Flinders University

Real-Time And Efficient Scene Graph Generation for Real-World Applications: An End-to-End Investigation

By

Maëlic NEAU
BCom, MCom

Thesis submitted to Flinders University and
Ecole Nationale d'Ingénieurs de Brest
under a cotutelle agreement
for the degree of

Doctor of Philosophy

College of Science and Engineering
27th of March 2025

Declaration

I certify that this thesis:

1. does not incorporate without acknowledgment any material previously submitted for a degree or diploma in any university
2. and the research within will not be submitted for any other future degree or diploma without the permission of Flinders University; and
3. to the best of my knowledge and belief, does not contain any material previously published or written by another person except where due reference is made in the text.

Maëlic NEAU

SUMMARY

Scene Graphs are powerful representations that abstract the content of images or videos in the form of relations triplets grounded to visual regions. Generating Scene Graphs through the task of Scene Graph Generation (SGG) seems especially promising for applications in Robotics such as in Human-Robot Collaboration (HRC) in domestic context where Scene Graphs can be used to model the environment and the interactions between the robot and the human. However, several years after the first inception of the task, the usage of Scene Graphs in real-world applications is still limited due to the poor performance of SGG models on out-of-distribution samples. In this thesis, we propose to bridge the gap between theoretical methods of SGG and their practical implementations in real-world settings, successfully contributing to the democratization of the usage of Scene Graphs. We first describe a new method for semi-automatic extraction of clean and qualitative annotations to create in-context Scene Graphs datasets from noisy data. This results in our first contribution, the IndoorVG dataset, a high-quality Scene Graphs dataset targeting scene understanding applications in a domestic context. When analyzing complex scenes, the number of relations triplets in SGG can grow quadratically, leading to a loss of performance for downstream tasks when the amount of non-informative relations predicted is high. To solve this issue, we propose a new inference process that selects a subset of highly informative relations from a set of biased and noisy predictions of an SGG model. This approach can substantially increase the performance of downstream tasks by improving the quality of generated relations. Our results on three different tasks (Visual Question Answering, Image Synthesis, and Image Captioning) demonstrate the importance of the informativeness of relations in Scene Graphs and the benefit of trading off accuracy for informativeness. To foster the usage of SGG in real-world applications and improve the deployment of models on embedded devices, we propose a new method for real-time SGG, based on state-of-the-art single-stage object detectors. Our method, named Real-Time SGG, is able to generate Scene Graphs in real-time on a single GPU without loss of accuracy, outperforming the current state-of-the-art methods in terms of speed and resource efficiency. We further extend the traditional static implementation of SGG to the time domain, introducing a Continuous SGG (C-SGG) architecture that aggregates relations from consecutive frames into a consistent representation.We applied our C-SGG method for real-time fine-grained activity understanding in the domestic context and demonstrated the advantage of our approach to model long-term complex activities in a Human-Robot Collaboration scenario.

ACKNOWLEDGEMENT

My first and most valuable thank goes indubitably to my first supervisor. I am grateful to Cédric Buche for his guidance, support, and encouragement throughout my PhD. More than a supervisor, Cédric has become a true friend and mentor throughout this journey. He is also the one that believed in me in the first place by giving me the opportunity of conducting this PhD, and I will be forever grateful for that. My second and no less valuable thank goes to my second supervisor Paulo E. Santos. Not only did Paulo shared is invaluable knowledge and expertise with me, but he also taught me the values and principles of scientific research. I am grateful for his wisdom, his guidance, and his friendship. If one day I become a successful researcher, Paulo you could be proud of yourself. My third thank goes to my third supervisor Anne-Gwenn Bosser. Anne-Gwenn was able to help me manage my fear and emotions throughout this journey. She was always there to listen to me, to support me, and to encourage me. I am grateful for her friendship, her kindness, and her patience. I also want to thank my fourth supervisor, Karl Sammut. Karl was of great support to help me manage my time at Flinders University. He is of the ones who greatly contributed to making me feel at home in Australia. I am grateful for his advices and his support.

I would also like to thank all the person that reviewed and evaluated my work. This includes Benoit Clément, Ricchard Lebrant, Philippe Rauffet and Mewhish Nassim. Through their fair and comprehensive feedback during the years, they helped me improve my work and my writing. These thanks also extends to all the kind anonymous reviewers of my publications.

A good PhD is nothing without good friends, here I would like to thank my friends from ENIB, namely Antoine, Amélie, Cédric L. B., Paul, Déborah, and all the others of the CERV family. At the same time, I would like to thank all my friends at Flinders University, including but not limited to, Thomas C., Thanh and Katell. My last home was without contest the CROSSING lab in Adelaide and all its members, Jean-Philippe, Thomas U., Thomas R., Raphaël, Quentin, Helene, Louis, Sinuo, Nilesh and all the others with whom I collaborated and shared good moments. I am grateful for their friendship, their support, and their kindness. A special thanks to Sarah for her unconditional support over the years.

Finally, I would like to thank my family, my dad, my mom and my sister for their unconditional love and support. I know they are proud of me, and I am proud of them.

List of Publications

Publications Related to This Thesis

Submitted, Under Review:

- **Maëlic Neau**, Paulo Santos, Anne-Gwenn Bosser, Cédric Buche. 2024. REACT: Real-time Efficiency and Accuracy Compromise for Tradeoffs in Scene Graph Generation. **Chapter 5**.

Published, Peer-Reviewed:

- **Maëlic Neau**, Paulo Santos, Anne-Gwenn Bosser, Alistair MacVicar, Cédric Buche. 2024. Mining Informativeness in Scene Graphs: Prioritizing Informative Relations in Scene Graph Generation for Enhanced Performance in Applications. *Pattern Recognition Letters*. **Chapter 4**.

- **Maëlic Neau**, Paulo Santos, Anne-Gwenn Bosser, Cédric Buche. 2023. Fine-Grained is Too Coarse: A Novel Data-Centric Approach for Efficient Scene Graph Generation. *Proceedings of the IEEE/CVF International Conference on Computer Vision (Workshops) (pp. 11-20)*. **Travel Grant Award (top 10%)**. **Chapter 3**.

- **Maëlic Neau**, Paulo Santos, Anne-Gwenn Bosser, Cédric Buche. 2023. In Defense of Scene Graph Generation for Human-Robot Open-Ended Interaction in Service Robotics. *RoboCup 2023: Robot World Cup XXVI (pp. 299-310). Springer Nature Switzerland.* **Chapter 3 and 6**.

Other Publications

Published, Peer-Reviewed:

- Cédric Buche, **Maëlic Neau**, Thomas Ung, Louis Li, Sinuo Wang, Cédric Le Bono. 2023. RoboCup@ Home SSPL Champion 2023: RoboBreizh, a Fully Embedded Approach. *RoboCup 2023: Robot World Cup XXVI (pp. 374-385). Springer International Publishing.*

- Sinuo Wang, **Maëlic Neau**, Cédric Buche. 2023. RoboNLU: Advancing Command Understanding with a Novel Lightweight BERT-based Approach for Service Robotics. *RoboCup*

2023: Robot World Cup XXVI (pp. 29-41). Springer Nature Switzerland. **Best Paper Candidate (top 5%)**.

- Natnael Wondimu, **Maëlic Neau**, Antoine Dizet, Ubbo Visser, Cédric Buche. 2023. Anthropomorphic Human-Robot Interaction Framework: Attention Based Approach. *RoboCup 2023: Robot World Cup XXVI (pp. 262-274). Springer Nature Switzerland.*

- Louis Li, **Maëlic Neau**, Thomas Ung, Cédric Buche. 2023. Crossing Real and Virtual: Pepper Robot as an Interactive Digital Twin. *RoboCup 2023: Robot World Cup XXVI (pp. 275-286). Springer Nature Switzerland.*

- Cédric Buche, **Maëlic Neau**, Thomas Ung, Louis Li, Tianjao Jiang, Mukesh Barange, Maël Bouabdelli. 2022. RoboBreizh, RoboCup@Home SSPL Champion 2022. *RoboCup 2022: Robot World Cup XXV (pp. 203-214). Springer International Publishing.*

- Thanh Tran, **Maëlic Neau**, Paulo E. Santos and David Powers. 2022. Contrastive Visual and Language Learning for Visual Relationship Detection. *The 20th Annual Workshop of the Australasian Language Technology Association (No. 20, pp. 170-177).*

- **Maëlic Neau**, Paulo E. Santos, Anne-Gwenn Bosser, Cédric Buche, Nathan Beu. 2022. Commonsense Reasoning for Identifying and Understanding the Implicit Need of Help and Synthesizing Assistive Actions. *2022 AAAI Symposium on Machine Learning and Knowledge Engineering - AAAI MAKE. CEUR Workshop Proceedings (Vol 3121)*

Prizes and Awards

- **HDR Leadership and Scholarly Excellence Award**: *Flinders University (2023).*
- **Travel Grant Award (top 10%)**: *SG2RL Workshop, IEEE/CVF International Conference on Computer Vision (2023).*
- **Best Paper Candidate (top 5%)**: *RoboCup 2023: Robot World Cup XXVI.*
- **RoboCup@Home SSPL Champion 2023**: *RoboCup 2023: Robot World Cup XXVI.*
- **RoboCup@Home SSPL Champion 2022**: *RoboCup 2022: Robot World Cup XXVI.*

Table of Contents

List of Figures

List of Tables

List of Algorithms

List of Acronyms

FN False Negative. 35

FP False Positive. 35

FPS Frame Per Second. 114

GSG Global Scene Graph. 137

HOI Human-Object Interaction. 51

HRC Human-Robot Collaboration. 25

HRI Human-Robot Interaction. 25

IoU Intersection over Union. 35

LLM Large Language Model. 67

mAP mean Average Precision. 35

NLI Natural Language Inference. 86

NLP Natural Language Processing. 85

NMS Non-Maximum Suppression. 35

PDDL Planning Domain Definition Language. 149

ROI Region of Interest. 116

RPN Region Proposal Network. 111

SG2IM Scene Graph to Image Generation. 42

SGG Scene Graph Generation. 25

STS Semantic Textual Similarity. 86

TDE Total Direct Effect. 37

TP True Positive. 35

TSG Textual Scene Graph. 83

U-SGG Unbiased Scene Graph Generation. 26

VQA Visual Question Answering. 25

VSG Visual Scene Graph. 83

Chapter 1

Introduction

Modeling the content of visual scenes has gained an important place in the field of computer vision in the last few years. From Image Captioning [1], [2], [3] to Visual Question-Answering (VQA) [4], [5], [6], the modeling of high-level representations of visual scenes has been a cornerstone for the development of AI systems that requires a deep understanding of the scene. One of the key components of these high-level representations is the notion of compositional relations [7], [8]. Compositional relations represent interactions between visual elements in a scene, such as objects or humans. These relations can be very diverse, from basic spatial descriptions such as "computer on top of desk" to partonomies such as "tail belongs to cat" or human actions such as "person eating pizza". Modeling such relations from images or videos can be enough for Image Captioning or VQA. On the other hand, other tasks could also benefit from the usage of compositional relations if these relations are grounded to the real word, such as in Embodied Agent Navigation [9], [10] or Human-Robot Interaction (HRI) [11].

The task of Scene Graph Generation SGG has been proposed [12] as an alternative to tackle these challenges. The task of SGG aims at creating a grounded representation of a scene by inferring relations between entities as a graph structure. Typically, approaches in SGG rely on detecting objects and their respective coordinates and then learning to model relations between these objects in the form of $\langle subject, predicate, object \rangle$ triplets [7], [13]. Connections between pairs of triplets form a directed acyclic graph in which each vertex refers to an object and its associated image region, and each edge a predicate expressed in natural language. Given its representation capabilities, this undertaking shows great potential as a foundational element for various subsequent tasks that hinge on compositional aspects such as Image Captioning [14] or Visual Question Answering [15]. SGG has also recently sparked interest in HRI [11] and Embodied Agent Navigation [9] tasks. However, we observed a significant unbalance between the progress in the SGG task itself and its adoption as a backbone in the aforementioned downstream tasks [16]. In this work, we aim at investigating the reasons behind this unbalance and propose new methods to bridge the gap between SGG approaches and their adoption in real-world applications. As a case study, we give insights on all aspects of the SGG task, from data collection to real-time implementation. To foster the development of the task of SGG in new domains and fields, we also propose a new usage of SGG models for Human-Robot Collaboration (HRC). In the next section, we detail the current challenges that SGG faces for

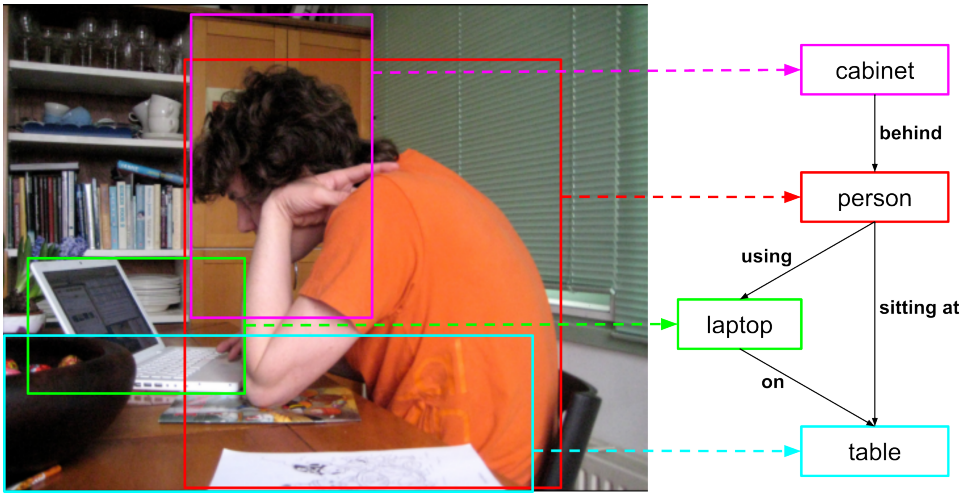


Figure 1.1: A Scene Graph (right) and corresponding image regions (left). Image taken from the Visual Genome dataset [8].

real-world applications and the opportunities to use the task for HRC use cases.

1.1 Scope and Motivations

The task of SGG exhibits a different learning paradigm from other vision-related tasks. In fact, compositional relations are not mutually exclusive and can be defined in a myriad of ways. For instance, the relation $\langle person, using, laptop \rangle$ in Figure 1.1 can also be defined as $\langle person, on, laptop \rangle$ or $\langle person, looking\ at, laptop \rangle$. When humans are asked to annotate images with compositional relations, they must choose a relation from all possible relations that can be defined between visual elements. This selection can heavily depend on the person's subjective interpretation of the scene, as well as the person's appreciation of English language. This process has led to a significant amount of noise in the annotations of leading datasets for the task [8], [15], [17], which can be detrimental to the learning process of neural networks [18], [19], [20]. One consequence of this problem has been identified by the community over the years [19]: the over-representation of certain predicate classes in the dataset, such as the predicate *on* which can be used by annotators in most situations, saving time and effort. Over-representation of certain predicate classes ultimately leads to strong unbalance and biases in the learning process, which has been overly addressed by the community through the development of the sub-task of Unbiased Scene Graph Generation U-SGG [19], [21], [22], [23].

A remaining issue with SGG datasets has not yet been addressed: the selection process of relations or more simply *which object pairs should we annotate in the first place?* Indeed, an image can contain a lot of objects and thus a quadratic amount of potential pairs. Annotating extensively all relations will then require a consequent amount of time and effort. In current

approaches [7], [8], [20], the types of relations which should or should not be annotated is left under the responsability of the annotators. This can lead to another bias in SGG datasets: the over-representation of obvious and irrelevant relations. Because some $\langle subject - object \rangle$ pairs appear frequently, they are more likely to be selected by the human annotators, leading to biased data. Ultimately, low data quality can lead to poor performance and poor generalization capabilities of the trained models, which can be detrimental to the deployment of AI systems in real-world settings [14], [18], [24]. Addressing the issue of relation selection in current leading datasets is the first motivation of our work.

Research Question 1: *Are current datasets for SGG biased and how can we improve the quality of the annotations to improve models' generalization?*

To successfully be used in downstream tasks, not only do generated scene graphs need to be correct, but they also have to contain as much information as possible. Even with a perfectly annotated dataset, there is no guarantee that an SGG model will predict *the most important relations* first. In fact, in traditional machine learning, only correctness is evaluated. However, in the context of SGG, a very high number of relations can be correct in a scene but only a few of them would play a significant role in the understanding of the scene. As an example, we display the predictions of a popular SGG model [19] in Figure 1.2. Even if the model has been trained to predict the relation $\langle person, using, laptop \rangle$, in this example, this relation was predicted with a confidence level too low to be kept in the final output because other more trivial relations (e.g. $\langle paper, on, table \rangle$, $\langle person, has, head \rangle$) were predicted with higher confidence. As a result, the predicted graph lacks an informative perspective (see Figure 1.1), even though all predicted relations are correct. This could be a challenge for downstream tasks which will intuitively benefit from a more concise and informative representation [16]. This leads to the second motivation of this work: the development of a method that can predict more relevant and informative compositional relations, with the goal of increasing the performance of SGG models in downstream tasks [14], [25].

Research Question 2: *Develop a method that can predict more relevant and informative compositional relations, which can be generic enough to benefit a wide range of downstream tasks.*

Another issue that can explain the slow adoption rate of SGG in subsequent applications is the lack of simple, low-cost, and effective approaches that could power applications that require low resources or real-time constraints (such as embodied agent reasoning [26]). While other tasks such as Object Detection or Image Captioning have seen the emergence of real-time and low-cost approaches [27], [28], the task of SGG did not. Nevertheless, generating comprehensive scene graph representations holds significant promise for edge computing use cases, for instance in HRI [11]. This leads us to the third motivation of this work: the development of an efficient implementation of SGG for real-time constraints.

Research Question 3: *How can we develop an efficient implementation of SGG for real-time*

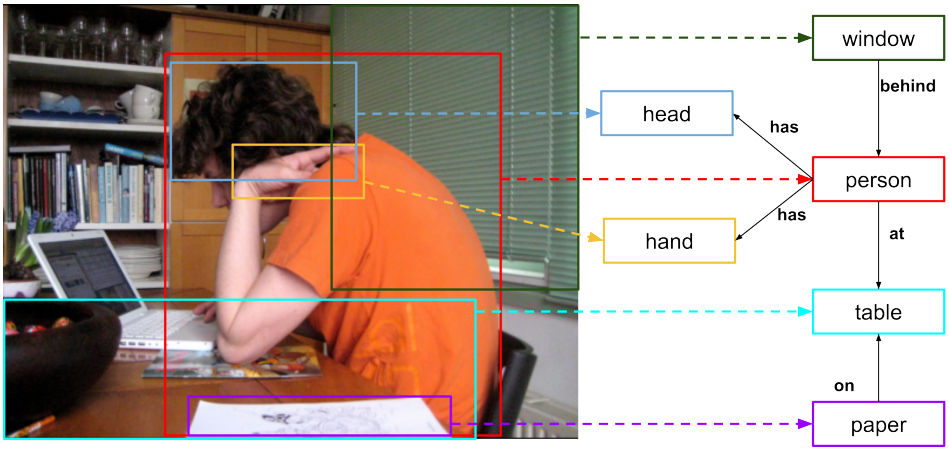


Figure 1.2: Predicted scene graph with the Transformer model [19] trained on the Visual Genome dataset [8].

constraints?

Finally, in a more general perspective, the task of SGG has been mostly studied in the context of visual understanding, but not so much in the context of Robotics and HRC [11], [29]. In the context of HRC, the robotic agent often needs to understand the actions being performed by the human and the context in which they take place. This process requires actively identifying agents, objects, and their interactions through time and space before proceeding with reasoning. Intuitively, SGG is a good candidate to power such representation. In practice nonetheless, challenges remain in the deployment of SGG models for HRC. First, comprehensive datasets for in-domain applications of HRC (e.g. in domestic settings) are lacking. Second, generating scene graphs for HRC requires extending the current architecture to the time domain, which may not be straightforward [30], [31]. These two challenges lead to the fourth motivation of this work: the development of a new method for the generation of a continuous SGG method that can be used as the internal World Model of an autonomous agent during HRC In particular, the task of HRC in domestic settings will be taken as a case study to illustrate the opportunities of compositional relations for these types of scenarios.

Research Question 4: *How can we develop a continuous SGG method that can be used as the internal World Model of an autonomous agent for SGG?*

In the following section, we detail the objectives of this work and our contributions to the field of SGG and HRC.

1.2 Proposed Approach

Our first motivation concerns the quality of the annotations in current datasets for the task of SGG. In this work, we propose to tackle the issue of data annotations by refining and augmenting

the existing large-scale dataset Visual Genome [8]. We aim at removing irrelevant annotations by categorizing compositional relations into semantic categories. Our approach can significantly improve the learning process of neural networks and improve the performance of baseline models in predicting compositional relations. As a case study, we proposed a new dataset, IndoorVG, that is tailored for the task of SGG for indoor scenes. This dataset has been carefully re-annotated by taking into account the before-mentioned issues and can be used as a benchmark for future research in the field. Here, we purposely chose the domain of indoor scenes as it is a common setting for SGG scenarios and can be used as a case study for our approach later on.

Once qualitative data has been gathered, training deep neural networks to predict compositional relations can start. To ensure that common SGG models predict informative relation first, we want to better align predictions with human description of scenes [32]. To solve this challenge, we draw inspiration from human perception of compositional relations through the notion of *image gist* [33]. The image gist (or gist of the scene) represent the essential information which is relevant in identifying a scene. This information is characterized in part by highly-informative compositional relations that are meaningful in the context of the scene [34], [35]. However, it is difficult to compute a proper value of the *informativeness* of a relation, mainly because this value is both *intrinsic* and *extrinsic*. We define *intrinsic informativeness* as the semantic value of a relation taken in isolation. For instance the relation $\langle person, eating, pizza \rangle$ is more informative than $\langle person, wearing, hat \rangle$ as it provides more meaning to the scene, i.e. the first relation is more likely to appear in the image gist than the second one, in images where both relations hold. On the other hand, we define *extrinsic informativeness* as the relevance of a relation to the overall context of the scene. Here, we define this context as the combination of all other relations that hold in the given scene. For instance, the relation $\langle person, has, hand \rangle$ is more informative in a scene where the person is holding an object than in a scene where the person is not. The combination of intrinsic and extrinsic informativeness are used to define a proper value of informativeness for a relation, which can be used to select only relevant relations in the context of the scene. By applying this method on SGG models, we significantly improve the performance of downstream tasks which rely on compositional relations as input [14], [25], [36].

Once an SGG model has been trained, it can be used to predict compositional relations in real-time. Real-time and low-resources consumption are both requirements for embodied agent applications. In the context of SGG, this requirement is not fulfilled by current methods, which are often slow and memory-consuming [26]. At the same time, current SGG methods give poor performance in object detection due to the jointly training of object detection and relation prediction tasks [37], [38]. To solve these issues, we propose to review the current architecture of SGG and adapt it to state-of-the-art strategies employed for real-time object detection [39]. All together, our proposed method improves the latency of an SGG model up to a factor of 10 compared to traditional methods without loss of performance.

