from the 'objects' list. When describing their placement, utilize the 'relationships' to accurately depict their positions relative to each other and the scene.

3. **Relationship Emphasis:** Vividly express the relationships between objects using descriptive language. Avoid simply stating the relationship; instead, showcase it through the objects' placement, appearance, or actions.

4. **Image-ability:** Craft the description to be easily translatable into a visual representation. Use evocative language that captures the essence of the scene and guides the diffusion model.

5. **Conciseness and Clarity:** Be concise and avoid unnecessary details. Ensure the language is unambiguous and accurately reflects the scene graph information.

Output Format: Provide the output as a JSON object with a single key `"description"` and the value being the generated text description.

Input Scene Graph (JSON format):

```json {scene_graph} ```

Output:" }];

example_input = ['objects': ['cymbal.1', 'cymbal.2', 'drum.3', 'guitar.4', 'microphone.5', 'person.6'], 'relationships': [{'source': 'person.6', 'target': 'guitar.4', 'relation': 'playing'}, {'source': 'person.6', 'target': 'microphone.5', 'relation': 'using'}, {'source': 'guitar.4', 'target': 'cymbal.1', 'relation': 'near'}, {'source': 'guitar.4', 'target': 'cymbal.2', 'relation': 'near'}, {'source': 'guitar.4', 'target': 'drum.3', 'relation': 'near'}]];

example_output= "A musician stands on a stage, the bright lights reflecting off their instruments. They are playing a guitar, their fingers dancing across the strings. A microphone is positioned in front of them, capturing their performance. The guitar rests near a pair of cymbals, and a drum sits nearby, adding to the musical ensemble."
```

References

[1] Google AI. Gemini 1.5 flash pricing. https://ai.google.dev/pricing#1_5flash, 2024.

[2] Oron Ashual and Lior Wolf. Specifying object attributes and relations in interactive scene generation. In *ICCV*, pages 4561–4569, 2019.

[3] Zechen Bai, Pichao Wang, Tianjun Xiao, Tong He, Zongbo Han, Zheng Zhang, and Mike Zheng Shou. Hallucination of multimodal large language models: A survey. *CoRR*, abs/2404.18930, 2024.

[4] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. In *ICLR*, 2019.

[5] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *NeurIPS*, 33:1877–1901, 2020.

[6] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. Coco-stuff: Thing and stuff classes in context. In *CVPR*, pages 1209–1218, 2018.

[7] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, pages 213–229, 2020.

[8] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, pages 213–229, 2020.

[9] Agneet Chatterjee, Gabriela Ben Melech Stan, Estelle Aflalo, Sayak Paul, Dhruba Ghosh, Tejas Gokhale, Ludwig Schmidt, Hannaneh Hajishirzi, Vasudev Lal, Chitta Baral, and Yezhou Yang. Getting it right: Improving spatial consistency in text-to-image models. In *ECCV*, pages 204–222, 2024.

[10] Junsong Chen, Jincheng Yu, Chongjian Ge, Lewei Yao, Enze Xie, Zhongdao Wang, James T. Kwok, Ping Luo, Huchuan Lu, and Zhenguo Li. PixArt-α: Fast training of diffusion transformer for photorealistic text-to-image synthesis. In *ICLR*, 2024.

[11] Shizhe Chen, Qin Jin, Peng Wang, and Qi Wu. Say as you wish: Fine-grained control of image caption generation with abstract scene graphs. In *CVPR*, pages 9959–9968, 2020.

[12] Tianshui Chen, Weihao Yu, Riquan Chen, and Liang Lin. Knowledge-embedded routing network for scene graph generation. In *CVPR*, pages 6163–6171, 2019.

[13] Tianshui Chen, Weihao Yu, Riquan Chen, and Liang Lin. Knowledge-embedded routing network for scene graph generation. In *CVPR*, pages 6163–6171, 2019.

[14] Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO captions: Data collection and evaluation server. *CoRR*, abs/1504.00325, 2015.