Given a real-time and informative SGG model, we can use its predictions as observation of the environment to power the *World Model* of an autonomous agent for SGG. For this last part, we will take as a case study the collaboration of a human and a robotic agent in domestic daily life activities. These scenarios can encompass collaboration in common activities such as cooking, cleaning or setting up dinner. In this context, the robotic agent needs to understand by itself, given its representation of the world, the actions being performed by the human and the context in which they take place. We proposed a new Continuous SGG method that aggregates relations over time and space to generate a consistent internal representation of the visual world. The grounding in space is done through the visual coordinates of the objects and humans in the scene, and the grounding in time is done through the temporal sequence of the relations. We specifically introduce a new method for the aggregation of relations which ensure that important relations are kept in memory and less important ones are forgotten. Relations can also be refined if their aggregation is breaking basic commonsense rules. For instance, if the relation $\langle person, sitting\ on, chair \rangle$ is aggregated with the relation $\langle person, sitting\ on, couch \rangle$, the model should delete the first one as it is not possible for a person to sit on two different furniture at the same time. Finally, to demonstrate the interest of such an architecture, we paired this representation with traditional planning systems [40]. We proposed a new method for automatic planning domain generation based on the interplay of compositional relations categories. As a result, we showed that our method can be used in the context of Symbolic Demonstration Learning, and can spark new research for SGG based on Scene Graphs.

1.3 Contributions

We can summarize the main contributions of this work as follows:

1. First, a new method of annotations selections and refining for the task of SGG has been proposed. By selecting only relevant annotations, we can improve the performance of baseline models in predicting compositional relations.

2. We introduced a new dataset, IndoorVG, that is tailored for the task of SGG for indoor scenes. Images and annotations from the IndoorVG dataset were extracted from the Visual Genome dataset. Subsequently, new refinement and augmentation methods were proposed to enhance both the quality and quantity of the data.

3. A new method for the prediction of compositional relations in visual scenes based on *intrinsic* and *extrinsic* informativeness is introduced. We showed that this method can greatly improve the performance of downstream tasks which rely on compositional relations as input.

4. Using state-of-the-art real-time object detectors, we proposed a new real-time SGG method. This method can improve the latency of SGG models up to a factor of 10 compared to traditional methods, without loss of performance.

5. We presented a new Continuous SGG method that autonomously generate a consistent internal representation of the visual world through compositional relations. This representation can be paired to traditional planing systems through a proposed planning domain generation algorithm. We demonstrated the effectiveness of this representation in the context of Human Activity Recognition and Learning from Observations.

1.4 Thesis Outline

In the following chapters, we detail the proposed approach and the contributions of this work. We start by reviewing the background concepts and related works to the notion of image gist and SGG in Chapter 2. Then, in Chapter 3 we introduced a new definition of relevant compositional relations and proposed a new method for automatically mining those relations in large-scale noisy datasets. In Chapter 4, we proposed a new method for predicting Scene Graphs based on the semantic importance and relevance of relations to the image gist. By leveraging real-time object detector, we presented a new method for real-time SGG in Chapter 5. Finally, in Chapter 6, we defined the concept of Continuous SGG and a new architecture that can be used as the internal World Model of an autonomous agent in SGG use cases. We conclude this thesis in Chapter 7 by summarizing the contributions and discussing future works.

Chapter 2

Background

In this chapter, we provide an overview of the main concepts and techniques related to the predictions of compositional relations from visual scenes. Compositional relations between objects in a scene are a key aspect of human perception and understanding of the world. Following previous work [7], we define a compositional relation as a relation that is described by a combination of two distinct visual elements in the scene, where one element s will serve as the *subject* and the other o as the *object*, related by a *predicate* p as follows:

$$r = \{(s, p, o) | p \in R, s \in V, o \in V, s \neq o\} \tag{2.1}$$

where V is the set of visual elements and R is the set of possible relations. In the following, we will often call the relation (s, p, o) a *triplet*, as per the Resource Description Framework (RDF) standard [41]. In the field of Computer Vision, the task of SGG aims at modeling these relations in a structured way. In this chapter, we will introduce the concept of Scene Graphs and their importance in the field of Computer Vision. We will also present the most popular approaches for the task and their main differences, see Section 2.1. The concept of Scene Graph is tight to the idea of *image gist* [32], a term coined by Aude Oliva in 2005 [33] to describe the structural summary of an image in human perception. To allow SGG methods to be more efficient in real-world applications, such methods need to focus on extracting relevant relations that are part of the image gist. We will discuss the concept of image gist and its relation to Scene Graphs in Section 2.2. Finally, we will outline the current trends and challenges of SGG for downstream tasks, and their implementations in real-world applications, see Section 2.3.

2.1 SGG

A symbolic representation system is a system that represents knowledge in a discrete form, such as a set of symbols and rules that is used to infer new knowledge. In the field of Human-AI collaboration, such representations are often used to represent the environment in which the AI agent is evolving, what we call a *world model*. This world model can take different forms, such as a semantic map [42] or an ontology [43] depending on the application. However, when representing a wide range of information types (such as spatial relations, object properties, etc.), one of the

most flexible representations is a graph structure. Graphs that represent visual information of a visual scene are called *Scene Graphs*. Scene Graphs can be defined as "a data structure that describes the object instances in a scene and the relationships between these objects" [16]. These compositional relations are represented as RDF [41] $\langle subject, predicate, object \rangle$ triplets. Each of the subject and object are usually tied to a 2D region of the image in the form of bounding box coordinates, thus we have:

- $B = \{b_1, ..., b_n\}$ the set of bounding boxes coordinates
- $O = \{o_1, ..., o_n\}$ the set of objects, with o_i denoting the class label of the object in the bounding box b_i
- $R = \{r_{1\to2}, ..., r_{n\to n-1}\}$ the set of relations, with $r_{i\to j}$ denoting the class label of the relation between the objects o_i and o_j.

For a scene graph representation G of an image I we can decompose the probability distribution as follows:

$$p(G|I) = p(B|I) \cdot p(O|B, I) \cdot p(R|O, B, I) \tag{2.2}$$

The objective of *SGG* is to infer the most likely graph G^* for a given image, as follows:

$$G^* = \arg\max_G p(G|I) \tag{2.3}$$

Given this definition, the SGG community divided the task into two main components: Object Detection and Relation Prediction. Object Detection is the task of detecting the objects' location B in the image and classifying them into a set of predefined classes O. Relation Prediction is the task of predicting the relations R between the detected objects. The output of the Relation Prediction component is a set of relation pairs and predicate labels. These two tasks are usually solved separately with different neural network models and then combined to generate the final scene graph. In that sense, SGG is similar to other multi-task problems that use Object Detection as a backbone such as Visual Question Answering (VQA) [5] or Visual Dialog [44]. In the following, we describe in more detail the traditional approach for Object Detection and Relation Prediction in SGG.

2.1.1 Object Detection

An object detection model usually takes as input an image and outputs a set of bounding boxes and their corresponding class labels. The bounding boxes are represented as a set of four coordinates (x, y, w, h), where (x, y) is the center of the box and (w, h) are the width and height of the box, in the pixel space. The class labels O are represented as a set of N probabilities, where N is the number of classes in the dataset. Each probability represents the confidence of

the model that the detected object belongs to the corresponding class. The predicted class label is selected by taking the class probability with the highest confidence score for each bounding box. The final set of bounding boxes is then usually refined using Non-Maximum Suppression (NMS). NMS is a post-processing step that removes redundant bounding boxes by selecting the one with the highest confidence score and removing all the other bounding boxes that have a high Intersection over Union (IoU) with the selected one. The IoU is a metric that measures the overlap between two bounding boxes. The IoU is defined as the ratio between the intersection area and the union area of the two bounding boxes, as follows:

$$IoU = \frac{Area(BB_1 \cap BB_2)}{Area(BB_1 \cup BB_2)} \tag{2.4}$$

After NMS, overlapping boxes for the same objects are usually discarded, and the final set of bounding boxes and corresponding labels is outputted.

Object detection models are evaluated on standard benchmarks such as the COCO dataset [45] using different metrics. Standard metrics for the task of Object Detection are Precision and Recall, which are defined as follows:

$$Precision = \frac{TP}{TP + FP}, \tag{2.5}$$

$$Recall = \frac{TP}{TP + FN}, \tag{2.6}$$

with TP being the number of True Positives, FP the number of False Positives and FN the number of False Negatives. Positives and Negatives are defined by the IoU between the predicted bounding boxes and the ground truth bounding boxes which share similar class labels, after the step of NMS. A predicted bounding box is considered a TP if its IoU with the ground truth bounding box is above a certain threshold. A predicted bounding box is considered a FP if its IoU with the ground truth bounding box is below the threshold, usually, this threshold is set at 0.5. To get a single value for all classes, all the results are averaged in the mean Average Precision (mAP) metric. The mAP is defined as follows:

$$mAP = \frac{1}{N} \sum_{i=1}^{N} AP_i, \tag{2.7}$$

with N being the number of classes and AP_i the Average Precision for class i. In recent works, the mAP metric is defined on a scale of IoU thresholds, ranging from 0.5 to 0.95 with a step of 0.05. This ensures that the model can predict fine-grained bounding box coordinates that are close to the ground truth bounding boxes. This metric is called mAP@50-95, and is the most widely used metric for evaluating Object Detection models. The mAP@50 metric does not take

into account the IoU threshold and is the average of the mAP for all IoU thresholds above 0.5. In this work, we will use both metrics to benchmark Object Detection models.

The bounding boxes and class labels predicted by an Object Detection model can be used as a backbone for subsequent modules in scene understanding. In the case of SGG, both the bounding box coordinates and object class labels will be used by the Relation Prediction component to predict compositional relations between objects.

2.1.2 Relation Prediction

The Relation Prediction model is a component that cannot be trained alone, in contrast with the object detection model as it needs bounding boxes and object class labels as input. In the relation prediction stage, the visual features, bounding box coordinates, and object class labels are used to create a graph representation of the scene. The visual features are extracted from the backbone of the object detection model and are used to initialize the node features of the graph. In addition to visual object features extracted from the backbone, the relation prediction model also uses textual features to model the representation of subject and object labels. This is usually done by using a pre-trained word embedding model such as GloVe [46]. The word embedding model is used to extract a vector representation of the subject and object class labels. This vector representation is then multiplied with the visual features of the subject and object proposals to form the final node features.

Edges are created between all possible pairs of nodes in the graph. Edge features are initialized with the union of the two nodes features. Then, different learning paradigms take place to refine node and edge features to learn the interdependencies between relations. We review the most popular ones in the following:

- Iterative Message Passing (IMP) [12]: Iterative Message Passing was proposed in 2017 by Xu et al. as the first end-to-end trainable model for SGG. Message Passing is a technic of Graph Neural Networks (GNN) that allows nodes to communicate with each other by exchanging messages along the edges during learning. IMP uses an RNN-based architecture to propagate information between nodes and edges in an iterative manner to form the final graph representation. In contrast to previous approaches to model compositional relations [7], IMP explicitly allows the model to learn the interdependencies between compositional relations.

- Neural-Motifs [38]: In 2018, Zellers et al. proposed Neural-Motifs, a new approach to SGG based on the concept of motifs. Motifs are recurrent sub-graph structures identified in SGG datasets that can be exploited to better memorize interdependencies of compositional relations in sub-regions of the image. To leverage such structures, Neural-Motifs proposed

a local-to-global context learning paradigm, with the local context referring to the inside-motif information and the global context the interplay of motifs at the image level.

- VCTree [37]: A year later, in 2019, Tang et al. proposed the VCTree model for the task of SGG. In contrast to IMP or Neural-Motifs which use an all-to-all matching of nodes during learning, VCTree introduces a dynamic tree-based structure to model the interdependencies between nodes. In VCTree, hierarchical information between nodes is learned through a Tree-LSTM network. For instance, the model will learn that the parent node of *car*, *person* or *bus* is *street* as a lot of relation pairs are annotated with the class *street* as *subject* and *car*, *person* or *bus* as *object* in the dataset. Parallel relations are also modeled in VCTree, as the model can learn that *car* and *bus* are both related to *street* in the same way. To form the final graph, nodes are paired through their parent or neighboring nodes in the tree structure.

- GPS-Net [47]: The Graph Property Sensing Network (GPS-Net) was proposed in 2020 by Lin et al. This approach builds on previous work by proposing to explicitly model the direction of compositional relations in the graph structure. In fact, a significant amount of compositional relations can be modeled in two directions, for instance, the relation $\langle person, wearing, shirt \rangle$ can also be modeled as $\langle shirt, on, person \rangle$. For better learning, it is important to model both directions of the relation. Inspired by IMP and Neural-Motifs, GPS-Net proposed a new Direction-Aware Message Passing mechanism for SGG.

- SGG-Transformer & Total Direct Effect (TDE) [19]: The SGG-Transformer model and the TDE debiasing method for SGG have been proposed by Tang et al. in 2020. The SGG-Transformer successfully leverages the Transformer [48] neural architecture to learn contextual information of compositional relations through the attention mechanism. In the same work, the authors proposed a new method based on TDE for debiasing other models' predictions. It is known that SGG models cannot achieve reasonable performance on the task because of the long-tail distribution of predicate classes in the dataset. Statistically significant predicates such as *on* will be overly predicted in comparison to fine-grained but less frequent predicates such as *sitting on*. The TDE method aims at solving this problem using Counterfactual intuition with the given contextual information for a selected $\langle subject, object \rangle$ pair. By analyzing the differences between predictions of an SGG model with and without contextual information from other visual regions, the TDE method can generate more reasonable relations. This method sparked a new shift in the SGG community and created the new task of U-SGG. U-SGG methods solely aim at solving the long-tail problem of SGG datasets and are thus better related to the subfield of long-tail learning than to SGG itself. As a result, we will not cover U-SGG methods in this work.

- Prototype Embedding Network (PE-NET) [49]: Recently, the PE-NET model was proposed

by Zheng et al. PE-NET is based on the idea of Prototype Learning, a method that aims at learning semantic prototypes as class representations. Prototypes are different from traditional visual representations as they are distinct from the pixel-space and learned from a separate semantic space, to reduce the dependence on pixel noise. In the case of SGG, prototype learning is used to generate representations of nodes and edges that are very distinct from each other, reducing the confusion of the model for the prediction of fine-grained compositional relations. At the time of release, the PE-NET model achieved state-of-the-art performance on the Visual Genome dataset [8] [1].

After context learning, the final graph is predicted by applying a softmax on the relation logits of every possible $\langle subject, object \rangle$ pair. Most approaches use cross-entropy loss to train their models, with the objective of maximizing the likelihood of the correct ground truth relation. It is important to notice here that cross-entropy loss is applied at the relation level and not at the graph level, as ground truth graphs can have varying sizes (in contrast, predicted graphs will always have a size of $n \times (n-1)$ with n being the number of detected objects).

The standard way of assessing SGG is to compute the average of the **Recall@K** (R@K) metric for every graph, as follows:

$$Recall@K = \frac{TP@K}{TP@K + FN@K}, \tag{2.8}$$

with $TP@K$ the number of TP computed in the top K predictions. This measures how often the correct relations are predicted in the top K confident predictions, traditional approaches use R@20, R@50, and R@100. The use of Recall instead of Accuracy is motivated by the observation that not all TP samples are annotated in SGG datasets [7], as in fact up to $n * (n-1)$ relations can be labeled in an image, it would be extremely intensive to ask human annotators to annotate all of them. Another problem of SGG datasets is the long-tail distribution of predicate classes which makes the performance of SGG models unstable for infrequent classes. To address this, a metric called **Mean Recall@K** (mR@K) was introduced [37], as follows:

$$meanRecall@K = \frac{1}{m} \sum_{i=0}^{m} \frac{TP_i@K}{TP_i@K + FN_i@K}, \tag{2.9}$$

with m being the number of predicate classes and $TP_i@K$ the number of TP for class i computed in the top K predictions. The Recall@K metric is the average recall for all classes, while the mR@K metric is the average recall for each class. The meanRecall@K metric is also computed for values of $K = [20, 50, 100]$ and averaged across every image to get a single value. This metric calculates recall for each predicate category independently and then takes the average

[1] As of 22/06/2023. This performance is compared only with other standalone methods (IMP, Neural-Motifs, GPS-Net etc..), this can be subject to change if adding any debiasing method to the PE-NET model, such as the TDE method.

of the results, giving equal weight to each category. By doing so, it reduces the impact of some frequently occurring predicates such as "on" and "of," while placing more emphasis on infrequent predicates like "reading" and "carrying". Instinctively, predicting more fine-grained predicates is beneficial for the high-level reasoning which depends on compositional relations. For instance, in the case of Human-AI collaboration, predicting that a person "is eating" a pizza is more beneficial than predicting that a person "has" a pizza. The former can be used to infer that the person is hungry and that the pizza is likely to be consumed, while the latter does not provide any useful information to infer new relations. However, predicting fine-grained relations is not sufficient to attain human-level performance in SGG. In fact, we, humans, also select the most relevant relations to describe a scene, and we do not predict relations that are not part of the *image gist*. In the next sections, we will discuss the concept of image gist and how it can be used to better align SGG models with human perception.

2.2 The Image Gist: on the Perception of Scene Semantics

In 2005, Aude Oliva [33] described the concept of an image *gist* as a "structural summary that's meaningful enough for recognizing the image". We could subdivide information from the image gist into two categories: perceptual information and conceptual information. The perceptual "part" of the gist is composed of shapes and textures and basic elements gathered during perception [33]. The conceptual part of the gist is composed of higher-level information such as the scene category, the spatial layout of objects, and the global semantics of the scene [33]. The conceptual aspect is particularly relevant to our work on SGG. In the following sections, we will refer to the conceptual part of the image gist as the *image gist*.

Subsequent studies have aimed to pin down what kind of information should make up this summary. For Mandler and Parker [50], image gist is centered around the appearance of objects and their spatial relations, on the other hand Biederman et al. [51] have proposed that image gist also encompasses global semantics and contextual information. In a long study, Li et al. [34] detailed that low-level sensory information precedes the extraction of high-level semantic information and that propositional relationships between objects make up for the majority of the image gist. We display an example of the study conducted by Li et al. in Figure 2.1, showing some description of the image gist after participants were shown the corresponding images for 500ms. In these descriptions, we can observe spatial information (i.e. "in the foreground", left image) but also actions (e.g. "sitting on", "standing in") or positional relations (e.g. "something in his hands"). Relations contained in the *image gist* can be aggregated in network representations [52], which will resemble closely to the definition of a scene graph.

So, can scene graphs generated by SGG models be used as a representation of image gist? In practice, it is more complex to draw such conclusions. Image gist traditionally encompasses

It was definately on a coast byt hte ocean with a large [r]ock in the forground and atleast three bird sitting on the rock.

This looks like a father or somebod helping a little boy. The man had something in his hands, like a LCD screen or laptop. they looked like they were standing in a cubicle.

Figure 2.1: Examples of image gist as a free-form description of scenes, when the scene has been shown to the participant for 500ms [34].

visually important and relevant information that is extracted within a very short amount of time by the human perception system (typically between 100ms to 300ms [53]). This information never refers to specific details and rather conveys the overall *meaning* of the scene and where it takes place [34]. By not incorporating any notion of relevance, scene graphs are sometimes too specific in the description of scenes and miss out on the image gist. In contrast to the concept of global context in SGG [38], the image gist does not necessarily refer to global relations between sub-regions of the image. It can be the case (see the relation between foreground/background in Section 2.2) or not (see Section 2.2). To align the representation of SGG models and human perception, one needs to take into account the relevance of relations to the image gist, i.e. the amount of information contained in the predicted graph that a human would instinctively perceive and elaborate when asked to shortly describe the image content [34]. Focusing on the relevance of relations in scene graphs is different from the current paradigm of SGG that aims at predicting correct relations, independently of their intrinsic meaning. We hypothesize that this new paradigm will lead to more efficient SGG models, and improve the performance of subsequent applications of SGG. In the next section, we will discuss why this is the case for several downstream tasks of SGG.

2.3 Applications of SGG

The task of SGG has recently gained interest as a backbone in a variety of other tasks. In a recent survey, Chang et al. [16] have identified the main applications of SGG in the literature, including

Image Retrieval, Visual Question Answering, Image Captioning, and Image Generation. Even if not directly addressed, the opportunities of SGG for robotics are also pointed out in this survey with applications in robot navigation. In an attempt to cover a broader range of applications, we will discuss the aforementioned applications as well as the usage of SGG in robotics in this section.

2.3.1 Image Retrieval

In the first inception of the task [54], Scene Graphs were used to improve the performance of image retrieval systems. The idea was to use the graph structure of the scene to retrieve images that are semantically similar to a query graph. Using Conditional Random Field (CRF) Johnson et al. [54] proposed a method that aims at grounding a given query graph to a set of images. Later on, SGG was proposed for the task of image-to-image retrieval, where the goal is to retrieve images that are visually similar to a query image [55]. In this approach, Scene Graphs are generated using the Bottom-Up Attention method [6] for all images and then embedded in a graph embedding space. By comparing the features of the query graph with the features of the scene graphs, the model retrieves images that are visually similar to the query image. In this work, the authors point out the difficulty of generating comprehensive graphs from images. The first discovery is that the performance of SGG methods on the standard benchmarks is not consistent with their respective performance in the image retrieval task. The authors have tried more qualitative SGG methods [12], [56], but the best performance reported is with the Bottom-Up Attention method [6]. This highlights the misalignment of current SGG benchmarks with the actual use of SGG methods in Image Retrieval.

2.3.2 Visual Question Answering

Visual Question Answering (VQA) is another downstream task that can benefit from SGG. In VQA, the goal is to answer complex questions about visual scenes. With the GQA dataset [15], Hudson et al. proposed a new approach for VQA with the support of scene graphs as an additional input to question-answer pairs. Next, the GQA dataset has been used to abstract the image content and facilitate answer prediction in VQA [25], [57]. Nevertheless, the quality of generated scene graphs for VQA has been pointed out by several studies [4], [57], [58]. In their work, Damodaran et al. [58] analyzed in depth the performance of the Motifs model [38] for SGG in the VQA task. Results show that predicted scene graphs correctly identify the objects but introduce a lot of noise by predicting many relations not related to the question-answer pairs. This noise can be detrimental to the performance of VQA models, as it introduces irrelevant information that confuses subsequent learning.

2.3.3 Image Captioning

The task of Image Captioning [59] aims at generating free-form textual descriptions of images. It can be convenient for Image Captioning to use scene graphs as abstract representations of the image content before proceeding to the generation of the caption [60]. As a result, the task of SGG was proposed as a backbone for Image Captioning [2], [6], [61]. In a so-called paper "Are scene graphs good enough to improve Image Captioning?", Milewski et al. [62] analyzed the quality of scene graphs produced by SGG models for the task of Image Captioning. Findings show a clear imbalance in scene graph quality generated by the IMP model for SGG [12]. Furthermore, the authors identify the quality of the scene graphs as a key factor for the performance of Image Captioning models. The noise of the scene graphs generated by SGG models has also been pointed out by subsequent work [14], [63].

2.3.4 Image Generation

Recently, the task of Image Generation has gained new interest with the democratization of Latent Diffusion models [64]. In Image Generation, the goal is to generate realistic images from a given textual description. Approaches such as Generative Adversarial Networks (GAN) or even Diffusion models have been criticized for their lack of compositionality [36], [65]. In this context, Scene Graphs have been proposed as an intermediate representation to guide the generation of images, leading to the task of Scene Graph to Image Generation (SG2IM) [36], [65]. In contrast to Image Generation from free-form text, SG2IM aims at generating images from structured representations of scenes. However, generating scene graphs from images and then using these graphs to re-generate images does not make a lot of sense and thus SGG models have not been used for Image Generation. Instead, ground truth scene graphs are used as inputs [65]. In recent works nonetheless, approaches in SGG are using the task of Image Generation from Scene Graphs as a benchmark to evaluate the quality of the generated scene graphs in SGG [35], [66]. This approach is particularly interesting as the SG2IM task is the only downstream task of SGG that requires the sole scene graph as input. In fact, in Image Captioning the image is also used as input, in VQA the question is used as input, and in Image Retrieval the query graph is used as input. This makes the SG2IM task a good benchmark to fairly evaluate the quality of scene graphs generated by SGG models, without other inputs as confounding factors [19]. In their work, Wang et al. [35] have shown qualitatively that the quality of scene graphs is a key factor for the performance of Image Generation models.

2.3.5 Robotics

Recently, we have witnessed the emergence of SGG for embodied agents, especially in autonomous exploration and navigation. Graphs representing spatial relations between static ob-

jects are leveraged from 2D image features [67] or from 3D sensors using PointCloud [68] during the navigation of an embodied agent. This representation is then used for autonomous exploration of the environment [69] or for complex symbolic planning [70]. In contrast to the aforementioned work, a recent approach [11] used SGG for high-level reasoning in HRI. In their approach, Amodeo et al. proposed to use an SGG model in a real-world application for telepresence robotics. For such a scenario, abstracting the scene content in a graph representation is beneficial for the person controlling the robot in a distance as it allows to define control commands in natural language (such as *"move toward the person holding a glass behind the table"*). This work sparks interest in more open-ended applications of SGG in robotics, especially in the field of HRI.

2.4 Redefining SGG for Downstream Tasks

SGG models have been heavily criticized not for predicting wrong relations but for introducing a large amount of noise in the predictions. Nonetheless, precisely identifying and quantifying this noise has not been done to this day. For the task of VQA, the noise seems to be any relations not related to the question-answer pair, but questions and answers can be very diverse, so it is hard to systematically eliminate the noise. For Image Captioning, defining noise is even more complicated due to the varying size and diversity of captions. In this work, we hypothesize that noise in SGG models can be defined as any relations not part of the *image gist* and that predicting relations relevant to the image gist will lead to better performance in the majority of downstream tasks. In the work of Amodeo et al. for telepresence robotics, the noise introduced by the SGG model is controlled by the addition of a handcrafted ontology. By explicitly defining the relations that matter for their particular use case, the authors were able to reduce the noise in the predictions of the SGG model. However, this method is not scalable to more advanced applications and can not generalize to other downstream tasks.

In contrast to traditional knowledge representations in robotics [43], [71], [72], SGG representations are static and do not allow for dynamic reasoning. In fact, the scene graph representation is computed from a single image and does not take into account the temporal evolution of the scene. When applied to navigation or manipulation tasks [69], [70], all relations are aggregated through time with no distinction. In addition, relations are never discarded which can lead to consequent overhead in long term tasks. Other approaches [11] simply do not aggregate relations through time and base their reasoning on a single image. This can lead to a lack of context in the reasoning process. Finally, the SGG approaches in robotics are not mentioning inference time and resources constraint of their implementations. In the context of robotics, the inference time of models is a key factor for the deployment on embedded systems. In addition to temporal aggregation of relations, a realistic low-cost implementation of SGG models for robotics is lacking.

We review the aforementioned challenges in a set of four distinct bottlenecks for the deployment

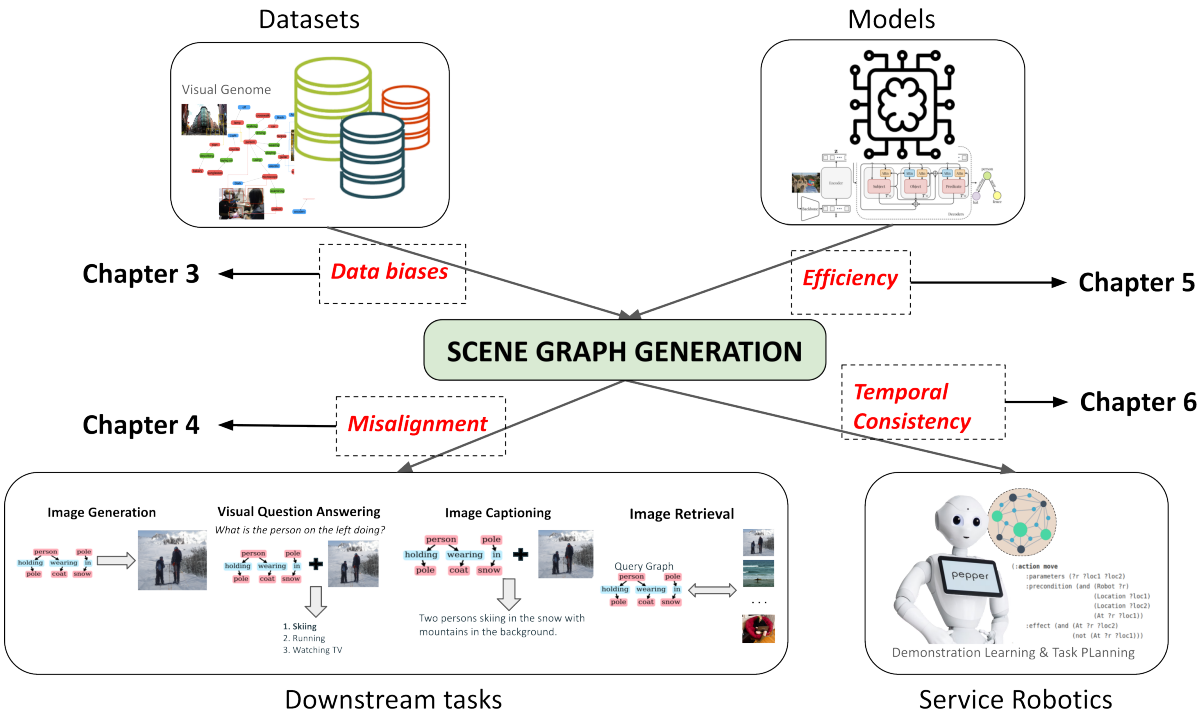


Figure 2.2: Overview of the different bottlenecks in SGG for real-world applications that will be discussed in this thesis.

of SGG models in real-world applications, and especially in service robotics:

1. Data biases: Current datasets SGG training are not representative of the real-world and introduce several biases that can be detrimental to the performance of SGG models.

2. Misalignment: Because of the noise introduced by SGG models, predictions are often not aligned with the constraints of downstream tasks, leading to a decrease in performance.

3. Efficiency: SGG models are not efficient for real-time applications and require a lot of resources to be deployed in real-world settings.

4. Temporal consistency: SGG models do not take into account the temporal evolution of the scene and do not allow for dynamic reasoning. To be successfully used in service robotics, SGG representations need to empower some sort of *consistency* when aggregating relations through time.

These different bottlenecks are represented in fig. 2.2. This thesis does not aim at solving each of these challenges but rather to propose new paradigms and methods to address them. Each of

the following chapters will focus on one specific bottleneck, detailing its scope, the current state of the art, and the proposed solution. In the next chapter (see Chapter 3), we will focus on the first bottleneck, Data biases, and propose a new method to refine data annotations in current SGG datasets. In Chapter 4, we will focus on the second bottleneck, Misalignment, and propose a new method to remove the noise of the predictions of SGG models. In Chapter 5, we will focus on the third bottleneck, Efficiency, and propose a new method to reduce the computational cost of SGG models. Finally, in Chapter 6, we will focus on the fourth bottleneck, Temporal consistency, and propose a new method to aggregate relations through time in SGG models.

Chapter 3

Modelling Compositional Relations: A Data-Centric Approach

Garbage in, garbage out.

George Fuechsel, 1957

> *Part of this chapter was published in the proceedings of the 2023 International Conference of Computer Vision (ICCV) as part of the SG2RL Workshop [73] and in the proceedings of the 2023 Robot World Cup Symposium [74].*

To be successfully used in real-world applications, it is crucial that predictions from SGG models represent the diversity and complexity of the real world. Like many other tasks, the paradigm of SGG is based on supervised learning. Supervised learning requires a large amount of high-quality labelled data, which can be difficult to acquire in the context of SGG. Namely, the high number of possible relations per image and the polysemic nature of natural language are two of the main challenges in the annotation process [75]. As a result, datasets that have been proposed for the task [8], [15] have been heavily criticized for having noisy and biased annotations [18], [19], [75]. Furthermore, these datasets are context-agnostic, which does not help with the generalization of models to specific contexts, for instance in domestic applications. Real-world applications, and especially HRI in domestic domain, require comprehensive and high-quality data of specific domains.

In this chapter, we focus on producing high-quality data for SGG in domestic applications, using this as a case study to propose two novel methods for data refinement and augmentation. These methods are centered on refining existing datasets by removing irrelevant annotations as well as selecting interesting classes which better represent real-world diversity. First, we will re-

view current datasets in SGG, discussing their biases and the data-centric approaches developed to address these issues in Section 3.1. We will then present our data-centric method to refine noisy annotations by tackling first the problem of irrelevant annotations, see Section 3.2, and second the problem of connectivity in SGG datasets, see Section 3.3. Finally, we will demonstrate the effectiveness of those approaches as well as a new data augmentation method for the semi-automatic annotation of a dataset in the context of domestic applications in Section 3.4.

3.1 From Noisy to Meaningful Graphs: A Literature Review

In the past few years, various approaches have been proposed for the task of SGG. However, the learning process in SGG is slightly different from other tasks in Computer Vision. In fact, predicting relations cannot be simply defined as a classification task because relations are not necessarily exclusive, defining the task as a multi-classification problem where each $\langle subject, object \rangle$ pair could be associated with up to n different relations. From a data perspective, this would require extensively annotating multiple positive and negative relations for each pair of objects which considerably enhances the resources required to produce large-scale datasets. Due to this constraint, current approaches have decided to simplify the task to a traditional classification problem where each pair is associated with one unique predicate label, described as the *graph-constrained* [38] settings. But this poses another problem: now it is the responsibility of the annotator to choose between a set of multiple valid labels to annotate a relation, which can lead to confusing and noisy annotations [76]. In the following, we will analyze the commonly used datasets for the task of SGG, their respective biases and approaches that have been proposed to solve them.

3.1.1 SGG Datasets

The first dataset which contains densely annotated scene graphs that have been proposed is the VRD dataset [77]. VRD is a dataset of 5,000 images annotated with 100 object categories and 70 predicates. The object and predicate classes have been manually selected to represent a wide range of possible interactions in different contexts. The dataset has been manually annotated to result in a total of 37,993 annotated relations and 6,672 possible triplets. The main criticism of this dataset is the small number of images and the lack of diversity in the annotations. As stated by the authors [77], the dataset is also biased towards the most frequent relations and does not contain a lot of rare relations. The large number of object classes for the small number of training images makes it also difficult for object detection.

A few months after the release of VRD, the Visual Genome dataset was proposed by Krishna et al. [8]. This dataset is the first large-scale densely annotated dataset for SGG. It contains 108,077 images with 53,304 object classes and 29,086 predicate classes. The dataset has been

annotated by human annotators in the form of region captions using an open-vocabulary class format. Then, each visual caption was parsed to a $\langle subject, predicate, object \rangle$ format using different linguistic rules and corresponding bounding boxes were extracted from the image in a matching process. This semi-automatic annotation process resulted in many classes with a lot of ambiguous and generic names such as the class "background" or "room". The dataset contains an average of 22 relations per image, with a total of 2,316,063 annotated relations. Due to the large scale of the dataset and the number of redundant or ambiguous object and predicate classes, this dataset has rarely been used as it is for SGG [78]. Quickly nonetheless the SGG community has proposed to use a subset of this dataset [12], VG150, which is composed of annotations of the top 150 object classes and top 50 predicate classes of Visual Genome, for a total of 636,722 annotated relations across 89,168 images. Even though a smaller amount of relations is used, the VG150 split suffers from the same long tail distribution of predicate and ambiguous annotations bias as the original dataset [18], [20]. In fact, due to the over-representation of vague predicate classes over more fine-grained ones, the dataset is heavily unbalanced (i.e. long tail distribution). In Figure 3.1 we display the actual distribution of predicate, where we can observe that the first class “on” is annotated 6,549 times more (196,465 samples) than the last class “flying in” (30 samples). The first 6 classes represent more than 75% of the total number of annotations. This long-tail distribution can lead to models biased towards the most frequent relations and that will perform poorly on rare relations. In Figure 3.1 we can also observe the second major problem of VG150: redundant classes. Some classes such as “wearing” and “wears” are actually the same class but have been annotated differently by different annotators. This redundancy in the annotations can lead to confusion in the learning process of the model and can also bias the learning process towards the most frequent relations.

The VG178 split has also been proposed [65], with a similar objective as VG150 but by selecting object classes with more than 1,000 samples and predicates with more than 100 samples, resulting in 178 object and 49 predicate classes.

The GQA dataset [15] is a subset of 85,638 images of Visual Genome with the addition of question-answer pairs. Regarding scene graph annotations, GQA sees the addition of *to the left* and *to the right* relations which have been automatically generated using bounding box coordinates for every pair. It also incorporates some manual refinement of existing annotations of Visual Genome. GQA is composed of 1,703 objects and 310 predicate classes and possesses 471,614 relations annotated other than *to the left* and *to the right*. Due to the sparsity of annotations and the large number of classes, GQA is a more challenging dataset for SGG than VG150 and is even more unbalanced [79]. Due to these two issues, the dataset as not been used a lot by the community in comparison to VG150 [16].

The OpenImage dataset [17] is the largest fine-grained annotated dataset in Computer Vision, with bounding boxes, captions, question-answer pairs and segmentation masks annotated for

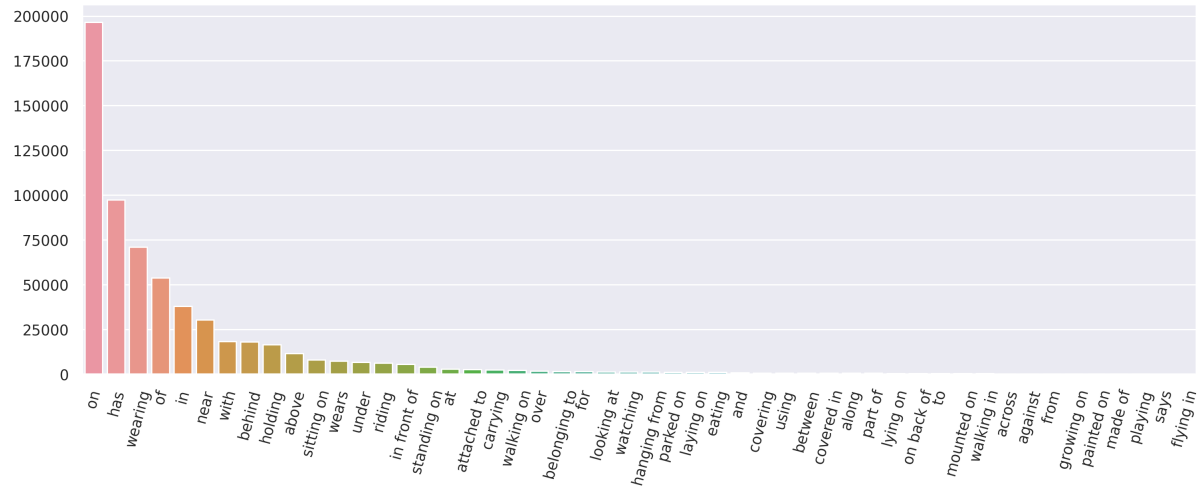


Figure 3.1: Distribution of predicate classes in VG150. The last class, *flying in*, is only annotated 56 times.

more than 1.7 million images. In the 6th release, OpenImageV6, a subset of 133,503 images sees an addition of relations annotations, making it the largest dataset for SGG by number of images. The dataset is composed of 601 object classes and 30 predicate classes. The dataset is nonetheless very scarce in terms of relations annotations, with only 367,914 relations annotated (less than 2.75 per image on average). The dataset is also biased towards the most frequent relations and only contains a small amount of triplets due to the small number of predicates.

In 2022, a new dataset has been proposed to correct the biases of previous datasets. The Panoptic Scene Graph dataset [20] is composed of 48,508 images densely annotated with segmentation masks and bounding boxes for 133 object classes and 56 predicate classes. All classes have been carefully chosen by authors (in contrast to VG and GQA) to represent non-ambiguous objects and predicates with low polysemy. Annotators have also been trained to annotate relations consistently to avoid annotating similar relations with different predicates. This process resulted in a high-quality densely annotated set of images taken from COCO [45] and VG. However, the strategy employed for the PSG dataset still does not enforce the use of a clear taxonomy of relations, which can lead to confusion in the learning process of the model. At the same time, annotator are still free of choosing which relations to annotate, which can lead to disparity in the annotations quality.

3.1.2 Related Datasets

One of the earliest definitions of a scene graph [54] can be stated as "the aggregation of any relation that can describe a scene" which does not constrain the type of relations that can be annotated in an SGG dataset. However, other tasks have proposed annotated Scene Graph

datasets with more constraints, such as the task of Human-Object Interaction detection (HOI). HOI datasets provide graph annotations centered around activities and actions. In HOI datasets, each subject of a relation is a person instance and most of the predicates are action-related. These constraints lower drastically the expressiveness of the data and make it less suitable for general SGG tasks such as VQA or Image Captioning. The most used HOI datasets are HICO-Det [80], V-COCO [81], and EPIC-KITCHENS [82].

In contrast to HOI which is centered around representing human activities, 3D scene graphs have been proposed to represent spatial relations between objects in 3D scans (RGB-D or Point-Cloud representations). The most used datasets in this area are 3D Scene Graphs [83] and 3DSGG [84]. Again here the constraints of the data make it less suitable for general SGG as the relations are mostly topological relations between furniture and common objects. Humans and activities are not represented in these datasets.

Finally, a set of Video SGG datasets has been proposed to represent relations between objects in videos. The most used dataset for this task is ActionGenome [85], which comprises 234,253 annotated frames with 36 object classes and 26 predicates. Even though some papers are referring to the task of predicting relations on Action Genome as "SGG" [30], [31], [86], the task is more related to HOI detection as the relations in Action Genome are only action-related and the subject is always a person instance. To this date, no video dataset combines all types of relations in a single dataset, without being human-centered, which is the case for Visual Genome and other image-based datasets.

3.1.3 Data Curation Approaches Based on Visual Genome

To counter the predicate long-tail bias of Visual Genome, a few approaches have been proposed in the past few years, resulting in the subtask of U-SGG [19], [21], [22], [87], [88], [89], [90]. These approaches are mainly focused on balancing the learning process of the model by re-weighting the loss function [91] or by sampling the data differently [92], which make them more related to the task of long-tail learning than SGG as relation learning becomes a secondary objective. It is worth to mentioned here that the task of U-SGG is only a result of the noisy annotations in Visual Genome and other datasets, as there is no incentive to the task of SGG to be biased with long-tail learning in the first place. It is also worth to mention that the amount of effort which has been put into the task of U-SGG from the model side (i.e. attempting to correct the long-tail learning problem by tweaking the learning process) is much more important than the amount of effort put into the task of refining the data itself [93].

To the best of our knowledge, only a few current approaches are considering the Visual Genome dataset biases from a data-centric perspective. In VrR-VG [18], the authors based their assumption on the fact that relations that can be easily inferred with only spatial information from an object pair (i.e. bounding box coordinates) are not visually relevant. Based on this fact,

the authors have pruned the original Visual Genome dataset to remove common relations. This led to sparse annotations where only rare and very specific relations are annotated for which the use in downstream tasks is very limited, while at the same time limiting the learning capabilities of SGG models. Other approaches are focusing on balancing the predicate distribution [94] or filtering similar or vague predicates [75] to improve the relevance of the annotations. However, these methods assume a consistent use of the same predicate across the annotations which is not true due to the inherent *polysemy* of natural language, as explained before [95]. In VG-KR, Wang et al. [32] have proposed to extract from the Visual Genome dataset key relations to form a concise and expressive smaller data split, with a set of 1 to 10 key relations per image in addition to common relations. To do so, they first selected the most frequent 200 object and 80 predicate classes in the VG dataset and then used COCO captions to match important triplets using common WordNet [96] synsets. This resulted in VG-KR, a subset of VG with 26,992 images and 250,755 relation instances. Even though the key relations annotated are deemed "more informative" because of the selection process, the dataset still suffers from the same biases as VG150 and Visual Genome, as the annotations are still noisy and ambiguous (no explicit curation has been performed to solve this last point).

	Dataset	**Number of Images**	**Object Categories**	**Predicate Categories**	**Number of Relations**	**Average Graph Size**
Original	Visual Genome [8]	108,073	95,394	33,121	2,316,063	22.32
	OpenImageV6 [17]	133,503	601	30	367,914	2.75
	VRD [77]	5,000	100	70	37,993	7.59
	PSG [20]	48,749	133	56	275,371	5.65
VG-based	GQA [15]	85,638	1,703	310	471,614	5.51
	VrR-VG [18]	56,254	1,321	117	176,488	3.13
	VG150 [12]	89,168	150	50	636,722	7.14
	VG178 [65]	91,753	178	49	646,267	7.04
	VG-KR [32]	26,992	200	80	250,755	9.29

Table 3.1: Comparison of different SGG datasets.

We display in Table 3.1 a comparison of the most commonly used datasets for SGG. A first consideration when building SGG datasets for real-world applications is the number of object classes. In fact, most object detector are not very efficient for learning a high number of classes. This is one reason why the authors of VG150 [12] chose a relatively small amount of object classes in the first place. Regarding relations annotations, datasets with a higher connectivity (i.e. higher average graph size) should benefit the learning process of SGG models as the inter-dependencies between relations are easier to capture. However, the balance between the number of object and predicate classes and the number of relations is crucial for the learning

process because a high number of possible predicates for a $\langle subject, object \rangle$ pair can make the data distribution long-tailed and the learning difficult [38]. To generate a new dataset targeting domestic applications, one solution could be to re-annotate images from indoor scenes in Visual Genome. However, manually generating new annotations is a time-consuming and expensive process, thus we propose instead to leverage the existing annotations in Visual Genome and refine them to improve the quality of the data. In the following sections, we will use the raw data from Visual Genome as a base for our experiments to refine annotations and extract clean and high-quality data from the mass of noisy annotations.