[15] Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. UNITER: universal image-text representation learning. In *ECCV*, pages 104–120, 2020.

[16] Zuyao Chen, Jinlin Wu, Zhen Lei, and Chang Wen Chen. What makes a scene? scene graph-based evaluation and feedback for controllable generation. *arXiv preprint arXiv:2411.15435*, 2024.

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

[18] Zuyao Chen, Jinlin Wu, Zhen Lei, Zhaoxiang Zhang, and Changwen Chen. Expanding scene graph boundaries: Fully open-vocabulary scene graph generation via visual-concept alignment and retention. *arXiv preprint arXiv:2311.10988*, 2023.

[19] Zuyao Chen, Jinlin Wu, Zhen Lei, Zhaoxiang Zhang, and Changwen Chen. GPT4SGG: Synthesizing scene graphs from holistic and region-specific narratives. *arXiv preprint arXiv:2312.04314*, 2023.

[20] Meng-Jiun Chiou, Henghui Ding, Hanshu Yan, Changhu Wang, Roger Zimmermann, and Jiashi Feng. Recovering the unbiased scene graphs from the biased ones. In *ACMMM*, pages 1581–1590, 2021.

[21] Jaemin Cho, Yushi Hu, Jason M. Baldridge, Roopal Garg, Peter Anderson, Ranjay Krishna, Mohit Bansal, Jordi Pont-Tuset, and Su Wang. Davidsonian scene graph: Improving reliability in fine-grained evaluation for text-to-image generation. In *ICLR*, 2024.

[22] Kevin Clark and Priyank Jaini. Text-to-image diffusion models are zero shot classifiers. *NeurIPS*, 36:58921–58937, 2023.

[23] Yuren Cong, Michael Ying Yang, and Bodo Rosenhahn. Reltr: Relation transformer for scene graph generation. *IEEE TPAMI*, 45(9):11169–11183, 2023.

[24] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, pages 4171–4186, 2019.

[25] Xingning Dong, Tian Gan, Xuemeng Song, Jianlong Wu, Yuan Cheng, and Liqiang Nie. Stacked hybrid-attention and group collaborative learning for unbiased scene graph generation. In *CVPR*, pages 19427–19436, 2022.

[26] Yu Du, Fangyun Wei, Zihe Zhang, Miaojing Shi, Yue Gao, and Guoqi Li. Learning to prompt for open-vocabulary object detection with vision-language model. In *CVPR*, pages 14064–14073, 2022.

[27] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *ICML*, 2024.

[28] Ying Fan, Olivia Watkins, Yuqing Du, Hao Liu, Moonkyung Ryu, Craig Boutilier, Pieter Abbeel, Mohammad Ghavamzadeh, Kangwook Lee, and Kimin Lee. Reinforcement learning for fine-tuning text-to-image diffusion models. *NeurIPS*, 2024.

[29] Azade Farshad, Yousef Yeganeh, Yu Chi, Chengzhi Shen, Böjrn Ommer, and Nassir Navab. Scenegenie: Scene graph guided diffusion models for image synthesis. In *ICCVW*, pages 88–98, 2023.

[30] Weixi Feng, Xuehai He, Tsu-Jui Fu, Varun Jampani, Arjun R. Akula, Pradyumna Narayana, Sugato Basu, Xin Eric Wang, and William Yang Wang. Training-free structured diffusion guidance for compositional text-to-image synthesis. In *ICLR*, 2023.

[31] Weixi Feng, Wanrong Zhu, Tsu-jui Fu, Varun Jampani, Arjun Akula, Xuehai He, Sugato Basu, Xin Eric Wang, and William Yang Wang. Layoutgpt: Com-

positional visual planning and generation with large language models. *NeurIPS*, 2024.

[32] Golnaz Ghiasi, Xiuye Gu, Yin Cui, and Tsung-Yi Lin. Scaling open-vocabulary image segmentation with image-level labels. In *ECCV*, pages 540–557, 2022.

[33] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Region-based convolutional networks for accurate object detection and segmentation. *IEEE TPAMI*, 38(1):142–158, 2015.

[34] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NeurIPS*, pages 2672–2680, 2014.

[35] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.

[36] Google AI for Developers. Gemini 2.0 flash. https://ai.google.dev/gemini-api/docs/models#gemini-2.0-flash, 2025.

[37] Jiuxiang Gu, Shafiq R. Joty, Jianfei Cai, Handong Zhao, Xu Yang, and Gang Wang. Unpaired image captioning via scene graph alignments. In *ICCV*, pages 10322–10331, 2019.

[38] Xiuye Gu, Tsung-Yi Lin, Weicheng Kuo, and Yin Cui. Open-vocabulary object detection via vision and language knowledge distillation. In *Int. Conf. Learn. Represent.*, 2022.

[39] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

[40] Tao He, Lianli Gao, Jingkuan Song, and Yuan-Fang Li. Towards open-vocabulary scene graph generation with prompt-based finetuning. In *ECCV*, pages 56–73, 2022.

[41] Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. CLIPScore: A reference-free evaluation metric for image captioning. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, *EMNLP*, pages 7514–7528, 2021.

[42] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *NeurIPS*, 2017.

[43] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *NeurIPS*, 33:6840–6851, 2020.

[44] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. In *ICLR*, 2022.

[45] Yushi Hu, Benlin Liu, Jungo Kasai, Yizhong Wang, Mari Ostendorf, Ranjay Krishna, and Noah A Smith. Tifa: Accurate and interpretable text-to-image faithfulness evaluation with question answering. In *ICCV*, pages 20406–20417, 2023.

[46] Drew A Hudson and Christopher D Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *CVPR*, pages 6700–6709, 2019.

[47] Aaron Hurst, Adam Lerer, Adam P. Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, Aleksander Madry, Alex Baker-Whitcomb, Alex Beutel, Alex Borzunov, Alex Carney, Alex Chow, Alex Kirillov, Alex Nichol, Alex Paino, Alex Renzin,

Alex Tachard Passos, Alexander Kirillov, Alexi Christakis, Alexis Conneau, Ali Kamali, Allan Jabri, Allison Moyer, Allison Tam, Amadou Crookes, Amin Tootoonchian, Ananya Kumar, Andrea Vallone, Andrej Karpathy, Andrew Braunstein, Andrew Cann, Andrew Codispoti, Andrew Galu, Andrew Kondrich, Andrew Tulloch, Andrey Mishchenko, Angela Baek, Angela Jiang, Antoine Pelisse, Antonia Woodford, Anuj Gosalia, Arka Dhar, Ashley Pantuliano, Avi Nayak, Avital Oliver, Barret Zoph, Behrooz Ghorbani, Ben Leimberger, Ben Rossen, Ben Sokolowsky, Ben Wang, Benjamin Zweig, Beth Hoover, Blake Samic, Bob McGrew, Bobby Spero, Bogo Giertler, Bowen Cheng, Brad Lightcap, Brandon Walkin, Brendan Quinn, Brian Guarraci, Brian Hsu, Bright Kellogg, Brydon Eastman, Camillo Lugaresi, Carroll L. Wainwright, Cary Bassin, Cary Hudson, Casey Chu, Chad Nelson, Chak Li, Chan Jun Shern, Channing Conger, Charlotte Barette, Chelsea Voss, Chen Ding, Cheng Lu, Chong Zhang, Chris Beaumont, Chris Hallacy, Chris Koch, Christian Gibson, Christina Kim, Christine Choi, Christine McLeavey, Christopher Hesse, Claudia Fischer, Clemens Winter, Coley Czarnecki, Colin Jarvis, Colin Wei, Constantin Koumouzelis, and Dane Sherburn. Gpt-4o system card. *CoRR*, abs/2410.21276, 2024.

[48] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *CVPR*, pages 1125–1134, 2017.