3.2 Irrelevant Data

As the Visual Genome dataset is annotated by human annotators in free-form text, it can be considered a good representation of information perceived by humans in images. That being said, the dataset has been extensively annotated, with annotators being tasked to annotate the maximum of relations they can find in an image. In contrast to the gist, which represents the minimal amount of information needed to extract the meaning of the scene [33], annotations in Visual Genome can contain irrelevant data that does not contribute to the gist of the scene. In this work, we consider two types of irrelevant data: data that is redundant with the information we commonly already know *a priori* about the image or data that is superfluous for the gist.

Identifying superfluous information without actual gist annotations can be challenging, however, identifying redundant information can be done by analyzing information in the scene graph that can be known without the need for visual information. For instance, the relation $\langle person, has, head \rangle$ is a relation that can be considered irrelevant as it is invariant and can be known with external knowledge of the object $person$. In contrast, the relation $\langle person, wearing, hat \rangle$ is a relevant relation as it depends on the visual features of the scene. We call the overrepresentation of invariant relations in Visual Genome the *invariant relationship bias*. Intuitively, invariant relations are of a different type than other relations as they convey different information. Invariant relations are for instance not likely to be characterized by action-related predicates such as *eating* or *holding*. To specifically identify irrelevant relations, we need to categorize them with the help of a clear taxonomy of relations.

3.2.1 A Taxonomy of Relations

It is ubiquitous to us that objects in a scene are related to each other in various ways. Compositional relations can represent actions (such as *person riding bike*) or spatial relations (such as *bike next to tree*). In prior work, Zellers et al. [38] proposed a taxonomy of relations in the Visual Genome dataset. They defined relations as *geometric*, *semantic* and *possessive* by manually clustering the different predicates used by annotators in the dataset. This categorization

possesses some common ground with research in the image gist and scene understanding [33], where for instance Mandler and Parker [50] detail the presence of spatial relations in the gist of the scene. Same for semantic relations later on [51]. However, the taxonomy proposed by Zellers et al. is not exhaustive and does not cover all possible relations that can be found in the dataset. Specifically, the *geometric* category is too broad, and a correct definition will be *topological* which covers relative spatial relations between neighboring objects [75]. The *semantic* category can be refined as *functional*, which is a better wording to define dynamic relations between objects. Finally, the *possessive* category is split into *part-whole* and *attributive* relations. The *part-whole* category covers relations where an object is part of another object, such as $\langle person, has, head \rangle$, whereas the *attributive* category covers relations where an object is an attribute of another object, such as $\langle person, wearing, shirt \rangle$. We provide a comprehensive list of definitions below and a set of examples for each category in Table 3.2.

- ***Functional***: dynamic relations between entities;
- ***Topological***: static spatial relation between any pair of entities;
- ***Part-Whole***: hierarchical and invariant relation between a defined entity (i.e. “whole") and one of its building blocks (i.e. “part");
- ***Attributive***: relation between a physical entity and a non-invariant attribute.

Category	Examples
Functional	person reading book, coat hanging on rack, cat sleeping on bed
Topological	phone on table, person next to window, paper on top of keyboard
Part-whole	person has head, key on keyboard, window on building
Attributive	person wearing jacket, frame has painting, writing on sign

Table 3.2: Example of compositional relations by semantic category.

3.2.2 Data Filtering

Defining relations categories is crucial for the task of SGG as it helps in understanding graph structures and interdependence between relations. More importantly, relation categories can be used to refine the annotations in a scene graph dataset by removing irrelevant relations. Based on the *part-whole* category, we present a new definition of visually relevant relation as follows: *a relation is not relevant if it describes a composition between parts of an entity that is true in a general sense and that could be inferred using external knowledge (e.g.* $\langle man, has, arm \rangle$*).*

Figure 3.2 shows the relations that are most annotated in Visual Genome, where we can see that part-whole relations are prevalent with 47.35% of the total number of occurrences for the

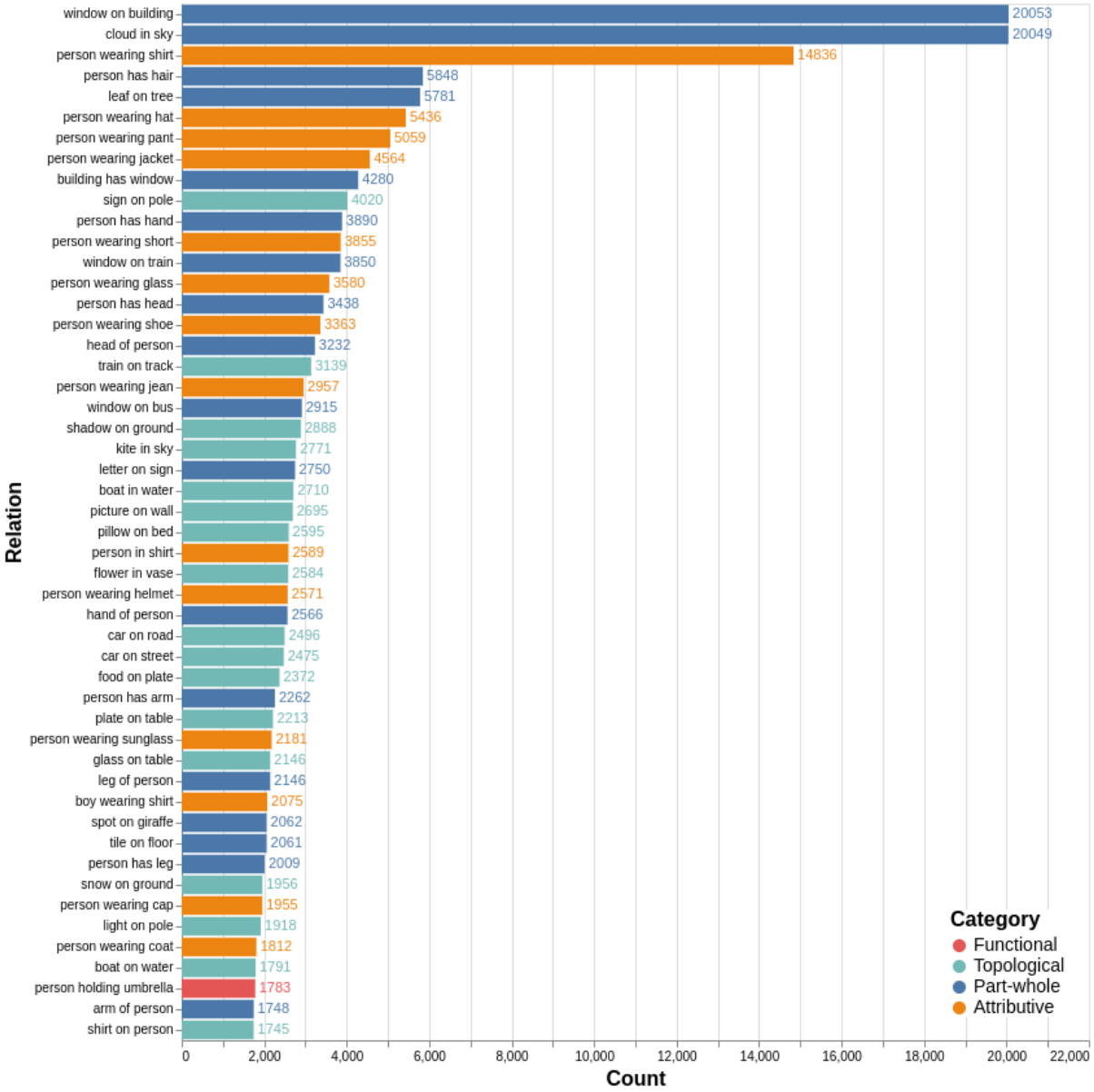


Figure 3.2: Distribution of the top 50 relations in Visual Genome.

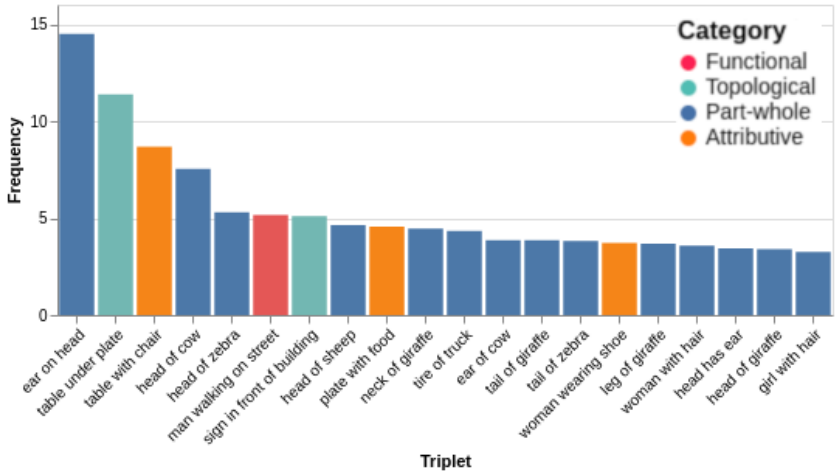


Figure 3.3: Ratio of predicted triplets over ground truth ones on the test set of VG150 by the Motifs model [38]. For clarity, we show only the top 20 triplets with more than 20 occurrences.

top 50 relations. We can, for instance, see the relation $\langle person, has, hair \rangle$ and $\langle leaf, on, tree \rangle$ very high on the list when their importance is debatable for downstream tasks. Moreover, as explained in previous approaches [16], [18], [32], [93], these relations may be biasing the learning process because they are true in the general sense and do not depend on visual features of the scene, which could lead to overfitting of the models. In order to verify this assumption, we conducted an experiment on predictions obtained by the Motifs-TDE model [19], [38] on the test set of the VG150 dataset. Results are shown in Figure 3.3 where we can see that part-whole relations are overly predicted in comparison to the ground truth annotations in the respective images. For instance, the relation $\langle ear, on, head \rangle$ is predicted 14.5 times more than the number of times it is annotated in the dataset. This shows that irrelevant data can bias the learning process of SGG models and hinder their performance. Building upon our new classification of relevant relations, we employed a new approach of filtering the dataset from *part-whole* triplets. To filter categories of relations, previous approaches rely on handcrafted predicate categories [38]. However, this categorization only takes into account the intended meaning of predicates, which suppose that annotations are consistent in the dataset. This assumption is wrong, given the polysemy of natural language [95]. For instance, the relation $\langle man, on, laptop \rangle$ does not represent a *topological* relation but rather a *functional* one, which is not the case with another relation $\langle man, on, bench \rangle$ even though the predicate and subject are the same. On the other hand, it has been noticed that there is a strong correspondence between the knowledge embedded in the Visual Genome and relations contained in linguistic commonsense knowledge sources such as ConceptNet [97] [66]. Thus, instead of manually labeling every triplet in VG, we chose to compare triplet annotations with a subset of ConceptNet [97] that contains only part-whole relations. If a relation has a significant similarity with one from ConceptNet,

Method	Recall	Precision	F1
Predicate only [38]	0.43	0.62	0.51
Lexical similarity	0.81	0.53	0.64
Glove 6B 300d *(cos=0.7)*	**0.88**	0.5	0.64
RoBERTa-large-v1 *(cos=0.7)*	0.75	0.58	0.66
MiniLM-L6-v2 *(cos=0.7)*	0.74	0.67	0.7
MpNet-base-v2 *(cos=0.75)*	0.64	**0.83**	**0.72**

Table 3.3: Part-whole relations filtering by comparing with ConceptNet, evaluation on a set of 1000 random samples.

then we can filter all its occurrences from the original data. We used the *part-whole* subset of ConceptNet, following the ontology introduced by Illievski et al. [98] with the relations 'PartOf', 'HasA', and 'MadeOf'. Then, we used different approaches to categorize relations between part-whole and non-part-whole from textual annotations only. To evaluate the performance of this filtering, we manually annotated a subset of 1000 random relations from Visual Genome. First, we evaluated the filtering using lexical similarity between $\langle subject, object \rangle$ pair in Visual Genome and ConceptNet. Second, we compared the representation of $\langle subject, predicate, object \rangle$ triplet in Glove embeddings [46] with those from ConceptNet using the cosine similarity. The cosine similarity σ between vector representations $B \in ConceptNet$ and $A \in VG$ is defined as:

$$\sigma = \frac{A \cdot B}{\|A\|\|B\|} \tag{3.1}$$

Third, we used different pre-trained Sentence Transformers [99] models to generate sentence similarity embeddings. Finally, we compared those approaches with the predicate-only classification proposed by [38] in which the 50 predicate classes were classified within semantic, geometric, and possessive classes. Results displayed in Table 3.3 show that the classification by [38] resulted in the lowest score, this is due to the polysemy of predicates, as explained before. Approaches based on Sentence-Transformers, as they have been pre-trained on a large corpus of texts, can generalize easily and give the best performance. In the choice of embeddings, we prioritized precision over recall as we do not want to discard anything else than *part-whole* relations. The *all-mpnet-base-v2* [1] model has shown the best performance in the task, giving satisfactory trade-off between precision and F1 score. This result is consistent with previous work as this model is ranked 5 in the task of Sentence Similarity [2].

Using the embeddings produced by *all-mpnet-base-v2*, we were able to extract 36,777 part-whole relations for a total of 416,318 occurrences in VG80K (18% of the annotations). Before removing those annotations from the original samples, we ensured that no other types of rela-

[1] `https://huggingface.co/sentence-transformers/all-mpnet-base-v2`
[2] `https://huggingface.co/spaces/mteb/leaderboard` *accessed on the 21/11/2022.*

tionships were dependent on them. This step is important because by removing some part-whole relations we could lose important structural information of the visual content. For instance, the sub-graph:

$$person \xrightarrow{\text{has}} hand \xrightarrow{\text{holding}} cup \tag{3.2}$$

describes a functional relation between the entity `person` and `cup`, even if the relation $\langle person, has, hand \rangle$ is classified as a part-whole relation by our method. In this case, the method proposed in this work can be applied as follows: we added a set of weights $w : E \to \mathbb{R}$ to the original graph $G = (V, E)$ such as $w = 1$ if the edge is a part-whole relation and 0 otherwise. Given this graph, we performed a pruning strategy that iterates through all edges and removed those that were only dependent on other part-whole relations. This removed from the graph relations deemed as irrelevant to the context of the scene.

To validate this approach, we conducted multiple experiments with different SGG models. To be able to compare with previous approaches, we used the VG150 dataset [12]. This subset of the data is the most used in the community. For a fair comparison, we kept the same train/val/test split of images. However, due to the removal of part-whole relations, some object and predicate classes ended with almost no annotations in the dataset (for instance the class "head" or the predicate class "belonging to"). To address this issue, we choose to replace those classes with new classes with higher occurrences in the dataset, following the frequency method by previous work [12]. We call this new split **VG150-cur**, standing for curated VG150 dataset. VG150-cur possesses 77% of similar object classes than VG150 and 88% of similar predicate classes. Due to the addition of new classes, the number of relations in the dataset stays similar to VG150 (622,705 for VG150 versus 636,175 for VG150-cur) which makes the comparison fair.

We follow previous work in the area [12], [19], [37], [38] by evaluating our approach on three distinct (but related) tasks, namely Predicate Classification *PredCls*, Scene Graph Classification *SGCls*, and SGG *SGGen*. *PredCls* concentrates on predicting a relation, given the bounding boxes and $\langle subject, object \rangle$ pairs. *SGCls* is analogous to *PredCls*, except that $\langle subject, object \rangle$ pairs are not known *a priori*, and they need to be inferred by the model. Finally, *SGGen* assumes no prior knowledge; thus, the task included the prediction of object regions, pairs, and relations. To be consistent with other related work, a selection of the most used baseline models were trained: IMP [12], Motifs [38], and VCTree [37]. For Motifs and VCTree, we trained the TDE version introduced in [19]. As other metrics have proven to be ineffective in measuring the performance for both the head and tail classes [37], we used the meanRecall@K metric introduced in [37]. We retrained every model using the code provided by the authors [19][3], whereby the original parameters were maintained, except for the batch size and learning rate that were fit to our hardware requirements. The training was conducted with a batch size of 32 and a base learning rate of 0.02 on one Nvidia RTX3090 within 20000 iterations (approximately 10 epochs)

[3] `https://github.com/KaihuaTang/Scene-Graph-Benchmark.pytorch`

Models	Dataset	PredCls mR@20/50/100	SGCls mR@20/50/100	SGGen mR@20/50/100	Improv. (avg.)
IMP [12]	VG150	8.8/10.80/11.62	4.63/5.82/6.42	2.76/4.02/5.0	-
	VG150-cur	9.61/12.61/13.92	6.99/8.74/9.44	4.09/6.21/7.41	↑ **28%**
Motifs-TDE [19]	VG150	18.5/25.5/29.1	9.8/13.1/14.9	5.8/8.2/9.8	-
	VG150-cur	21.38/30.90/36.58	13.75/18.55/21.54	10.49/14.28/17.10	↑ **37%**
VCTree-TDE [19]	VG150	18.4/25.4/28.7	8.9/12.2/14.0	6.9/9.3/11.1	-
	VG150-cur	22.03/32.25/38.24	13.73/18.14/20.70	10.89/14.52/17.09	↑ **39 %**

Table 3.4: Reported performance of baseline models on the two splits of Visual Genome. Improvements are the relative average against the baseline VG150.

or 30000 iterations for SGGen. IMP was retrained on the baseline split (VG150) with the above settings, this is why the reported results in Table 3.4 are slightly different from those reported in the original paper [19]. For comparison, the same training/validation/test split of the original VG150 was used for all datasets.

Results are displayed in Table 3.4. We can see that the performance of all models is significantly improved on the VG150-cur split which does not contain irrelevant annotations. We also display the relative improvement of the average meanRecall@K for each model over the baseline dataset VG150. We can see that the average relative improvement is 28% for IMP, 37% for Motifs, and 39% for VCTree. This shows that our filtering method is effective in improving the performance of SGG models by removing irrelevant data from the dataset.

With this small experiment, we have shown that the quality of annotations in the dataset is crucial for the performance of SGG models. We have also shown that the presence of certain types of invariant relations (part-whole) can bias the learning process of SGG models and hinder their performance. Moreover, as SGG models are very sensitive to data distribution, we observed a consequent improvement in performance by targeting specific data biases rather than changing the model architecture. For a comparison, approaches in SGG traditionally improve the meanRecall@K by only a few % by focusing on models architecture [16], [19]. In contrast, our approach improves the meanRecall@K by a relative 28 to 39% with a data-centric approach. This shows that improving the quality of the data is a more effective way to improve the performance of SGG models than changing the model architecture. These findings are consistent with previous data-centric approaches for SGG [95].

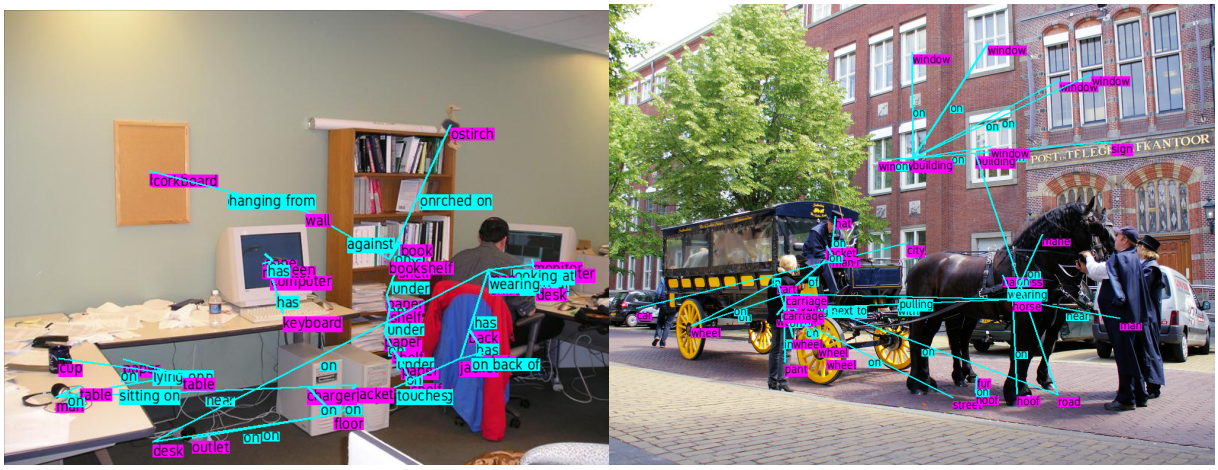
3.3 Class Selection

A second problem with annotating relations for SGG datasets is the background-foreground problem, or more simply, which subject-object pair should have a relation in the first place? Indeed, the number of possible relations (even after simplifying the problem to a standard classification task) will grow quadratically with the number of object proposals, which will also

require a lot of effort and time for extensive annotation. In practice, annotators cannot annotate all pairs but rather focus on a small part of them. But again here, the problem is that the edge between what should/should not be annotated is left to the appreciation of the annotators. If we look at annotations in the resulting dataset, we can observe an interesting phenomenon: relations are not distributed uniformly across the image. If we take two reference images from two very distinct contexts such as a Figure 3.4a and Figure 3.4b we can observe that relations are distributed across important and meaningful subregions. In Figure 3.4a, relations are concentrated on the person sitting at the desk, and in Figure 3.4b relations are concentrated on the foreground objects (the carriage and horses), and relations with the background (trees, cars, etc...) are neglected. The exact explanation of this phenomenon will be further investigated in the future. From those qualitative examples, we can also observe that highly connected regions can be different from what we could expect to be visually salient regions. For instance, in Figure 3.4a, the person is not the most visually salient object in the image, but it is the most connected one. This observation is consistent with previous work in the area of image understanding [100] where it has been shown that the content of the gist is not necessarily related to the most visually salient regions.

In SGG, a highly connected sub-graph that can be encountered multiple times in the dataset is called a *motifs* [38]. These sub-structures are defined as "structural regularities" in the data distribution, i.e. frequent inter-dependent relations. We postulate here that a high-quality scene graph dataset should be centered around those motifs to keep the maximum of the meaning conveyed not only by relations but also by the structure of the graph. In practice, this means that a dataset with more connected regions will improve the frequency of motifs which will lead to the following benefits: (1) an increase in model performance as more patterns can be spotted and (2) more meaningful graphs as learning will focus on meaningful regions of the image. To validate this assertion, we propose to generate a new subset of Visual Genome by specifically paying attention to preserving the different motif structures during the class selection process.

As discussed in the previous section, training an SGG model requires the selection of a small subset of classes to reduce the long-tail distribution of the data. In traditional approaches, the number of object classes is reduced to 150. This number has been chosen arbitrarily [12] with the idea of making the object detection task easy. Similarly, the number of predicate classes is 50 to represent a large range of relations possible between those 150 object classes. To choose the classes, the dominant method is to select the most frequent classes in the dataset by number of annotations, for both objects and predicates. However, this approach is not optimal as it does not take into account the connectivity of the graph. In fact, the most frequent classes in the dataset are not necessarily the most connected ones. Selecting classes this way will result in poorly connected graphs with relations that have no inter-dependences on each other. To address this issue, we propose a new method to select the most connected classes in the dataset.