[49] Jaehyeong Jeon, Kibum Kim, Kanghoon Yoon, and Chanyoung Park. Semantic diversity-aware prototype-based learning for unbiased scene graph generation. In Ales Leonardis, Elisa Ricci, Stefan Roth, Olga Russakovsky, Torsten Sattler, and Gül Varol, editors, *ECCV*, volume 15126, pages 379–395, 2024.

[50] Tianlei Jin, Fangtai Guo, Qiwei Meng, Shiqiang Zhu, Xiangming Xi, Wen Wang, Zonghao Mu, and Wei Song. Fast contextual scene graph generation with un-

biased context augmentation. In *CVPR*, pages 6302–6311, 2023.

[51] Justin Johnson, Agrim Gupta, and Li Fei-Fei. Image generation from scene graphs. In *CVPR*, pages 1219–1228, 2018.

[52] Justin Johnson, Ranjay Krishna, Michael Stark, Li-Jia Li, David A. Shamma, Michael S. Bernstein, and Li Fei-Fei. Image retrieval using scene graphs. In *CVPR*, pages 3668–3678, 2015.

[53] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *CVPR*, pages 4401–4410, 2019.

[54] Bahjat Kawar, Shiran Zada, Oran Lang, Omer Tov, Huiwen Chang, Tali Dekel, Inbar Mosseri, and Michal Irani. Imagic: Text-based real image editing with diffusion models. In *CVPR*, pages 6007–6017, 2023.

[55] Franklin Kenghagho Kenfack, Feroz Ahmed Siddiky, Ferenc Balint-Benczedi, and Michael Beetz. Robotvqa - A scene-graph- and deep-learning-based visual question answering system for robot manipulation. In *IROS*, pages 9667–9674, 2020.

[56] Siddhesh Khandelwal and Leonid Sigal. Iterative scene graph generation. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh, editors, *NeurIPS*, 2022.

[57] Kibum Kim, Kanghoon Yoon, Jaehyeong Jeon, Yeonjun In, Jinyoung Moon, Donghyun Kim, and Chanyoung Park. Llm4sgg: Large language models for weakly supervised scene graph generation. In *CVPR*, pages 28306–28316, 2024.

[58] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In Yoshua Bengio and Yann LeCun, editors, *ICLR*, 2014.

[59] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al.

Visual genome: Connecting language and vision using crowdsourced dense image annotations. *IJCV*, 123:32–73, 2017.

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

[61] Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *ACM SIGOPS*, 2023.

[62] Kimin Lee, Hao Liu, Moonkyung Ryu, Olivia Watkins, Yuqing Du, Craig Boutilier, Pieter Abbeel, Mohammad Ghavamzadeh, and Shixiang Shane Gu. Aligning text-to-image models using human feedback. *CoRR*, abs/2302.12192, 2023.

[63] Soohyeong Lee, Ju-Whan Kim, Youngmin Oh, and Joo Hyuk Jeon. Visual question answering over scene graph. In *GC*, pages 45–50, 2019.

[64] Chenliang Li, Haiyang Xu, Junfeng Tian, Wei Wang, Ming Yan, Bin Bi, Jiabo Ye, He Chen, Guohai Xu, Zheng Cao, Ji Zhang, Songfang Huang, Fei Huang, Jingren Zhou, and Luo Si. mplug: Effective and efficient vision-language learning by cross-modal skip-connections. In *EMNLP*, pages 7241–7259, 2022.

[65] Jiankai Li, Yunhong Wang, Xiefan Guo, Ruijie Yang, and Weixin Li. Leveraging predicate and triplet learning for scene graph generation. In *CVPR*, pages 28369–28379, 2024.

[66] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *ICML*, pages 19730–19742, 2023.

[67] Junnan Li, Dongxu Li, Silvio Savarese, and Steven C. H. Hoi. BLIP-2: bootstrapping language-image pre-training with frozen image encoders and large language models. In *ICML*, volume 202, pages 19730–19742, 2023.

[68] Junnan Li, Dongxu Li, Silvio Savarese, and Steven C. H. Hoi. BLIP-2: bootstrapping language-image pre-training with frozen image encoders and large language models. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *ICML*, pages 19730–19742, 2023.

[69] Liunian Harold Li, Pengchuan Zhang, Haotian Zhang, Jianwei Yang, Chunyuan Li, Yiwu Zhong, Lijuan Wang, Lu Yuan, Lei Zhang, Jenq-Neng Hwang, Kai-Wei Chang, and Jianfeng Gao. Grounded language-image pre-training. In *CVPR*, pages 10955–10965, 2022.

[70] Rongjie Li, Songyang Zhang, and Xuming He. Sgtr: End-to-end scene graph generation with transformer. In *CVPR*, pages 19464–19474, 2022.

[71] Rongjie Li, Songyang Zhang, Dahua Lin, Kai Chen, and Xuming He. From pixels to graphs: Open-vocabulary scene graph generation with vision-language models. In *CVPR*, pages 28076–28086, 2024.

[72] Rongjie Li, Songyang Zhang, Bo Wan, and Xuming He. Bipartite graph network with adaptive message passing for unbiased scene graph generation. In *CVPR*, pages 11109–11119, 2021.

[73] Xingchen Li, Long Chen, Wenbo Ma, Yi Yang, and Jun Xiao. Integrating object-aware and interaction-aware knowledge for weakly supervised scene graph generation. In *ACMMM*, pages 4204–4213, 2022.

[74] Yikang Li, Wanli Ouyang, Bolei Zhou, Kun Wang, and Xiaogang Wang. Scene graph generation from objects, phrases and region captions. In *ICCV*, pages 1270–1279, 2017.

[75] Long Lian, Boyi Li, Adam Yala, and Trevor Darrell. Llm-grounded diffusion: Enhancing prompt understanding of text-to-image diffusion models with large language models. *arXiv preprint arXiv:2305.13655*, 2023.

[76] Tsung-Yi Lin, Priya Goyal, Ross B. Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, pages 2999–3007, 2017.

[77] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, pages 740–755, 2014.

[78] Xin Lin, Changxing Ding, Jinquan Zeng, and Dacheng Tao. Gps-net: Graph property sensing network for scene graph generation. In *CVPR*, pages 3746–3753, 2020.

[79] Xin Lin, Changxing Ding, Yibing Zhan, Zijian Li, and Dacheng Tao. Hl-net: Heterophily learning network for scene graph generation. In *CVPR*, pages 19454–19463, 2022.

[80] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning, 2023.

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

[82] Hengyue Liu, Ning Yan, Masood S. Mortazavi, and Bir Bhanu. Fully convolutional scene graph generation. In *CVPR*, pages 11546–11556, 2021.

[83] Jinxiu Liu and Qi Liu. R3CD: Scene graph to image generation with relation-aware compositional contrastive control diffusion. In *AAAI*, pages 3657–3665, 2024.

[84] Minghao Liu, Le Zhang, Yingjie Tian, Xiaochao Qu, Luoqi Liu, and Ting Liu. Draw like an artist: Complex scene generation with diffusion model via composition, painting, and retouching. *arXiv preprint arXiv:2408.13858*, 2024.

[85] Nan Liu, Shuang Li, Yilun Du, Antonio Torralba, and Joshua B Tenenbaum. Compositional visual generation with composable diffusion models. In *ECCV*, pages 423–439, 2022.

[86] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, and Lei Zhang. Grounding DINO: marrying DINO with grounded pre-training for open-set object detection. *CoRR*, abs/2303.05499, 2023.

[87] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, pages 9992–10002, 2021.

[88] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019.

[89] Shilin Lu, Yanzhu Liu, and Adams Wai-Kin Kong. Tf-icon: Diffusion-based training-free cross-domain image composition. In *ICCV*, pages 2294–2305, 2023.

[90] Jiayuan Mao. Scene graph parser. https://github.com/vacancy/SceneGraphParser, 2022.

[91] George A Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.