(a) (b)

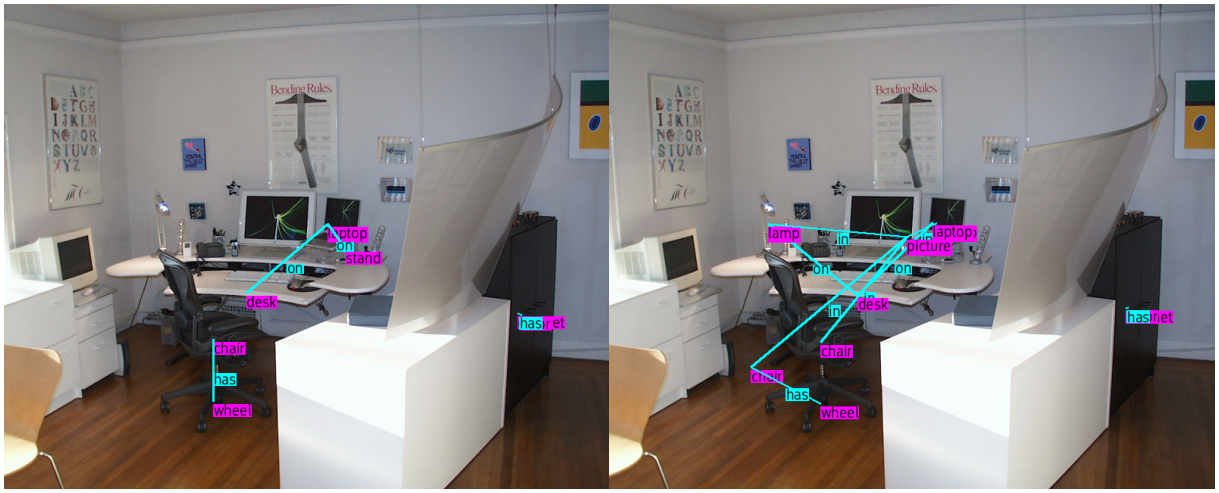
Figure 3.4: A visualization of the connectivity of Scene Graphs: two images annotated from the Visual Genome dataset [8].

We based our selection algorithm on the average graph size, trying to optimize the set of top classes that maximize the average connectivity of graphs over the entire dataset. Thus, for every graph G, the most connected object ($\hat{\mathbf{o}}$) and predicate ($\hat{\mathbf{p}}$) classes were selected from the set of n images as follows:

$$\theta(\hat{\mathbf{o}}, \hat{\mathbf{p}}) = \max_{\hat{\mathbf{o}}, \hat{\mathbf{p}}} \sum_{k=1}^{n} Conn(\hat{\mathbf{o}}, \hat{\mathbf{p}}, G_k) \tag{3.3}$$

$$Conn(\hat{\mathbf{o}}, \hat{\mathbf{p}}, G) = |G(u, v, w)|, w \in \hat{\mathbf{p}} \vee [u, v] \in \hat{\mathbf{o}} \tag{3.4}$$

To be consistent with VG150, we chose $|\hat{\mathbf{o}}| = 150$ and $|\hat{\mathbf{p}}| = 50$. As this is a complex optimization problem, a satisfying solution can be found by first optimizing $\theta(\hat{\mathbf{o}})$ and then $\theta(\hat{\mathbf{o}}, \hat{\mathbf{p}})$ with a fixed set of classes $\hat{\mathbf{o}}$. We applied this method to the original data and obtained a new split that we call VG150-connected (**VG150-con**). This split possesses a significantly higher number of relations (22% more than VG150), with an average graph size $\overline{s}$ of 8.37 versus 6.98 for VG150. More interestingly, we see a net improvement in the average vertex degree in VG150-con, moving up from 2.02 to 2.2. This shows that relations are also more interdependent and thus should benefit the context learning of SGG models. Regarding class distribution, VG150-con possesses 77% of similar object classes than VG150 and 92% of similar predicate classes. We display a comparison between resulting annotations in VG150-con and the original method based on class frequencies in Figure 3.5. We can observe here that the annotations from Figure 3.5a are loosely connected and split apart the image, which makes it difficult to reconstruct the layout of the scene with only graph annotations. In contrast, the annotations from Figure 3.5b are more connected and form a more meaningful graph structure. This shows that our method is effective in selecting interesting *motifs* from the data distribution.



(a) (b)

Figure 3.5: Comparison between annotations selected using class frequencies (a) and our method based on connectivity (b).

Models	Dataset	PredCls mR@20/50/100	SGCls mR@20/50/100	SGGen mR@20/50/100	Improv. (avg.)
IMP [12]	VG150	8.8/10.80/11.62	4.63/5.82/6.42	2.76/4.02/5.0	-
	VG150-con	8.8/11.9/13.35	5.63/6.76/7.16	2.59/4.26/5.61	↑ **10.3%**
Motifs-TDE [19]	VG150	18.5/25.5/29.1	9.8/13.1/14.9	5.8/8.2/9.8	-
	VG150-con	20.38/28.76/34.06	10.3/14.6/17.25	8.15/11.53/13.15	↑ **16.1%**
VCTree-TDE [19]	VG150	18.4/25.4/28.7	8.9/12.2/14.0	6.9/9.3/11.1	-
	VG150-con	22.5/31.22/37.02	9.38/13.32/15.29	8.56/10.84/13.09	↑ **19.5 %**

Table 3.5: Reported performance of baseline models on the two splits of Visual Genome. Improvements are the relative average against the baseline VG150.

To validate our approach, we conducted experiments with three different baseline SGG models. We follow previous work in the area [12], [19], [37], [38] by evaluating our approach on three distinct (but related) tasks, namely Predicate Classification *PredCls*, Scene Graph Classification *SGCls*, and SGG *SGGen*. Similar to our previous experiments, a selection of the most used baseline models were trained: IMP [12], Motifs [38], and VCTree [37]. For Motifs and VCTree, we trained the TDE version introduced in [19]. We can see that the performance of all models is significantly improved on the VG150-con split. We also display the relative improvement of the average meanRecall@K for each model. We can see that the average improvement is 15.3% for IMP, 16.1% for Motifs, and 19.5% for VCTree. This shows that our selection method is effective in improving the performance of SGG models with a more connected dataset.

It is no surprise that model architectures that specifically rely on the presence of motifs for learning such as VCTree [37] or NeuralMotifs [38] are the ones that benefit the most from our method. However, because we replace some classes in both VG150-cur and VG150-con,

Datasets	Connectivity		Samples		Pred. Distribution	
	$\bar{d}(v)$	$\bar{s}(G)$	#Rels	#Triplets	ID [102]	LRID [103]
VG80K	2.34	19.02	2,316,063	514,526	29,278	13.75
VG150	2.02	6.98	622,705	35,412	40.7	2.99
VG150-con	2.20	8.38	799,412	44,851	40.69	2.98
VG150-cur	2.12	7.14	636,175	41,164	39.68	2.79

Table 3.6: Graph's connectivity and size of the different splits; where $\bar{d}(v)$ represents the average vertex degree; $\bar{s}(G)$ the average graph size; #Rels is the total number of relations samples, and #Triplets is the number of different triplets.

the imbalance in the predicate distribution could be different from the original VG150 which could facilitate learning. In fact, it is known that SGG models are heavily biased towards the head predicate classes [19], [90], [101] and that the tail classes are poorly learned. We display in Table 3.6 some statistics about the sample distribution in the different splits. We compare the Imbalance Degree (ID) [102] of the different splits of Visual Genome. ID compares the normalized distance between the actual distribution and a perfect distribution across all classes. However, as highlighted in [103], this metric is highly dependent on the type of distance chosen as well as the number of minority classes. As the Imbalance Degree does not give a full picture of the imbalance in multi-class distribution, we also measure the likelihood-ratio imbalance degree (LRID) [103] as follows:

$$LRID = -2\sum_{c=1}^{C} n_c \log \frac{N}{Cn_c} \tag{3.5}$$

Where C is the number of unique classes, n_c is the frequency of each class and N is the perfect distribution. This metric tests the actual distribution against a complete balance distribution of the data, and is reported to be more accurate for multi-class problems [103]. From Table 3.6 we observe that the imbalance over the global distribution is very similar, with a small advantage for VG150-cur in both ID and LRID. These findings confirm that our method does not change the imbalance of the data and thus improvements can be attributed to the better quality of the data rather than the change in the data distribution. The increase in connectivity by looking at the average vertex degree or average graph size also confirms this claim.

Finally, we can observe an increase in the number of triplets (the number of different $\langle subject, predicate, object \rangle$ combinations in the dataset) for both VG150-con and VG150-cur which demonstrate a more diverse data split and thus more challenging learning [18]. For VG150-cur we believe that this better diversity comes from the removal of part-whole related classes that had poor diversity, due to the invariant bias. Regarding VG150-con, the diversity comes from the largely higher number of samples (799,412 versus 622,705 annotations) which statistically would increase the number of possible combinations.

In this section, we have shown that connectivity is an important factor in the learning of SGG models. We have proposed to select classes by connectivity rather than overall frequency and have shown that this method is effective in improving the performance of SGG models. This method, coupled with the previously proposed filtering method for part-whole relations (see Section 3.2), can be used to generate better-quality data splits from the original annotations of the Visual Genome dataset. In the next section, we propose to leverage these methods to generate a new dataset for the task of SGG for the domestic context.

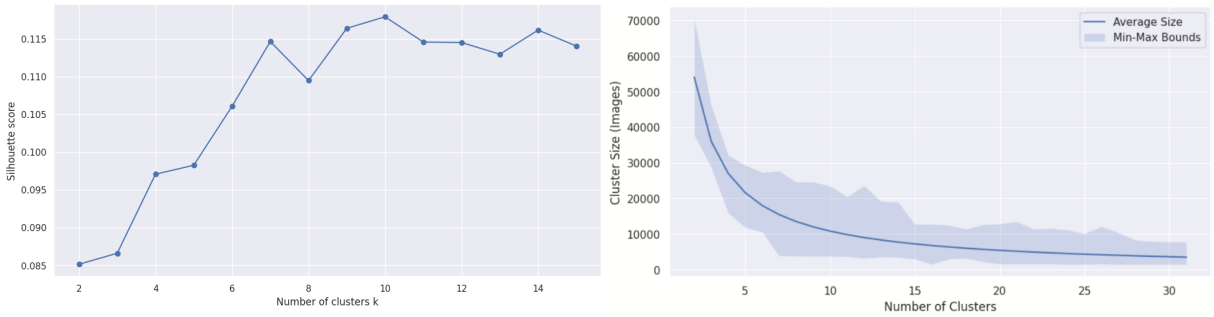
3.4 Application: The IndoorVG Dataset

Annotating data for the task of SGG is very time-consuming as not only bounding boxes and labels need to be annotated but also object pairs and predicates in every image. Noticing that the large-scale dataset Visual Genome [8] contains densely annotated indoor scene snapshots, we focused on leveraging the original annotations provided by authors [8]. Annotations in Visual Genome were collected by annotators in the form of region captions (i.e. annotators were asked to describe part of the image using free-form text annotations), and then various parsing techniques were applied to retrieve $\langle subject, predicate, object \rangle$ triplets for each region, as well as corresponding bounding boxes. This process has resulted in free-form labels that are noisy and contain duplicate bounding boxes, ambiguous classes, or synonyms. The raw dataset is thus impossible to use as it is and requires a lot of preprocessing to be used by SGG models. On the other hand, the number of annotations per image and the diversity of classes make it very suitable for generating new data splits oriented in specific contexts. In the following, we build on this assumption to generate a new dataset for the task of SGG in the domestic context.

3.4.1 Clustering Indoor Scenes

The Visual Genome dataset contains 108,077 images that represent different contexts, such as *household*, *sports*, or *streets* that can be clustered by looking at the content of regions captions. In this work, we focus on domestic context and thus our goal is to select a subset of images from Visual Genome that represent indoor scenes. To do so, we introduce a method that uses Sentence Embeddings Transformer to compute sentence embeddings from each region caption using the MpNet pre-trained model [4], then region embeddings are averaged per image and clusters are computed using the *k-means* algorithm. This method is more beneficial than classical image clustering based on visual features as textual features are more discriminative and can be used to generate better clusters [104]. To find the best number of clusters for *k-means*, we ran experiments with different numbers of clusters, ranging from 2 to 15. We do not report results for more than 15 clusters as we observed that the average number of images per cluster greatly

[4] `https://huggingface.co/sentence-transformers/all-mpnet-base-v2`



(a) Silhouette score for different $k \in [2; 15]$.

(b) Cluster size distribution for $k \in [2; 30]$.

Figure 3.6: Comparison between cluster sizes and average silhouette score for the semantic clusters extracted from Visual Genome [8].

decreased after this value (see Figure 3.6b), hindering the potential of extracting a large subset of images. We found out that 10 was the best value, with respect to the average silhouette score (see Figure 3.6a). The silhouette score computes the average distance a between points in a cluster and the distance b between points with the nearest cluster. It can be expressed as follows, for every point $i \in C$:

$$s(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))}, if|C|\rangle 1 \tag{3.6}$$

By averaging the silhouette score over data points, we obtain the average silhouette score. The silhouette score is bounded between -1 and 1, where a score close to 1 indicates that the point is far away from the neighboring clusters. A score close to 0 indicates that the point is close to the decision boundary between two neighboring clusters and a score close to -1 indicates that the point might have been assigned to the wrong cluster. We display the silhouette score for different cluster sizes in Figure 3.6. We can see that the silhouette score is maximized for a cluster size of 10 with a value of 0.1179.

To validate this clustering, we use a T-SNE visualization in Figure 3.7. T-SNE is a technique used to visualize high-dimensional data by reducing it to two or three dimensions [105]. In the visualization, we can see a clear distinction between image clusters, that show clear patterns or motifs [38] present in similar contexts. This suggests that indoor scenes can be easily clustered by looking at the representation in embedding space of region captions. The obtained clusters have a size ranging from 3,927 to 20,698 images. For this work, we selected a cluster that encapsulates indoor scenes with a size of 17,740 images. We call the new distribution of Visual Genome extracted from this cluster the **IndoorVG** split.

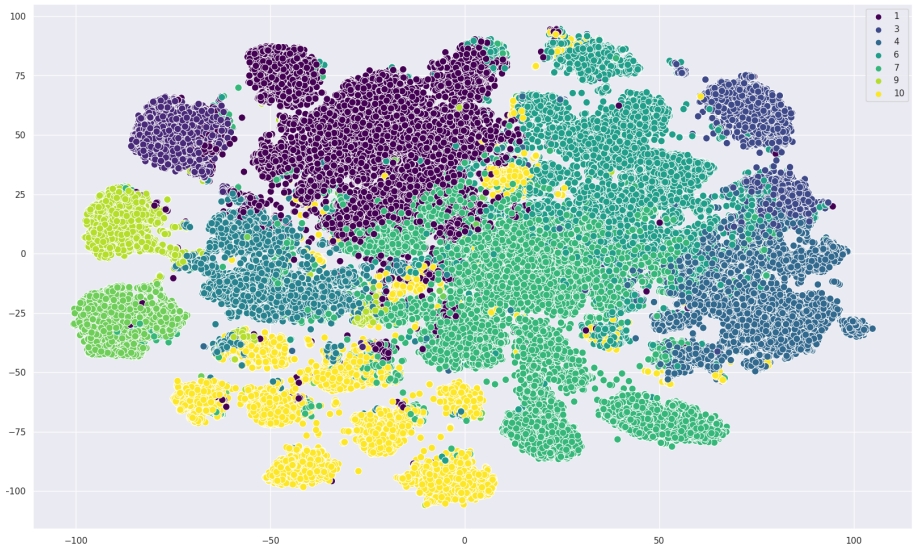


Figure 3.7: T-SNE visualization of Visual Genome image clusters, with k clusters = 10.

3.4.2 Selecting Classes

As stated in numerous works (e.g. [38], [106]), annotations from Visual Genome are noisy and contain duplicate bounding boxes, ambiguous classes, or synonyms. For all images in the IndoorVG subset, annotations were processed as follows: for object regions, we replicated the approach proposed by [12] to merge bounding boxes with an IoU (IoU) greater than or equal to 0.9. For the textual annotations, we also followed [12] to remove stop-words and punctuation using the alias dictionaries provided by the authors of the dataset [5]. We then merged synonyms of object and predicate classes using WordNet synsets [96]. At this point, we obtained a long-tailed data split of 11,620 objects and 5,407 predicate classes where most of the classes still possess only one sample. As a dataset of this scale would be of no use for real-world applications, only the most representative relations were used from the selection of images from IndoorVG. To solve this problem, we used the Connectivity-based Selection method described in Section 3.3 to select the most connected classes in the dataset. We set $|\hat{\mathbf{o}}| = 130$ and $|\hat{\mathbf{p}}| = 50$ and ran the method on the image subset obtained previously using *kmeans* clustering. This resulted in a new split with 149,020 relation samples across 16,221 images. This new split still contains duplicate or ambiguous object classes or predicate classes. As a result, we manually removed classes that represent abstract concepts (such as “object", “edge" or “background") or groups of entities (such as the class “people" and “food") to keep only physical, tangible, and unitary object classes. We also set a minimum of 100 occurrences for each object or predicate class as we believe that below this threshold learning would be difficult. The final resulting split is composed of 84 object classes and 37 predicate classes with a total of 98,824 annotated relations and 9,095

[5] `http://visualgenome.org/api/v0/api_home.html`

triplet combinations across 14,674 images. We maintain the same train/val/test split ratio as in VG150 with 0.65/0.05/0.3 respectively. We will call this new split the **IndoorVG** dataset from now on. This split contains traditional triplets annotations and corresponding bounding boxes from Visual Genome. However, these annotations may not be sufficient to model the gist of the scene. In fact, as we have seen in Section 3.2, relation categories are also of utmost importance for the learning of SGG models. In the next section, we propose to extend the annotations of the IndoorVG dataset with this additional information.

3.4.3 Relation Categories

Traditional Scene Graphs annotations only contain bounding boxes, predicate and subject-object pairs. However, in real-world applications, it is important to have more detailed annotations such as object attributes, spatial relations, or object functional interactions. Inspired by our findings on the impact of relations categories on the learning of SGG models and the importance of the quality of the data, we propose to extend the annotations of the IndoorVG dataset with additional information. We propose to add to every relation an additional label representing one of the four categories previously introduced: *part-whole*, *functional*, *topological*, or *attributive*.

As explained in Section 3.2, finding the correct category of each triplet is a complex task as we can not annotate triplets by only looking at the predicate due to the polysemy of natural language. In addition, the method proposed in Section 3.2 which uses external knowledge bases is complex to extend to *functional*, *topological*, and *attributive* categories due to their high sparsity in commonsense knowledge sources [98]. To address this issue, we proposed to use a pre-trained Large Language Model (LLM) to classify relations in a few-shot manner. We first manually annotated a set of 1,200 random triplets between the four categories `topological`, `functional`, `part-whole`, and `attributive` by looking at corresponding images. For every triplet, we looked at a set of 5 different images containing the triplet with corresponding bounding box annotations, and we selected the category that best fits the relation for all images. This way, we ensure to have a categorization as close as possible to the actual usage of relations in images from the dataset. Then, we used these samples to fine-tune OpenAI's GPT3.5 [107] LLM to give an effective classification of all triplets in the dataset. Here, we used GPT3.5 to benefit from its pre-training on a large corpus of data that showed some abilities in making inferences about commonsense knowledge [108]. Commonsense knowledge is needed to differentiate between ambiguous cases, such as $\langle man, on, phone \rangle$ (which is a `functionnal` relation, even though the predicate "on" is usually used in `topological` relations). We fine-tuned the *turbo* version of GPT3.5 using a training set of 1,000 triplets and a validation set of 200, results are displayed in Table 3.7.

We also compared the fine-tuning of GPT-3.5 with standard linear regression on different text and sentence embeddings. For text embeddings, we used the average of the $\langle subject, predicate, object \rangle$ word embeddings of the Glove 6B 300d embeddings [46]. For sentence transformers,

Method	Precision	Recall	F1-Score
glove.6B.300d			
attribute	0.73	0.39	0.51
functional	0.83	0.77	0.80
part-whole	0.63	0.67	0.65
topological	0.82	0.91	0.86
macro avg	0.75	0.69	0.71
weighted avg	0.79	0.79	0.78
bge-base-en-v1.5			
attribute	0.61	0.61	0.61
functional	0.87	0.87	0.87
part-whole	0.71	0.56	0.63
topological	0.87	0.89	0.88
macro avg	0.76	0.73	0.75
weighted avg	0.82	0.82	0.82

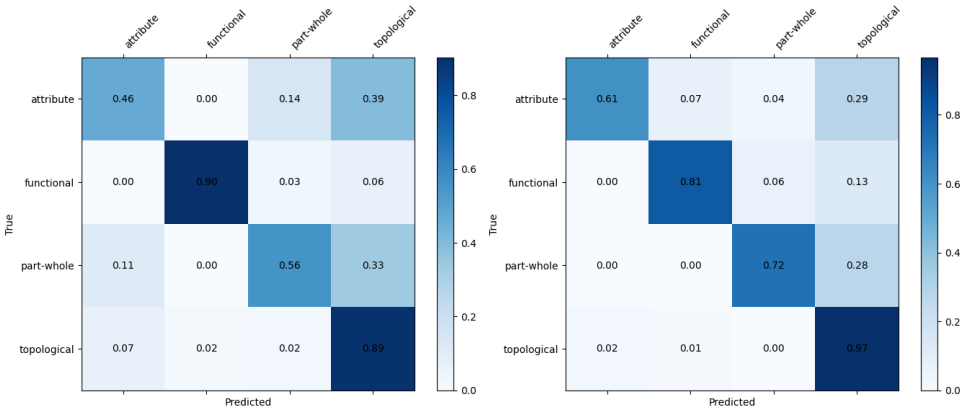
Method	Precision	Recall	F1-Score
BERT-base			
attribute	0.57	0.46	0.51
functional	0.93	0.90	0.92
part-whole	0.56	0.56	0.56
topological	0.85	0.89	0.87
macro avg	0.73	0.70	0.71
weighted avg	0.80	0.80	0.80
gpt-3.5-turbo-0613			
attribute	0.85	0.61	0.71
functional	0.89	0.81	0.85
part-whole	0.81	0.72	0.76
topological	0.87	0.97	0.92
macro avg	0.86	0.78	0.81
weighted avg	**0.87**	**0.87**	**0.86**

Table 3.7: Compositional relations classification with different methods. Left: linear regression with different pre-trained sentence transformer embeddings. Right: fine-tuning of GPT3.5-turbo and BERT models.

we used the BGE-base-en-v1.5 [109] embeddings. We also compared with a fine-tuning of the BERT-base model [110] on the same training and validation sets. We display the results in Table 3.7 where we can see that GPT-3.5 outperforms all other methods with an F1-score of 0.86. This shows that GPT-3.5 is effective in classifying triplets into `topological`, `functional`, `part-whole` and `attributive` categories. Using sentence transformer embeddings or fine-tuning the BERT-base model also gives good results, with an F1-score of 0.82 and 0.80 respectively. This shows that even using smaller models can give good results for the task of triplet classification.

We analyze the confusion matrices of the BERT and GPT3.5 models in Figure 3.8. We can see for both methods a confusion between the attribute or part-whole categories and the topological one. After identifying conflicting cases, this seems to be due to the predicate "on" which is wrongly associated with topological triplets most of the time. For instance, the triplet $\langle keyboard, on, laptop \rangle$ is classified as a topological relation, even though it is a part-whole relation because models tend to have trouble to select the correct syntactical sense of the word. This is a common issue in the field of SGG, as the same predicate can be used in different contexts. This is why we believe that using a LLM is a good approach to classifying relations, however, it may not be optimal. The triplet-level classification applied is still too general to disambiguate rare and complex cases. For instance, we noticed that the triplet $\langle arm, on\ top\ of, man \rangle$ could even be a `part-whole` relation or `topological` relation, depending on the visual features. To solve this issue, future work should consider annotating the relation triplets per image and training a classifier based on the union of the visual features and the triplet embedding.

The effect of selecting a subset of the original annotations and then splitting relations into different categories led to sparse annotations per relation category. In fact, we observed that



(a) (b)

Figure 3.8: Comparison of the confusion matrix for the Relations Classification task using BERT-base (a) and GPT3.5 (b).

only a small percentage of images contained annotations from the 4 different categories at once. To alleviate this issue, data augmentation was used to enlarge the number of annotated relations between existing object pairs.

3.4.4 Data Augmentation

Data Augmentation in SGG aims at re-labeling missed true annotation on the training set to improve performance on the test set. We leveraged the external data-transfer method proposed in [95]. This approach aims at re-labeling missed annotations by ranking new predictions of an SGG model on the set of overlapping bounding boxes. The hint here is that overlapping object regions have a higher probability of forming a compositional relation than non-overlapping ones. Considering U as the set of unannotated relations where $U = P \cup N$, with P as the set of missed true positives and N true negatives, the external transfer T is defined as, for every triplet $(s, \theta, o) \in U$:

$$T(s, \theta, o) = \{p | (p \in R) \wedge (\beta(c_s, p, c_o) \rangle 0) \wedge (IoU(b_s, b_o) \rangle 0)\} \tag{3.7}$$

with R being the set of possible predicates between (s, o) and the IoU of b bounding boxes. β denotes the set of existing relations in the original dataset. This new set of T possible relations is then ranked by confidence and the top predicate θ is selected, following Zhang et al. [95]. As it is, this method cannot discover any zero-shot triplet but can still relabel a consequent amount of missed true positive annotations. We display the complete algorithm in Algorithm 1. This algorithm takes as input ground truth pairs and predicate labels as well as predicted pairs and labels. It also takes as input triplet frequencies and predicate frequencies. The triplet

frequencies compute the number of times a triplet appears in the dataset, while the predicate frequencies compute the number of times a predicate appears in the dataset, from the overall number of triplets. By using both quantities we can compute the attraction factor *attr_score* which corresponds to the ratio of the triplet frequency over the predicate frequency, following Zhang et al. [95]. This factor is used to avoid transferring new predicates which are most likely to be annotation noise (i.e. low frequency in the dataset). The algorithm then selects the most relevant label from the confusion labels by ranking them with the attraction factor.

Algorithm 1 External Transfer for Data Augmentation

Inputs:
  Training images ($train\_imgs$)
  Ground truth annotations (gt_rels)
  Ground truth pairs with no predicate ($pairs\_no\_rel$)
  Triplet frequencies ($trip\_freq$)
  Predicate frequencies ($pred\_freq$)
Output: Updated relationship data (out_data)
for each img in $train\_imgs$ **do**
  $pred\_pairs$, $pred\_labels$ = model(img)
  for each $pair$ in $pairs\_no\_rel[img]$ **do**
    $pair\_idx \leftarrow \text{IoU}(pair, pred\_pairs)$
    $pred\_dist \leftarrow pred\_labels[pair\_idx]$
    if argmax(pd_dist) $\neq$ $background$ **then**
      $sorted\_dist \leftarrow \text{argsort}(pd\_dist)$
      $bg\_index \leftarrow \text{first\_similar\_index}(sorted\_dist, background)$
      $confusion\_labels \leftarrow sorted\_dist[: bg\_index]$
      $attr\_scores \leftarrow \emptyset$
      for each i, c_label in $confusion\_labels$ **do**
        $attr\_scores[i] \leftarrow trip\_freq[pair, c\_label]/pred\_freq[c\_label]$
      end for
      $sorted\_labels \leftarrow \text{sort}(confusion\_labels, attr\_scores)$
      $out\_data[pair] \leftarrow sorted\_labels[0]$
    end if
  end for
end for

Similar to the base dataset Visual Genome, the predicate distribution of our IndoorVG dataset is long-tailed with an Imbalance Degree of 28.71 and a log-likelihood ratio of 2.87. For instance, the class "on" solely counts for more than 41% of the annotated samples. This is a common issue in SGG datasets and is known to hinder the learning of SGG models. To address this issue, we propose to use the internal transfer method of Zhang et al. [95] to re-label existing annotations for more fine-grained predicates. The method goes as follows: for every ground truth annotated triplet, we select the new predicate from the top predictions that possesses the

highest attraction factor to replace the original predicate. The idea here is that if the attraction factor of the new predicate is higher than the ground truth one, then it is more likely to belong to the tail of the distribution. Using this method, a relation such as $\langle hand, of, man \rangle$ could be replaced by $\langle hand, belongingto, man \rangle$ which will help the model to learn more fine-grained relations. However, in their original implementation, Zhang et al. did not take into account relation categories and only focused on the predicate label, which could lead to the selection of wrong annotations. For instance, given the ground truth triplet $\langle arm, next\ to, man \rangle$ a more fine-grained predicate $\langle arm, belonging\ to, man \rangle$ could be selected by their method. Nonetheless, in this example, the meaning conveyed by the new relation would be different (a *part-whole* category) from the original one (a *topological* category) and most likely to be a false positive. A better option that will take into account the relation category here could be for instance $\langle arm, in\ front\ of, man \rangle$. A solution for this issue would be to simply not allow transfer between different relation categories. By a closer look, we observe that some relation categories can encompass others. For instance, the intended meaning of *functional* relations often indicate a *topological* proximity between subject and object. We can thus allow transfer between these two categories. However, we should not allow transfer between *part-whole* and *functional* categories for instance as a part-whole relation can not convey a functional meaning. Similarly, as we have seen before with the $\langle keyboard, on, laptop \rangle$ example, we should also allow transfer between *topological* and *part-whole* categories. We summarize the different new transfer rules based on relation categories dependence below:

- functional → functional, topological
- topological → topological, functional, part-whole
- part-whole → part-whole
- attributive → attributive

Given the introduced new transfer rules, we redefine the internal transfer method in Algorithm 2. This algorithm is similar to external transfer except that we loop over every ground truth pair that already has a predicate. We introduce the categories axioms such that the transfer to a more fine-grained predicate is only possible if the corresponding axiom is respected. It is important to notice here that in contrast to Zhang et al. [95], we authorize the transfer of all predicates. In their work, Zhang et al. do not transfer the top 30% most frequent predicates, even if their attraction factor is high. We believe that this is not necessary as the attraction factor already takes into account the frequency of the predicate in the dataset, also the proportion of 30% is arbitrary and could be different for other datasets (authors used VG150 in their work). Finally, in their work, Zhang et al. rank all possible transferred predicate labels by their attraction factor and then only transfer the top 70% of them. Again this proportion is arbitrary

and seems to improve the performance of models slightly compared to transferring all predicates. We believe that this is not necessary and that transferring all predicates is more beneficial in our case because we are working with a smaller dataset than VG150 and the number of possible transfers is already low.

Algorithm 2 Internal Transfer for Data Refinement

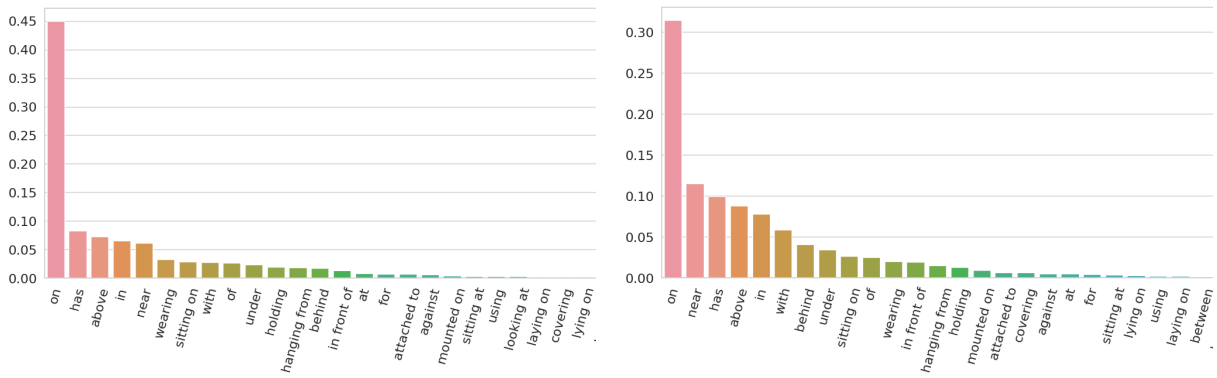
Inputs:
  Training images ($train\_imgs$)
  Ground truth labels (gt_label)
  Ground truth pairs ($pairs\_rel$)
  Triplet frequencies ($trip\_freq$)
  Predicate frequencies ($pred\_freq$)
  Triplet categories ($triplet\_cat$)
Output: Updated relationship data (out_data)
for each img in $train\_imgs$ **do**
  $pred\_pairs$, $pred\_labels$ = model(img)
  for each $pair$ in $pairs\_rel[img]$ **do**
    $pair\_idx \leftarrow$ IoU($pair, pred\_pairs$)
    $pred\_dist \leftarrow pred\_labels[pair\_idx]$
    if argmax(pd_dist) $\neq gt\_label[pair]$ **then**
      $sorted\_dist \leftarrow$ argsort(pd_dist)
      $gt\_label\_index \leftarrow$ first_similar_index($sorted\_dist, gt\_label[pair]$)
      $confusion\_labels \leftarrow sorted\_dist[: gt\_label\_index]$
      $attr\_scores \leftarrow \emptyset$
      for each i, c_label in $confusion\_labels$ **do**
        $attr\_scores[i] \leftarrow trip\_freq[pair, c\_label]/pred\_freq[c\_label]$
      end for
      $sorted\_labels \leftarrow$ sort($confusion\_labels, attr\_scores$)
      $gt\_cat \leftarrow triplet\_cat[pair, gt\_label[pair]]$
      for each c_label in $sorted\_labels$ **do**
        $c\_cat \leftarrow triplet\_cat[pair, c\_label]$
        if $c\_cat \in$ allowed_transfers(c_cat, gt_cat) **then**
          $out\_data[pair] \leftarrow c\_label$
          Break
        end if
      end for
    end if
  end for
end for

To evaluate our new data augmentation method, we ran experiments on IndoorVG with different SGG models. We chose the Transformers [19] and PE-NET [49] as a baseline. We first trained both models using the full IndoorVG dataset for 20 epochs with batch size 8 on one RTX A6000 GPU. We used the same hyperparameters and official implementation by the authors.

Models	Transfer	Int. rels.	Ext. rels.	F1@20/50/100	Improv.
Transformers [19]	None	-	-	12.79/16.89/19.00	-
	No Cat.	4,619	464	13.28/17.14/20.22	↑ **4.02%**
	With Cat.	4,485	464	13.21/17.40/20.49	↑ **4.97%**
PE-NET [49]	None	-	-	12.23/15.89/18.24	-
	No Cat.	8,167	12,980	13.57/17.07/19.68	↑ **8.54%**
	With Cat.	6,534	12,980	14.86/19.18/21.58	↑ **19.97%**

Table 3.8: Results of our new method for internal transfer on IndoorVG with different SGG models. **Improv.** is the relative improvement for the average of F1@20/50/100 for each method against the baseline with no augmentation (Transfer = None).

Then, we ran the two-step data augmentation on the training set of IndoorVG and used the unaltered test set for validation to create new data splits. Finally, we retrained each model for 10 epochs using the augmented training split. It is worth noticing here that re-training each model on the augmented training set generated by the other models is possible, however, to save time and training resources we focus on re-training each model on the augmented training set generated by the same model. In Table 3.8 we compare the performance of models trained on the dataset with normal internal transfer and our new method based on transfer rules with relations categories. Metrics used are the F1@K [95] at 20, 50, and 100 which correspond to the harmonic average of Recall@K and meanRecall@K for predicate classes. We used the same externally transferred data for all models to fairly compare the internal transfer only. The number of internal transferred relations drop by a quarter when introducing our transfer rules, from 8,167 to 6,534 for PE-NET and from 4,619 to 4,485 for Transformers. 12,980 new relations are added with external transfer using PE-NET, whereas this number drops to only 464 with Transformers. We can see that our new method improves the performance over the baseline by 4.97% and 19.97% for each model, where the improvement is only 4.02% and 8.54% with the previous method. This shows that SGG models are sensitive to the quality of the data and that our new method is effective in improving the performance of SGG models even with fewer relations transferred. Performance improvements are poor with the Transformers model because of the low number of relations transferred. We believe this comes from the fact that the Transformers model is overfitting a little to the data and will weigh the background class heavily in the predictions, making it difficult to obtain new candidates for internal transfer. Figure 3.9 represents the predicate distribution shift before and after internal and external transfer with the PE-NET model. We see clearly in this image that the distribution of predicates is more balanced after the transfer. We can see that the number of relations for the predicate "on" has decreased by 12% for instance. This more balanced distribution surely contributes to the improvement of the SGG models to detect more fine-grained relations. We decided to use the data split generated by the PE-NET model for the rest of the experiments as it is the largest one and has the best



(a) Distribution of predicates in IndoorVG after class selection and refinement.

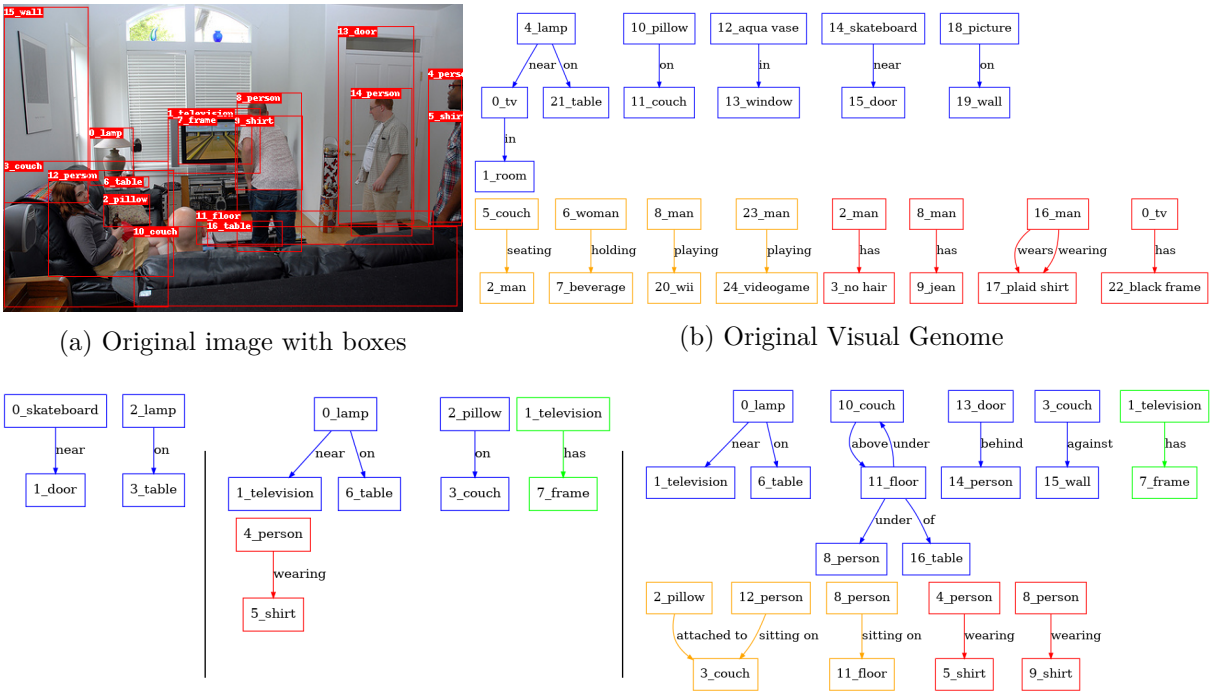
(b) Distribution of predicates in IndoorVG after transfer with the PE-NET model.

Figure 3.9: Comparative distribution of predicates in IndoorVG. For clarity, we only display the 25 most frequent predicates.

performance.

3.4.5 Comparison with Other Datasets

The final split of our IndoorVG dataset after data augmentation contains 112,804 annotations distributed over 14,733 images. Analytics of the dataset are available in Section A.3. It is interesting to compare this data split with other datasets that have been presented over the years for the task of SGG. In the Figure 3.10 we display an example of the difference between annotations in the original data and the IndoorVG dataset. Specifically, the original annotations from Visual Genome (Figure 3.10b) are noisy and contain a lot of useless information (such as $\langle man, has\ no, hair \rangle$). Regarding VG150 (Figure 3.10c), we see here a big problem with the data split, because classes have been chosen on a large set of different image contexts, the graphs in this split are generally very sparse and not representative of the image gist at all. By focusing on a specific context (indoor homes) and by selecting highly connected classes we can extract a much better representation, see Figure 3.10d. Finally, we can see that the annotations are more detailed and fine-grained in the IndoorVG dataset after the internal and external transfer, see Figure 3.10e. This shows that our method is effective in improving the quality of the data which did improve the performance of SGG models, as seen previously. This last point also interrogates the usage of the VG150 dataset as a benchmark for SGG models. Indeed, the VG150 dataset rarely represents comprehensive graphs about a scene and is more a collection of individual relations between objects, with a low connectivity. Issues arise because SGG models are designed to specifically model a graph structure and not to predict individual relations. This is why we believe that smaller but more qualitative data splits such as IndoorVG are more suited to benchmark models for the task of SGG.



(a) Original image with boxes

(b) Original Visual Genome

(c) VG150

(d) IndoorVG

(e) IndoorVG with transfer

Figure 3.10: Comparison between the annotations from the original data and our IndoorVG dataset, with and without transfer. Colors indicate the functional, part-whole, topological and attributive categories, respectively.

Dataset	$\|E\|$	$\|H\|$	$\bar{d}(V)$	D	Images	Obj.	Pred.
VG80K	19.03	3.43	2.35	0.34	104,832	53,304	29,086
VG150	6.98	1.93	2.02	0.29	89,168	150	50
IndoorVG	6.87	2.31	1.95	0.28	14,733	84	37
IndoorVG †	8.18	2.21	2.18	0.27	14,733	84	37

Table 3.9: Comparison of connectivity and number of images in the different datasets. †denotes the dataset after internal and external transfer with PE-NET.

Table 3.9 compares the connectivity of IndoorVG and VG150. We see a net improvement in both the average degree of nodes $\bar{d}(V)$ and overall graph size $|E|$ in IndoorVG compared to VG150. The average number of subgraphs $|H|$ also indicates a broader coverage of the image content with more different regions connected. We also computed the graph density $D = |E||V|(|V| - 1)$ which is the ratio of the number of edges over the number of possible edges in the graph. We see that the density of IndoorVG is lower than VG150, which is expected as the number of bounding boxes annotated per image is higher in IndoorVG because object classes are frequently present together (indoor context). Compared to all the data splits based on VG that have been proposed (see Table 3.1), IndoorVG is one of the most connected ones while at the same time being the only one to have semi-automatically curated object and predicate classes. This last point ensures that the dataset is more qualitative and easier to learn from as no classes are intersecting.

3.5 Concluding Remarks

In this chapter, we explored data biases in current SGG datasets, especially the Visual Genome dataset. We proposed a set of methods to alleviate these biases and improve the quality of the data. We first introduced a method to reduce the number of irrelevant relations using a new taxonomy based on the intended meaning of relations. Our findings showed that the current benchmark for SGG, VG150, is biased into invariant relations which hindered the performance of SGG models. In a new set of experiments, we demonstrated that SGG models are sensitive to *motifs* in the data (i.e. connected subregions of the image). By selecting classes based on connectivity rather than overall frequency, motifs are more present and the performance of models increases. Finally, we presented the IndoorVG dataset, a new dataset for the task of SGG in the domestic context. We first introduced a method to cluster indoor scenes from the Visual Genome dataset using sentence embeddings to extract a set of visually related images. We proposed to add to every annotation on this set of images an additional label representing one of the four categories: *part-whole*, *functional*, *topological*, or *attributive*. To do so, we used a pre-trained LLM to classify relations by categories in a few-shot manner. We then used data augmentation to enlarge the number of annotated relations between existing object pairs. We improved the original data augmentation method by limiting commonsense violations using our introduced relations categories. This approach has been thoroughly tested by running experiments on IndoorVG with different SGG models which validates our findings with improved overall performance.

This work is a step towards building more qualitative and representative datasets for the task of SGG, targeted for real-world applications domains. Our findings investigate the current usage of the VG150 dataset as a benchmark for SGG models. This dataset is highly biased and we demonstrated that smaller but more qualitative datasets can be more suited to benchmark mod-

els for the task. In fact, a data split from a defined model will encompass more inter-dependent relations which will be harder to predict than invariant relations that span over different contexts. By using such an approach, we can differentiate models that can learn complex interdependencies to form the global scene graph rather than only focusing on predicting invariant relations. Our new internal transfer method for data augmentation also shows the potential of using relations categories to filter incoherent or wrong predictions and improve the performance of SGG models. This last point could spark new interests at the crossroads of Knowledge Representation and Machine Learning, where knowledge priors could be used to improve the performance of models in real-world applications [111].

Constructing the IndoorVG dataset was a necessary step to leverage SGG in domestic service robotics. As a new step forward, we employed ourselves to use this dataset to train a model for the task of SGG. However, as we will see, SGG models are not tailored for out-of-the-box use in real-world applications. The inference process of these models is noisy and results in a high number of useless relations that need to be post-processed before being fed to a downstream module for reasoning. In the next chapter, we will present our work on post-processing the output of SGG models by introducing the notion of *Informativeness* of relations and how we can use it to filter noisy relations generated by an SGG model.

Chapter 4

Mining Informativeness in Scene Graphs

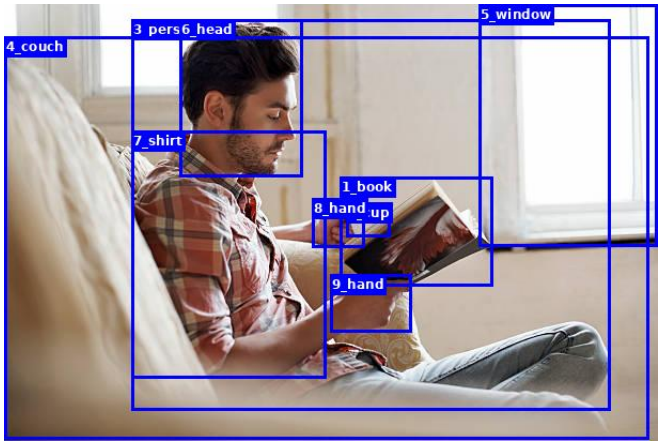
Information: the negative reciprocal value of probability.