[92] Kien Nguyen, Subarna Tripathi, Bang Du, Tanaya Guha, and Truong Q. Nguyen. In defense of scene graphs for image captioning. In *ICCV*, pages 1387–1396, 2021.

[93] Van-Quang Nguyen, Masanori Suganuma, and Takayuki Okatani. Grit: Faster and better image captioning transformer using dual visual features. In *ECCV*, pages 167–184, 2022.

[94] Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. Glide: Towards photo-realistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741*, 2021.

[95] Sai Vidyaranya Nuthalapati, Ramraj Chandradevan, Eleonora Giunchiglia, Bowen Li, Maxime Kayser, Thomas Lukasiewicz, and Carl Yang. Lightweight visual question answering using scene graphs. In *CIKM*, pages 3353–3357, 2021.

[96] OpenAI. Gpt-3.5 turbo. https://platform.openai.com/, 2023.

[97] OpenAI. GPT-4 technical report. *CoRR*, abs/2303.08774, 2023.

[98] OpenAI. Gpt api pricing. https://openai.com/api/pricing/, 2025.

[99] Dong Huk Park, Samaneh Azadi, Xihui Liu, Trevor Darrell, and Anna Rohrbach. Benchmark for compositional text-to-image synthesis. In *NeurIPS*, 2021.

[100] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. SDXL: improving latent diffusion models for high-resolution image synthesis. In *ICLR*, 2024.

[101] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark,

Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *ICML*, pages 8748–8763, 2021.

[102] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *ICML*, pages 8821–8831, 2021.

[103] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *CVPR*, pages 779–788, 2016.

[104] Machel Reid, Nikolay Savinov, Denis Teplyashin, Dmitry Lepikhin, Timothy Lillicrap, Jean-baptiste Alayrac, Radu Soricut, Angeliki Lazaridou, Orhan Firat, Julian Schrittwieser, et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*, 2024.

[105] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *NeurIPS*, 28, 2015.

[106] Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian D. Reid, and Silvio Savarese. Generalized intersection over union: A metric and a loss for bounding box regression. In *CVPR*, pages 658–666, 2019.

[107] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, pages 10684–10695, 2022.

[108] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *NeurIPS*, 29, 2016.

[109] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *NeurIPS*, 35:25278–25294, 2022.

[110] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[111] Shuai Shao, Zeming Li, Tianyuan Zhang, Chao Peng, Gang Yu, Xiangyu Zhang, Jing Li, and Jian Sun. Objects365: A large-scale, high-quality dataset for object detection. In *ICCV*, pages 8430–8439, 2019.

[112] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Y Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

[113] Guibao Shen, Luozhou Wang, Jiantao Lin, Wenhang Ge, Chaozhe Zhang, Xin Tao, Yuan Zhang, Pengfei Wan, Zhongyuan Wang, Guangyong Chen, et al. Sg-adapter: Enhancing text-to-image generation with scene graph guidance. *arXiv preprint arXiv:2405.15321*, 2024.

[114] Gopika Sudhakaran, Devendra Singh Dhami, Kristian Kersting, and Stefan Roth. Vision relation transformer for unbiased scene graph generation. In *ICCV*, pages 21882–21893, 2023.

[115] Shuzhou Sun, Shuaifeng Zhi, Qing Liao, Janne Heikkilä, and Li Liu. Unbiased scene graph generation via two-stage causal modeling. *IEEE TPAMI*, 45(10):12562–12580, 2023.

[116] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *CVPR*, pages 2818–2826, 2016.

[117] Kaihua Tang, Yulei Niu, Jianqiang Huang, Jiaxin Shi, and Hanwang Zhang. Unbiased scene graph generation from biased training. In *CVPR*, pages 3713–3722, 2020.

[118] Kaihua Tang, Hanwang Zhang, Baoyuan Wu, Wenhan Luo, and Wei Liu. Learning to compose dynamic tree structures for visual contexts. In *CVPR*, pages 6619–6628, 2019.