Claude Shannon

> *Part of this chapter was published in the "Graph-based Representations for Pattern Recognition: New Results and Challenges" special issue of the Pattern Recognition Letters journal in 2025 [112].*

As described in the previous chapter, SGG datasets have many biases that hinder the performance of models in the task. Unfortunately, the models and especially the inference method of SGG models also possess biases. During inference, an SGG model typically computes relations between all pairs, giving a total number of $n * (n-1)$ predictions for n objects. Aside from the computational costs, this process generates a considerable amount of noise, as the main part of generated relations are either false positives or uninformative triplets (such as $\langle man, has, head \rangle$). When conducting inference on out-of-distribution images, the process of selecting a reasonable number of relations to construct the graph becomes challenging, especially when dealing with a high number of detected objects. If we want to use a SGG model as a backbone for reasoning and planning, we need to ensure that predictions are as informative as possible and, more importantly, that a minimum amount of noise is present in the graph. Here we define noise as valid but useless relations.

In the current paradigm of SGG, relations are predicted and then ranked given the confidence of the model. This process is not optimal, as the relations with the highest confidence values are not necessarily the most informative ones. In practice, we experienced the opposite effect where relations deemed trivial (such as $\langle person, has, hand \rangle$) are more likely to be predicted first by the model, as we have seen in Section 3.2. We display an example of this bias in Figure 4.1 where we can see that relations deemed "very informative" (e.g. $\langle person, on, couch \rangle$,



1. 5_window behind 9_hand
2. 6_head of 3_person
3. 3_person wearing 7_shirt

⋮

14. 3_person on 4_couch

⋮

27. 3_person reading 1_book

Figure 4.1: Example of relation predictions of the Motifs [38] model for an image from the VG150 dataset.

$\langle person, reading, book \rangle$) are still predicted by the model, but with low confidence and then a low ranking. When evaluating the performance of the model the ranking is not very important because current metrics measure the correctness of relations on a large sample of the predictions (typically between 20 and 100 samples). However, for some downstream tasks [14], [36], the ranking of relations is of utmost importance. We want to be able to easily select a very low amount of relations (for instance 5 to 10) and be sure that they are correctly representing the scene, both by diversity and informativeness. To do so, we need to define a novel metric that can measure the informativeness of a relation in a scene graph. This metric should be able to measure the amount of information that a relation brings to the understanding of the scene, both by itself and to other relations in the graph.

From a human standpoint, it is fairly simple to evaluate the quality of a graph by the amount of information that each relation provides with respect to (1) an image region and (2) the overall context of the scene, provided by associated relations. We give an example of this problem in Figure 4.2. Here we can see two graphs (A) and (B) that describe the left image. Those two graphs possess the same number of relations and approximately the same connectivity, however, we can see easily that graph (A) conveys more information about the scene than graph (B) as it contains more informative relations. Here we define informativeness as the quantity of information necessary to reconstruct the scene solely from the graph input. We state that a relation is highly informative if it helps significantly to describe the scene, following human judgment. In the example here, the relation $\langle laptop, on, woman \rangle$ is more informative than $\langle face, on, woman \rangle$ as a human agent is more likely to select the former than the latter to describe this scene where both relations hold. Differentiate between informative and not informative relations is for now outside the scope of SGG models. Indeed, current SGG models are trained with the only learning objective of generating correct relations between entities and

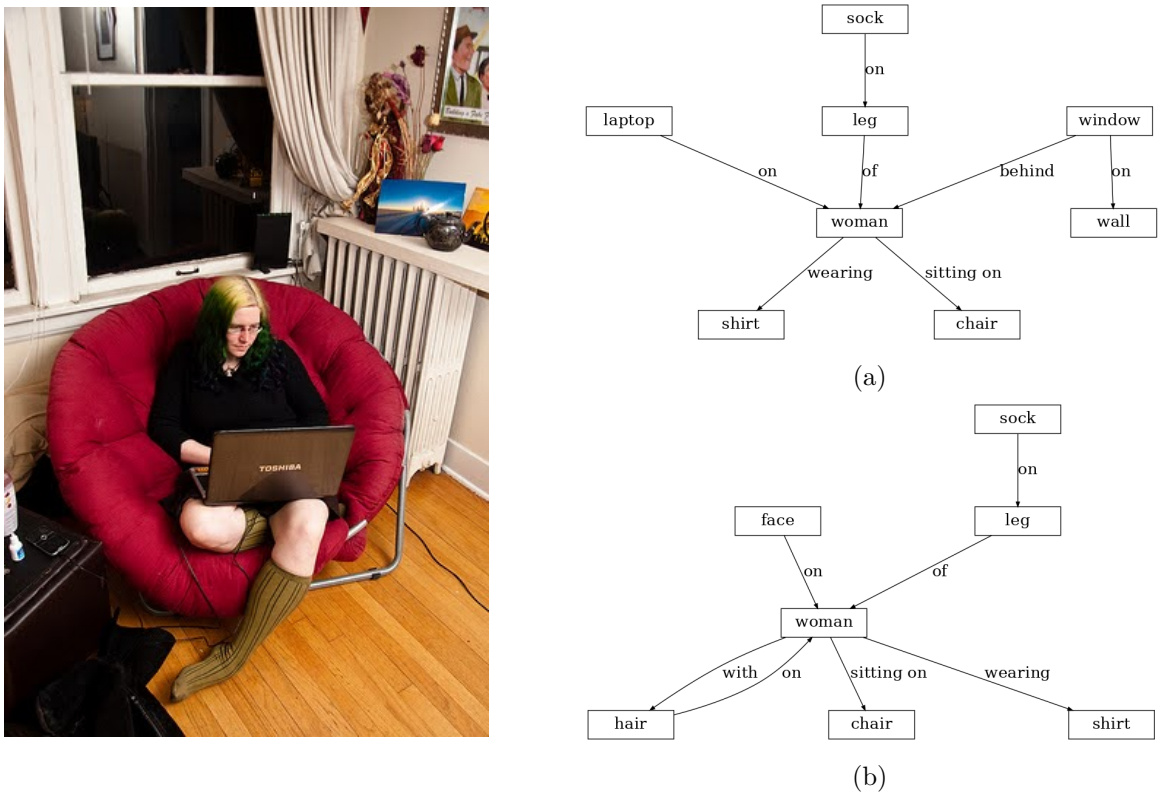


Figure 4.2: Example of two different scene graphs describing the same image.

not maximizing the overall information content of the graph. Thus, they tend to predict the most easy to guess relations, which are not necessarily the more informative ones in the current scene, as we saw previously with irrelevant relations (see Chapter 3). This is not a problem for the task in itself but is a major challenge for downstream tasks that rely on the information contained in the scene graph for planning or reasoning. Recent approaches [14], [25] highlight this issue for at least two downstream tasks, namely Visual Question Answering and Image Captioning. We strongly believe that this problem is not limited to those two tasks and that the usage of SGG in robotics could also benefit from more informative predictions. In fact, to support effective planning and reasoning algorithms, knowledge representations need to be comprehensive as possible [113].

Automatically measuring the informativeness of a relation in a scene graph is a challenging task. In Section 4.1 we review the current metrics used in SGG, their advantages and flows and highlight the challenges to measure informativeness of relations in scene graphs. Measuring informativeness is not straightforward because a relation is not only informative by itself but also for other relations in the graph. We define the independent information of a relation as its *intrinsic* information value and the information it brings to other relations in the graph by its

connectedness as its *extrinsic* information value. A correct definition of informativeness should then encompass both the *intrinsic* information that the relation brings to the understanding of the scene and its *extrinsic* importance on the graph structure. Regarding *intrinsic* information, due to the annotation biases and noise contained in scene graph datasets, it is difficult to measure the informativeness of a relation solely based on its frequency in the dataset. We then propose to use external data to measure *intrinsic* information based on scene descriptions in Section 4.2. On the other hand, *extrinsic* information given by a relation can be measured by looking at the topology of the graph, see Section 4.3. We then propose to use the *intrinsic* information to assert the performance of SGG models in producing informative graphs with the introduction of the InformativeRecall@K metric Section 4.4. By leveraging the *intrinsic* and *extrinsic* information measures, we can help SGG models to generate more informative scene graphs during inference, see Section 4.5. To prove this last point, we evaluated our approach in a set of three different downstream tasks and show that our method can improve the performance of SGG models in real-world settings, see Section 4.6.

4.1 Measures of Scene Graph Quality

SGG models traditionally output the top predicate probability $p \in P$ for each possible pair of n objects in the image, such as $|P| = n(n-1)$. Considering that the object detector can detect more than a hundred objects in an image, the number of possible relations to predict can be very high in comparison to the number of ground truth annotated relations. This is mainly because a lot of true positive annotations are missing from SGG datasets, as the task of annotating all $n(n-1)$ relations will be extremely intensive. To address this issue, the Recall@K (R@K) metric was introduced in the SGG literature [77] to evaluate the quality of the predicted relations. This metric computes the number of times a ground truth relation is detected in the top K predicted relations ranked by confidence, as follows:

$$\text{Recall@K} = \frac{1}{N} \sum_{i=1}^{N} \frac{|P_{0 \to K}(i) \cap \text{GT}(i)|}{|\text{GT}(i)|}, \tag{4.1}$$

where $P_{0 \to K}(i)$ is the set of predicted relations for the i-th object pair in the top K relations, and $\text{GT}(i)$ is the set of ground truth relations for the same object pair, matched by the IoU (IoU) of the bounding boxes (threshold $>= 0.5$) [77]. However, due to the heavy long-tail distribution of predicates in SGG datasets, it is fairly simple to obtain a high Recall@K by only predicting a handful of relations that are very frequent in the dataset [37]. This is why the meanRecall@K (mR@K) metric has been proposed [37], [114], which averages the recall of the top K relations

for each predicate class individually:

$$\text{meanRecall@K} = \frac{1}{|P|} \sum_{p \in P} \text{Recall@K}(p) \tag{4.2}$$

This way, the performance of the model on the least frequent classes is given the same importance as the performance on the most frequent classes, which is more representative of the overall quality of the model. These two metrics are usually computed for values of $K \in [20, 50, 100]$.

To easily measure the trade-off between Recall@K and meanRecall@K, the F1@K (or simply F@K) metric has been recently proposed [95]. F1@K is the harmonic average between Recall@K and meanRecall@K, as follows:

$$\text{F1@K} = 2 \times \frac{\text{R@K} \times \text{mR@K}}{\text{R@K} + \text{mR@K}}, \tag{4.3}$$

where R@K and mR@K are the Recall@K and meanRecall@K metrics, respectively. This metric is particularly useful to evaluate the quality of a model for real-world applications where the trade-off between the performance on the most frequent classes (R@K) and a balance evaluation of all classes (mR@K) is important.

Recall@K, meanRecall@K, and F1@K are the standard metrics in the SGG literature to evaluate the quality of a model. These metrics are hard-matching metrics at the relation level, neglecting the graph structure in the evaluation. To tackle this issue, Tang et al. [19] proposed to evaluate the quality of a model on the downstream task of Sentence-to-Graph Retrieval (S2GR). This task uses textual descriptions of scenes (i.e. image captions) to extract Textual Scene Graphs (TSG) and match them with the generated Visual Scene Graphs (VSG) for the same image. The objective is, given a detected VSG to be able to retrieve from a gallery of TSG the correct one which corresponds to the same image. To match the vocabulary of the VSG and TSG, the authors used a Bilinear Attention Network [115] to encode both graphs in the same embedding space. In the original implementation, a gallery size of 1,000 and 5,000 images for retrieval is used. The performance is measured using Recall@K and the median ranking index of retrieved results (Med). This approach is indeed interesting, however, it does not solve all the challenges introduced above. First, the process does not evaluate the quality of the generated graphs directly but rather the performance of a trained model (the Bilinear Attention Network) with the generated graphs as inputs, which can introduce further biases. Second, the performance of the model is evaluated on a small gallery of TSG for computational reasons, which may not be efficient given the sparsity of different contexts in the dataset. For instance, a graph representing a scene that occurs a lot will be harder to correctly match as a lot of images can be very similar, on the other hand, a graph representing a scene that occurs only once will be easier to match as it is unique. To our knowledge, this approach has not been used in the SGG literature outside the original paper. In the following, we take as inspiration the approach employed by Tang et al.

with the TSG extracted from captions to provide a new method of measuring the quality of scene graphs. In the next sections, we present a new method which is simpler and more efficient than the S2GR task, and can be used to evaluate the quality of a model directly. By contrast to the S2GR task, our method does not need any further training or fine-tuning, which reduces biases and computational costs. Our method also does not require a large gallery of TSG for evaluation, which makes it more efficient for large datasets. This method can also be used to evaluate the quality of generated graphs on out-of-distribution images, further boosting the performance of SGG models with no re-training. This method is based on *intrinsic* and *extrinsic* information of relations in scene graphs.

4.2 Intrinsic Information

As introduced before, the informativeness of scene graphs lies in the way we humans choose a relation instead of another when asked to describe major elements in an image. Based on this definition, a coherent choice would be to compare relations contained in human-annotated image captions to relations contained in scene graphs. The hint here is that captions can be seen as representations of the *gist* [33] which contains the "true meaning" of the scene. By measuring the alignment of each relation in the graph with relations contained in the corresponding caption, we can measure the semantic importance of each relation in a dataset. This defines a practical implementation of the *intrinsic information* value of a relation in a scene graph. The following section describes our implementation for measuring this information value based on textual similarity between scene graphs and image captions. In the following, we will use the VG150 dataset for evaluation to demonstrate that our method can be applied to a large dataset with diverse contexts, however, our findings are directly applicable to the IndoorVG dataset or any other SGG dataset.

4.2.1 Textual Scene Graphs

The Visual Genome (VG) dataset [8] (and by extension VG150 and IndoorVG) does not contain image description annotations. However, part of the images in VG is also present in the COCO dataset [45], which contains 5 image captions per image. Only half of the 108,077 images of VG intersect with COCO, which leaves us with 56,575 images with no captions. To overcome this issue, we used the pre-trained BLIP-2 Vision-to-Language model [116] to generate textual descriptions for the remaining images in a zero-shot manner. We used the BLIP-2 ViT-g $\text{OPT}_{2.7\text{B}}$ model [116] fine-tuned on the COCO-captions dataset, which ensures consistent data across the part of the new annotations which has been manually annotated and automatically generated. We generated 5 captions for each remaining image to fit the format of the original COCO captions. To ensure a diversity of generation, we used different prompt styles and temperature

settings for each caption, more details in the Section B.1. Figure 4.3 shows an example of 5 generated descriptions for an image from the Visual Genome dataset. We can see that generated captions are very reasonable but also diverse, which is a good sign for matching relations contained in the scene graph with those contained in the textual descriptions. Due to the different nature of free-form texts and graphs, direct matching is impossible. Inspired by Tang et al. [19], we employed ourselves to convert the captions into TSGs to be able to compare them with VSGs from Visual Genome.

1. A man sits at a desk surrounded by multiple computers
2. A guy sitting at a desk in a room with a computer
3. A room with two computers and a man in a jacket
4. A man working on two computer screens in a large office
5. A man sitting at a desk with two computers on it

Figure 4.3: Example of 5 generated captions for an image from the VG150 dataset.

Extracting relations from human-made image captions is not a straightforward process. It encompasses various challenges of Natural Language Processing (NLP) such as quantifier scoping, pronoun resolution, or dependency parsing. The method used by Tang et al. [19] for the task of Sentence-to-Graph Retrieval is based on the Stanford NLP Parser [117] which has been criticized recently for its lack of robustness [118], [119]. The Stanford NLP Parser is purely based on heuristic rules, which are not designed to handle complex sentence. An example can be a sentence with multiple quantifiers such as "three men reading books" [118], which will be parsed distributively resulting in three relations with the same book, which is not realistic. To overcome this issue, we propose to use a pre-trained language model to generate TSGs from the captions. We used the Flan-T5-large model trained on the FACTUAL dataset [118] which has better accuracy and can handle more complex sentences than the Stanford NLP Parser. We extract one graph for each of the 5 captions which are then merged together by removing duplicates to form the final TSG. Furthermore, we added a step of filtering to remove relations that contains a quantifier (such as $\langle person, are, three \rangle$) or adjectives (such as $\langle living\ room, is, large \rangle$) using the Spacy [120] dependency parser to keep only relations of the same type as the ones in the VSGs. We display an example of a TSG generated from the captions in Figure 4.4. In this image, we can observe that the node *man* has been decoupled with the node *person*, which is likely due to the change of vocabulary between captions number 1,3,4,5 and caption 2, see Figure 4.3. Similarly, the object *computer* mentioned in caption 2 cannot be associated with one of the

computers mentioned in the other captions, resulting in two different nodes in the graph. These issues are common in generated TSG and are due to the lack of grounding of the relations to the image because the model is text-based only. We argue that this is not a major problem for our approach because relations that are similar in meaning (such as $\langle room, with, computers \rangle$ and $\langle room, with, computer \rangle$ in this example) should be matched with the same unique relation in the VSG if it exists.

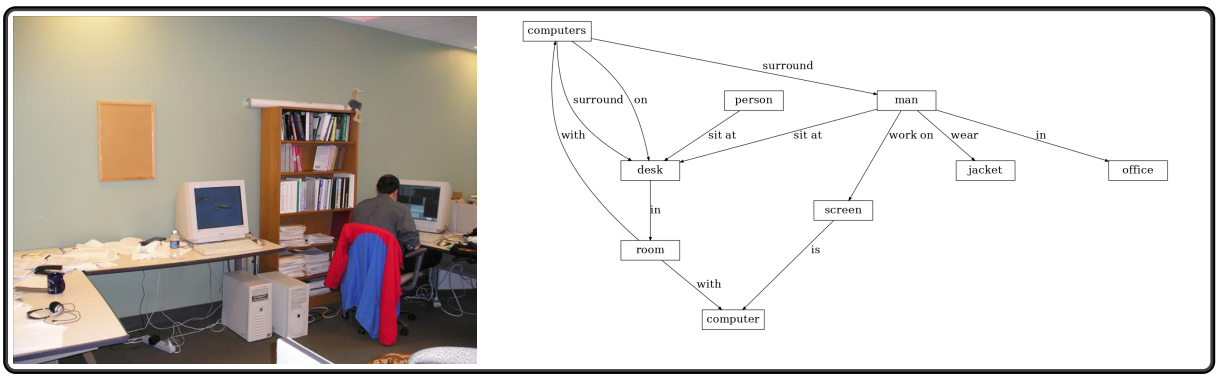
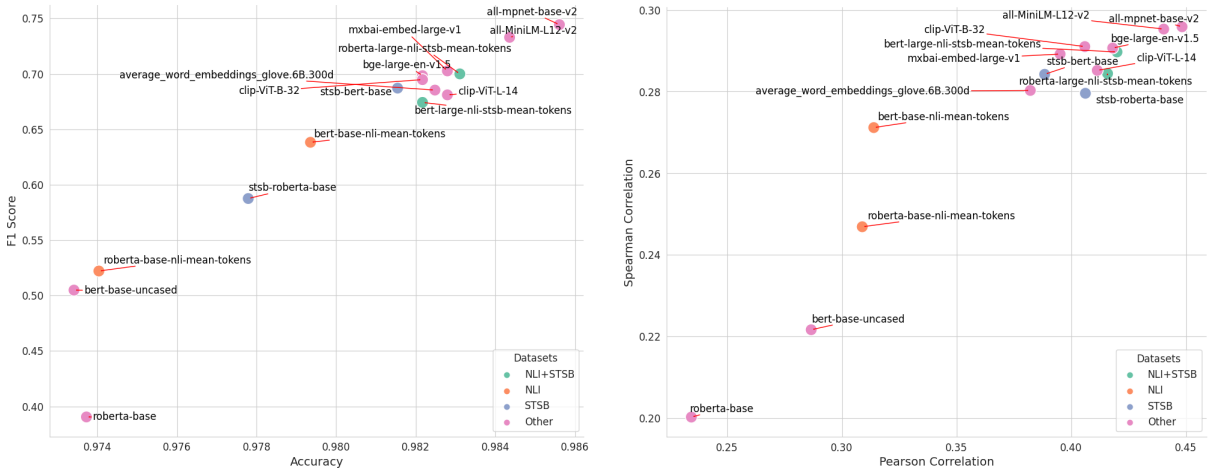


Figure 4.4: Example of TSG extracted from the set of captions in Figure 4.3 using the Flan-T5-large model [118] and our custom post-process function.

4.2.2 Semantic Matching

To perform the matching between triplets in the TSG and VSG based on their inherent meaning, a direct comparison cannot be applied because of the distinct vocabularies and decoupled nodes [19]. In addition, we want to be able to count as a match inverse relation such as $\langle person, holds, phone \rangle$ and $\langle phone, held\ by, person \rangle$ which is a more difficult problem. To solve the first problem, we propose to use word embeddings. However, a one-to-one matching of embeddings of the subject, predicate, and object would not solve the inverse relation problem, as the order will still matter. To solve this issue, we propose to use sentence embeddings on the whole $\langle subject, predicate, object \rangle$ triplet to match the TSG and VSG relations. Sentence embeddings trained for Semantic Textual Similarity (STS), can easily pair relations with different subjects and objects but similar meanings such as $\langle person, holds, phone \rangle$ and $\langle phone, held\ by, person \rangle$. However, more complex cases such as $\langle cup, on\ top\ of, table \rangle$ and $\langle table, below, cup \rangle$ are still challenging to match. This type of pairing is related to the task of Natural Language Inference (NLI) which aims to decide if a sentence entails, contradicts, or is neutral with respect to another sentence. In our example here with the cup and the table, the first relation entails the second one, which is different from the similarity depicted with our first example with the phone (which is the kind of similarity measured by STS). To solve this problem, we propose to use pre-trained sentence embedding models specifically fine-tuned for the task of NLI and STS [99].



(a) F1-score versus Accuracy for different models on our collected dataset (distance metric = best cosine).

(b) Pearson versus Spearman correlation for different models on our collected dataset (distance metric = best cosine).

Figure 4.5: Measuring Semantic Similarity between TSG and VSG relations using different sentence embedding models [99].

To benchmark existing models, we introduce the task of *Visual Relation Semantic Similarity* which aims at measuring both the direct similarity and entailment in relations contained in visual scenes. As an evaluation dataset for this task, we manually annotated 4,000 triplet pairs randomly sampled from the IndoorVG annotations with corresponding TSG annotations obtained from COCO captions. For every pair, we annotated the similarity as a binary value (0 for no similarity, 1 for similarity). Here, the value 1 can signify either a direct similarity or an entailment, as explained before. We tested a range of 13 different sentence embedding models, fine-tuned on either NLI datasets (SNLI and MultiNLI [121]) or STS datasets (STS-Benchmark [122]). We mainly benchmarked BERT [110] and ROBERTA [123] variants, as well as some more recent approaches targeting sentence similarity tasks [124], [125]. Furthermore, we also tested the CLIP text encoder [126] for the task. The hint here is that CLIP has been trained on a large dataset of images-text pairs, and should be able to capture the semantics of relations in visual scenes better than models trained on text-only datasets.