[119] Damien Teney, Lingqiao Liu, and Anton van den Hengel. Graph-structured representations for visual question answering. In *CVPR*, pages 3233–3241, 2017.

[120] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *CoRR*, abs/2307.09288, 2023.

[121] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

[122] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *NeurIPS*, pages 5998–6008, 2017.

[123] Patrick von Platen, Suraj Patil, Anton Lozhkov, Pedro Cuenca, Nathan Lambert, Kashif Rasul, Mishig Davaadorj, Dhruv Nair, Sayak Paul, William Berman, Yiyi Xu, Steven Liu, and Thomas Wolf. Diffusers: State-of-the-art diffusion models. https://github.com/huggingface/diffusers, 2022.

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

[125] Dalin Wang, Daniel Beck, and Trevor Cohn. On the role of scene graphs in image captioning. In *LANTERN@EMNLP-IJCNLP*, pages 29–34, 2019.

[126] Jianfeng Wang, Zhengyuan Yang, Xiaowei Hu, Linjie Li, Kevin Lin, Zhe Gan, Zicheng Liu, Ce Liu, and Lijuan Wang. Git: A generative image-to-text transformer for vision and language. *arXiv preprint arXiv:2205.14100*, 2022.

[127] Mengmeng Wang, Jiazheng Xing, and Yong Liu. Actionclip: A new paradigm for video action recognition. *CoRR*, abs/2109.08472, 2021.

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

[129] Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Yang Fan, Kai Dang, Mengfei Du, Xuancheng Ren, Rui Men, Dayiheng Liu, Chang Zhou, Jingren Zhou, and Junyang Lin. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024.

[130] Weiyun Wang, Yiming Ren, Haowen Luo, Tiantong Li, Chenxiang Yan, Zhe Chen, Wenhai Wang, Qingyun Li, Lewei Lu, Xizhou Zhu, et al. The all-seeing project v2: Towards general relation comprehension of the open world. In *ECCV*, pages 471–490. Springer, 2024.

[131] Yunnan Wang, Ziqiang Li, Wenyao Zhang, Zequn Zhang, Baao Xie, Xihui Liu, Wenjun Zeng, and Xin Jin. Scene graph disentanglement and composition for generalizable complex image generation. In *NeurIPS*, 2024.

[132] Jianzong Wu, Xiangtai Li, Shilin Xu, Haobo Yuan, Henghui Ding, Yibo Yang, Xia Li, Jiangning Zhang, Yunhai Tong, Xudong Jiang, et al. Towards open vocabulary learning: A survey. *IEEE TPAMI*, 2024.

[133] Size Wu, Wenwei Zhang, Sheng Jin, Wentao Liu, and Chen Change Loy. Aligning bag of regions for open-vocabulary object detection. In *CVPR*, pages 15254–15264, 2023.

[134] Tsung-Han Wu, Long Lian, Joseph E Gonzalez, Boyi Li, and Trevor Darrell. Self-correcting llm-controlled diffusion models. In *CVPR*, pages 6327–6336, 2024.

[135] Danfei Xu, Yuke Zhu, Christopher B. Choy, and Li Fei-Fei. Scene graph generation by iterative message passing. In *CVPR*, pages 3097–3106, 2017.

[136] Mingjie Xu, Mengyang Wu, Yuzhi Zhao, Jason Chun Lok Li, and Weifeng Ou. Llava-spacesgg: Visual instruct tuning for open-vocabulary scene graph generation with enhanced spatial relations. In *WACV*, pages 6362–6372, 2025.

[137] Tao Xu, Pengchuan Zhang, Qiuyuan Huang, Han Zhang, Zhe Gan, Xiaolei Huang, and Xiaodong He. Attngan: Fine-grained text to image generation with attentional generative adversarial networks. In *CVPR*, pages 1316–1324, 2018.

[138] Jingkang Yang, Yi Zhe Ang, Zujin Guo, Kaiyang Zhou, Wayne Zhang, and Ziwei Liu. Panoptic scene graph generation. In *ECCV*, pages 178–196. Springer, 2022.

[139] Ling Yang, Zhilin Huang, Yang Song, Shenda Hong, Guohao Li, Wentao Zhang, Bin Cui, Bernard Ghanem, and Ming-Hsuan Yang. Diffusion-based scene graph to image generation with masked contrastive pre-training. *CoRR*, abs/2211.11138, 2022.

[140] Ling Yang, Zhaochen Yu, Chenlin Meng, Minkai Xu, Stefano Ermon, and CUI Bin. Mastering text-to-image diffusion: Recaptioning, planning, and generating with multimodal llms. In *ICML*, 2024.

[141] Xu Yang, Kaihua Tang, Hanwang Zhang, and Jianfei Cai. Auto-encoding scene graphs for image captioning. In *CVPR*, pages 10685–10694, 2019.

[142] Hu Ye, Jun Zhang, Sibo Liu, Xiao Han, and Wei Yang. IP-Adapter: Text compatible image prompt adapter for text-to-image diffusion models. *CoRR*, abs/2308.06721, 2023.

[143] Keren Ye and Adriana Kovashka. Linguistic structures as weak supervision for visual scene graph generation. In *CVPR*, pages 8289–8299, 2021.

[144] Sangwoong Yoon, Woo Young Kang, Sungwook Jeon, SeongEun Lee, Changjin Han, Jonghun Park, and Eun-Sol Kim. Image-to-image retrieval by learning similarity between scene graphs. In *AAAI*, pages 10718–10726, 2021.

[145] Alireza Zareian, Kevin Dela Rosa, Derek Hao Hu, and Shih-Fu Chang. Open-vocabulary object detection using captions. In *CVPR*, pages 14393–14402, 2021.

[146] Rowan Zellers, Mark Yatskar, Sam Thomson, and Yejin Choi. Neural motifs: Scene graph parsing with global context. In *CVPR*, pages 5831–5840, 2018.

[147] Zhiyuan Zeng, Qinyuan Cheng, Zhangyue Yin, Yunhua Zhou, and Xipeng Qiu. Revisiting the test-time scaling of o1-like models: Do they truly possess test-time scaling capabilities? *arXiv preprint arXiv:2502.12215*, 2025.

[148] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris N Metaxas. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *ICCV*, pages 5907–5915, 2017.

[149] Ji Zhang, Kevin J. Shih, Ahmed Elgammal, Andrew Tao, and Bryan Catanzaro. Graphical contrastive losses for scene graph parsing. In *CVPR*, pages 11535–11543, 2019.

[150] Yong Zhang, Yingwei Pan, Ting Yao, Rui Huang, Tao Mei, and Chang Wen Chen. Learning to generate language-supervised and open-vocabulary scene graph using pre-trained visual-semantic space. In *CVPR*, pages 2915–2924, 2023.

[151] Chengyang Zhao, Yikang Shen, Zhenfang Chen, Mingyu Ding, and Chuang Gan. Textpsg: Panoptic scene graph generation from textual descriptions. In *ICCV*, pages 2839–2850, 2023.

[152] Yiwu Zhong, Jing Shi, Jianwei Yang, Chenliang Xu, and Yin Li. Learning to generate scene graph from natural language supervision. In *ICCV*, pages 1823–1834, 2021.

[153] Yiwu Zhong, Jianwei Yang, Pengchuan Zhang, Chunyuan Li, Noel Codella, Liunian Harold Li, Luowei Zhou, Xiyang Dai, Lu Yuan, Yin Li, and Jianfeng Gao. Regionclip: Region-based language-image pretraining. In *CVPR*, pages 16772–16782, 2022.

[154] Chaoyang Zhu and Long Chen. A survey on open-vocabulary detection and segmentation: Past, present, and future. *CoRR*, abs/2307.09220, 2023.

[155] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV*, pages 2223–2232, 2017.

[156] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable DETR: deformable transformers for end-to-end object detection. In *ICLR*, 2021.