We evaluate the task of *Visual Relation Semantic Similarity* by computing the cosine similarity between every pair of relations and assign the prediction 1 for a similarity obtained above a certain α threshold, 0 otherwise. We then compute the precision, recall and F1-score of the predictions with respect to the ground truth annotations. To pick the best α threshold fairly, we ran experiments for all models with ranging values from 0.1 to 0.9, with a step of 0.05 and selected the best threshold for each model based on the F1-score. In Figure 4.5 we display the results of our experiments. Traditionally, the performance of sentence embeddings is measured using the

Spearman and Pearson correlation factor with ground truth annotation (human evaluation) as well as standard accuracy. In addition, we display the F1-score which is a more robust metric in our case as we also do not want to mislabel any false positive. We can see that the models fine-tuned on STS datasets perform better than the models fine-tuned on NLI datasets. However, models fine-tuned on both NLI and STS perform better than models fine-tuned on STS only. This is likely because NLI models are trained to capture entailment but also contradiction and neutrality, which are not present in our dataset. We can see that the CLIP text encoder performs better than most specialized models, which shows the versatility and robustness of the model to capture the semantics of relations in visual scenes. Finally, we observe overall supremacy of newer models trained on a large set of tasks (such as all-mpnet-base-v2 or all-minilm-L12-v2) which are usually trained on a set of 32 or more datasets containing NLI and STS tasks [1]. We can see that the best model is *all-mpnet-base-v2* (best F1-score obtained with $\alpha = 0.77$) which is a variant of the MPNet model [124] fine-tuned on a large set of tasks. This model is likely better at capturing the semantics of relations in visual scenes because it has been trained on a more diverse set of tasks. We will use this model with the $\alpha = 0.77$ in the following to measure the *intrinsic information* value of relations in scene graphs.

4.2.3 Relation Ranking

To find the overall similarity of each relation across the IndoorVG dataset, we propose to use the cosine similarity distance between sentence embeddings of the VSG and TSG computed by MPNet. For every image, we compute the cosine distance between embeddings of every VSG relation with all TSG relations and keep only the top-1 distance as a match. By averaging cosine distances across images for every VSG triplet, we built a ranking of the likelihood of a relation in a generated VSG to be present in the corresponding caption, which in fact corresponds to the likelihood of belonging to the image gist. This gave us the *intrinsic information* value of relation in Scene Graphs. Table 4.1 shows a few examples of relations ranked from highly relevant (Top-5) to completely irrelevant (Bottom-5) using the *intrinsic information* value computed on the Visual Genome images and averaged. We can see that the Top-5 describes mostly spatial relations between what we can expect to be important elements of the scene, whereas the Bottom-5 describes relations related to attributes of entities that are *a priori* commonsense knowledge [66] (e.g. $\langle wheel, on, car \rangle$) or which seems to be very minor details of the scene (e.g. $\langle fork, behind, woman \rangle$). All top-5 relations have a cosine similarity score of 1 (exact matching in every image they appear in) and all bottom-5 relations have a cosine similarity score of 0 (no matching in any image they appear in). This can sometimes be misleading as some relations only appear in a few images.

[1] For more information on the models or training data, please consult `https://sbert.net/docs/sentence_transformer/pretrained_models.html`, *accessed on the 17/08/2024*.

Top-5	Bottom-5
bowl on head	wheel on car
woman in front of giraffe	fork behind woman
umbrella above dog	plant covered in plant
person on airplane	leaf over bear
beach near mountain	men watching person

Table 4.1: Top-5 and Bottom-5 relation after ranking all triplets in Visual Genome according to our definition of *intrinsic information* value.

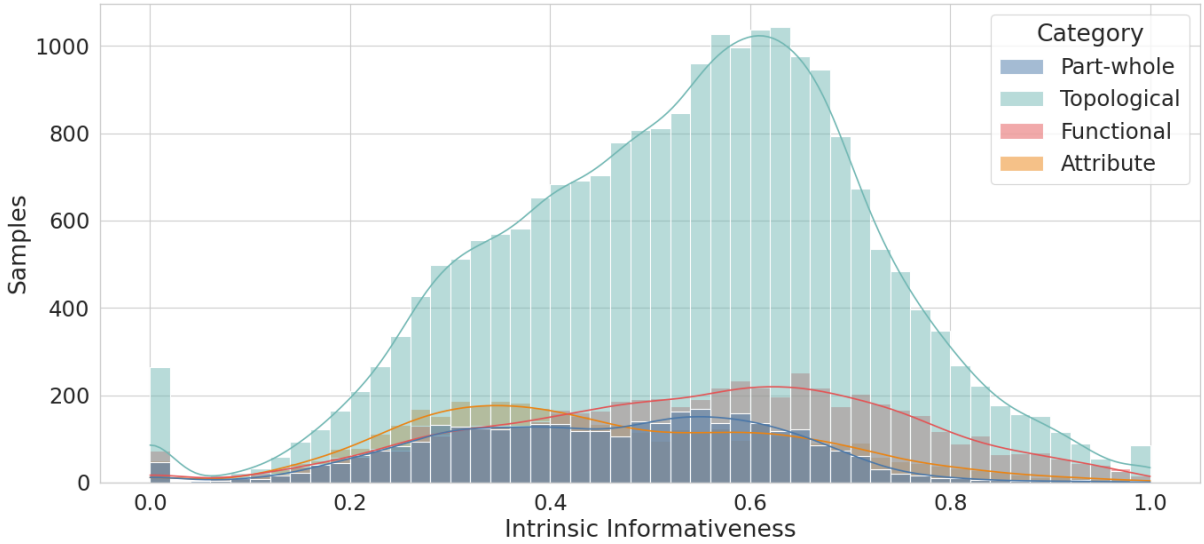


Figure 4.6: Distribution of relations based on their *intrinsic information* value and semantic category.

For a relation represented as an edge $e = \langle s, p, o \rangle$ in a scene graph, we define the *intrinsic information* value of the edge $\zeta(e)$ as a statistical prior between the reference graphs distribution ($\sum_{i=1}^{N} TSG$) and the target distribution ($\sum_{i=1}^{N} VSG$), where N is the number of images in a given dataset. Using cosine similarity as a distance metric, we have:

$$\zeta(e) = \frac{1}{N} \sum_{i=1}^{N} \text{cosine\_similarity}(e \in TSG_i, VSG_i) \tag{4.4}$$

After ranking relations based on *intrinsic* information value, it is interesting to look at the types of relations deemed more informative than others, with the paradigm of relations categories introduced in the previous chapter. We analyze the distribution of relations based on their *intrinsic information* value and semantic type in Figure 4.6. First, we observe that the distribution is not a normal distribution, with a significant number of outliers at 0 (no similarity at all) and 1 (complete similarity for all samples). Regarding the distribution per category, we

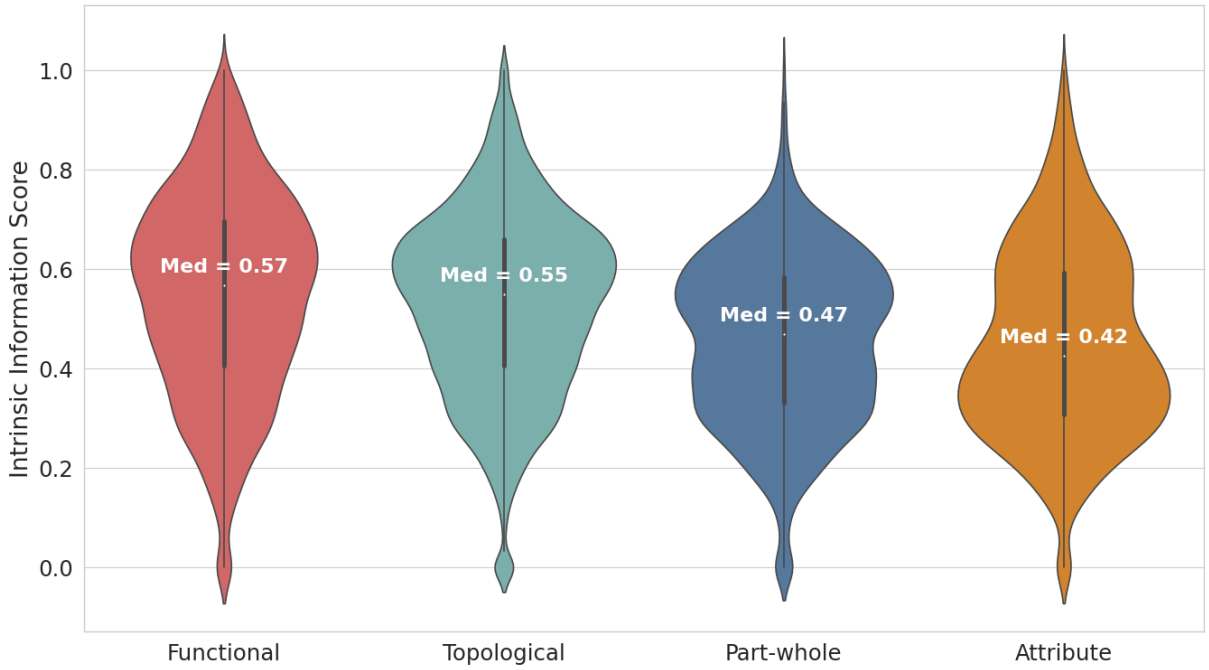


Figure 4.7: Distribution of relations based on their *intrinsic* information value and semantic type as a violin plot, Med is the respective median value for each category.

can see that all categories are distributed all over the spectrum of *intrinsic information* values, with a slight tendency for the Functional and Topological categories to have higher values than the Attributive and Physical Part-whole categories. In the Figure 4.7 plot we can observe this disparity more clearly. We see a significant difference in the distribution but also median and mean values of the Topological and Functional categories compared to the Attributive and Physical Part-whole categories. Specifically, the median values for the Functional and Topological categories are 0.57 and 0.55 respectively, whereas the median values for the Attributive and Physical Part-whole categories are 0.47 and 0.42 respectively, which is a relative gap of 20% between the two groups.

To verify the statistical significance of these results, we performed a Kruskal-Wallis H test [127] on the distribution of the *intrinsic* information value of relations based on their category. Here we used the Kruskal-Wallis H test as the data was not normally distributed and the sample sizes were different between each group. Our assumption is that the four categories are not equally represented in the dataset, thus not a single distribution can be fitted to the data. The Kruskal-Wallis H test showed that there was indeed a statistically significant difference between informativeness score and relation categories with $p_{value} = 0.03$, with $p_{value} < 0.05$ for significance. Then, we performed a post-hoc Tukey-Kramer test [128] to see which categories were significantly different from each other. The results are shown in Table 4.2, with Lower and

Pairs		Mean Diff	p-adj	Lower	Upper	Reject H_0
attribute	functional	0.0981	0.0000	0.0883	0.1079	True
attribute	part-whole	0.0066	0.4162	-0.0045	0.0178	False
attribute	topological	0.0817	0.0000	0.0736	0.0899	True
functional	part-whole	-0.0914	0.0000	-0.1018	-0.0811	True
functional	topological	-0.0163	0.0000	-0.0234	-0.0093	True
part-whole	topological	0.0751	0.0000	0.0663	0.0839	True

Table 4.2: Tuckey-Kramer test for the *intrinsic* information value of relations based on their category.

Upper being the 95% confidence interval bounds. We can see that all categories are significantly different from each other except for Attributive and Part-Whole, as the adjusted p-value (p-adj) is above 0.05 for this particular case. This is consistent with the fact that the Attributive and Part-Whole categories are very similar in semantics, as they both convey the belonging, more or less strictly, of the object to the subject entity. We can also confirm these results by looking at Figure 4.6 and Figure 4.7 where we can see that the Attribute and Part-Whole categories have very similar distribution and mean values.

The conclusion from this study is that, when describing scenes, Topological and Functional relations are the ones that convey the most information. However, Attribute and Part-Whole relation types are not completely useless as they are still informative, with an average *intrinsic information* value at 0.45 (again, here a value of 0 corresponds to no importance at all, which means that the relation could be discarded without changing the representation of the "true meaning" of the scene). It is also important to notice here that these numbers are computed using *intrinsic information* only and do not take into account the *extrinsic information* value of relations from their interplay in the graph structure. In real-world examples, taking this information into account could lead to different results.

4.3 Extrinsic Information

At the graph level, gauging informativeness becomes inherently intricate. A relation that might be considered "not informative" within a specific context can transition to being informative in another scenario, particularly when other relations rely on it. Here we draw a specific example given the relation $\langle man, has, hand \rangle$ in two different contexts:

$$man \xrightarrow{\text{has}} hand \xrightarrow{\text{has}} finger \tag{4.5}$$

$$man \xrightarrow{\text{has}} hand \xrightarrow{\text{has}} book \tag{4.6}$$

While the structure of those two graphs is very similar, the information conveyed by the relation $\langle man, has, hand \rangle$ is very different. In the graph represented in Equation (4.5), the relation is descriptive of the entity "man" and gives no contextual information whereas Equation (4.6) characterizes a functional relation between the entity "man" and "book" that could be understood as "man *is holding* book *with* hand". Thus, we define the *extrinsic information* of a relation through its importance in the graph structure with respect to other relation values. The key idea here is that relations that, when removed, are likely to disrupt the flow of information in the graph should be considered more informative than others. To measure the *extrinsic information* of a relation, we then need to look at the topology of scene graphs and, specifically, edge importance measures.

There exists multiple algorithms to measure the importance of an edge e in a graph, a simple method could be to average the degree (i.e. number of connections) of the input node v_1 and output node v_2 of the edge. This method is called the *degree centrality*. For a directed graph, we can take into account the number of ingoing and outgoing edges from v_1 and v_2, respectively, such that:

$$c_D(e) = \frac{d_{in}(v_1) + d_{out}(v_2)}{2} \tag{4.7}$$

The PageRank algorithm [129] can also rank the importance of nodes in a graph by their respective connectivity. PageRank is defined as the probability of a random walker being at a given node (or webpage in the initial implementation [130]) at a given time. The PageRank of a node v is defined as follows:

$$PR(v) = \frac{1-d}{N} + d \sum_{u \in B_v} \frac{PR(u)}{d_{out}(u)}, \tag{4.8}$$

where d is the damping factor, typically set to 0.85, to model the probability that a user will continue clicking on links (or the probability of moving from one node to another). The term N represents the total number of nodes. The summation iterates over the set B_v, which consists of all pages that have links (or edges) to page v, and $\frac{PR(u)}{d_{out}(u)}$ calculates the PageRank contribution from a page u to page v, with $d_{out}(u)$ being the number of outgoing links from page u. PageRank only computes values for nodes, to compute values for edges, we average the PageRank of the input and output nodes.

The average degree and PageRank algorithm are satisfying ways of measuring the importance of edges in a graph. However, none of them are specifically taking into account the connection between highly connected subgraphs (i.e. motifs) which we believe is essential to measure the importance of a relation in a scene graph. For a scene graph, *motifs* can be referred to as communities in graph theory. A community is a set of nodes that are more connected to each other than to the rest of the graph. Finding connections between communities is likely important to extract key relations from the graph structure because if we remove a relation that connects

two communities, we are likely to remove important contextual information about the scene. In 2002, Girvan and Newman [131] introduced the concept of edge betweenness centrality, which is an extension of node betweenness centrality proposed by Freeman [132] to measure connectivity between communities in a graph. The definition goes as follows: for each edge $e \in E$ of a graph $G = (V, E)$, its edge betweenness is represented as the sum of all shortest paths $\sigma(s, t|e)$ that passes through e over all possible shortest paths $\sigma(s, t)$, see Equation (4.9):

$$c_B(e) = \sum_{s,t \in V} \frac{\sigma(s, t|e)}{\sigma(s, t)} \tag{4.9}$$

Thus, we propose to extend the definition of edge betweenness centrality to Scene Graphs as follows: given a weighted graph $G = (V, E, w)$, we define the *extrinsic information* of a relation as the sum of all shortest paths that pass through the relation e over all possible paths in the graph. To take into account the *intrinsic information* of the relation, noted $V_i(e)$, we add a distance value to every edge in the graph $w(E) = 1 - V_i(e)$ such that the shortest path $\sigma(s, t)$ is computed as follows:

$$\sigma(s, t) = \min_{p \in P(s,t)} \sum_{e \in p} w(e) \tag{4.10}$$

The shortest paths are computed using Dijkstra's algorithm. Finally, values are normalized by the maximum number of paths, i.e. $\frac{1}{(n(n-1))}$ where n is the number of nodes in G. This gives us, for every relation, its *extrinsic information* value $\varepsilon(e)$, highlighting the information flow of Scene Graphs. By combining *intrinsic* and *extrinsic* information, we define the overall *information score* of a relation as follows:

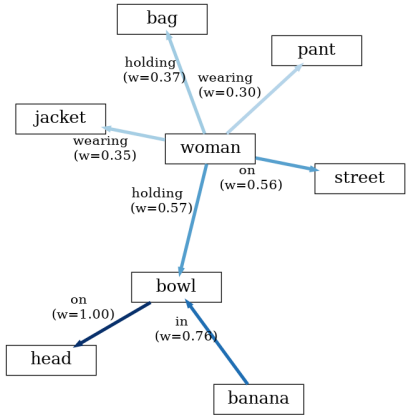
$$\tau(e) = \frac{\zeta(e) + \varepsilon(e)}{2}, \tag{4.11}$$

which efficiently takes into account the trade-off between the intrinsic importance of relations and their structural extrinsic importance in the graph.

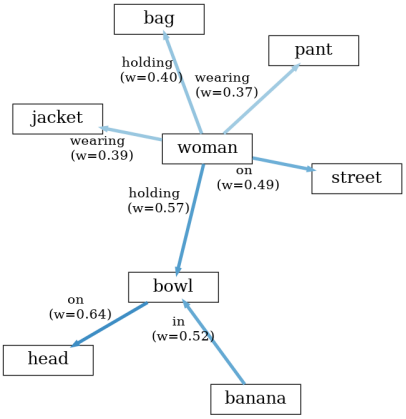
We display an example image and the corresponding graphs in Figure 4.8. We compare our approach with the baseline of *intrinsic information* value only, using average degree and PageRank as edge importance measures. First, we can see that the relation $\langle bowl, on, head \rangle$ is indeed the most informative relation by *intrinsic* score (see Figure 4.8b) because it gives contextual information about the action being performed, which is almost certain to appear in the corresponding caption. We can also see that the important relation about the context of the scene (e.g. $\langle woman, on, street \rangle$) is highly weighted ($w > 0.5$) and that other semantically important relations, related to activities or spatial relations, are weighted even more (e.g. $\langle woman, holding, bowl \rangle$, $\langle bowl, on, head \rangle$). Finally, the rest of the relations with lower weights give more details about the scene but are not important to understand the *image gist*

(e.g. $\langle woman, wearing, jacket\rangle$, $\langle woman, wearing, pant\rangle$). When we compare the baseline Figure 4.8b with the re-weighted graphs using the degree centrality Figure 4.8c, we observe that non-informative relations such that $\langle woman, wearing, jacket\rangle$ have increased in weight, which is not desirable. Regarding the PageRank centrality Figure 4.8d, we can observe that the relations $\langle woman, holding, bowl\rangle$ and $\langle woman, on, street\rangle$ have almost the same weight even though their respective importance in the graph structure is very different, which is also not desirable. Finally, when we use the edge betweenness centrality for edge importance measure Figure 4.8e, we can see that all relations are more fairly weighted. If we compare with the baseline Figure 4.8b, we can see that the relation $\langle woman, holding, bowl\rangle$ is now much closer to the very informative relations (e.g. $\langle bowl, on, head\rangle$ and $\langle banana, in, bowl\rangle$) and far from the other relations. This is consistent with the fact that this relation is very important for the flow of information in the graph and thus should be re-weighted accordingly.

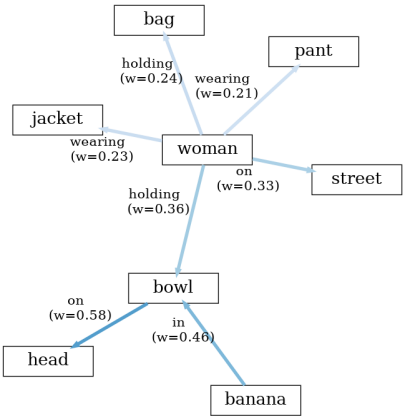
(a) Original image



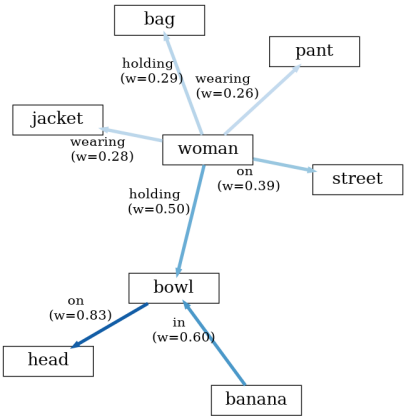
(b) Baseline: *Intrinsic Information* score $\zeta(e)$ only



(c) *Informative* score $\tau(e)$ with $\varepsilon(e) = Degree$



(d) *Informative* score $\tau(e)$ with $\varepsilon(e) = PageRank$



(e) *Informative* score $\tau(e)$ with $\varepsilon(e) = Betweenness$

Figure 4.8: Comparison of different edge importance measures for the same image. The color of edges gives the weight of the relative weight of the relation $\in [0; 1]$ for better visualization.

4.4 Informative Recall @ K

Algorithm 3 Informative Recall@K

Input: Predicted Scene Graph G_{pred}, TSG G_{text}, α, K
Output: Informative Recall@K
$IR@K = 0$
for each $\langle s, p, o \rangle \in G_{pred}$ **do**
    $sum \leftarrow 0$
    for each $\langle s', p', o' \rangle \in G_{text}$ **do**
        if cosine_similarity($\langle s, p, o \rangle$, $\langle s', p', o' \rangle$) $\geq \alpha$ **then**
            $match\_idx \leftarrow$ index($\langle s, p, o \rangle, G_{pred}$)
            $sum \leftarrow (K - match\_idx)/K$
            break
        end if
    end for
end for
$IR@K \leftarrow \frac{sum}{|G_{text}|}$
return $IR@K$

As we have seen in Section 4.1, the current metrics used in the SGG literature are not sufficient to evaluate the quality of a model in real-world application settings. The Recall@K and meanRecall@K do not take into account the graph structure and the semantic importance of relations. To solve this problem, we define a new *Informative* metric, InformativeRecall@K (IR@K), for the task of SGG, based on our definition of the *intrinsic* information value. This metric computes the number of times a predicted relation is detected to be similar to one of the corresponding TSG in the top K relations ranked by confidence. This metric differs from Recall@K [37] used in SGG in three ways: (1) it compares the entire $\langle subject, predicate, object \rangle$ and not only the predicate, (2) it does not match predicted bounding boxes of relations as TSG are not grounded to the scene, (3) it computes matches using the cosine similarity of language embeddings (from the *all-mpnet-base-v2* model) above an α threshold ($\alpha = 0.77$) instead of exact matching as the vocabulary of TSG and VSG classes is different. We used the Recall@k settings for $k \in [5, 10]$ as downstream tasks must have informative relations predicted with high confidence, a description of the metric is given in Algorithm 3. Instead of solely counting the number of matches, as done in Recall@K, we compute the sum of the rank of the match in the top K predictions. This way, we give more importance to relations that are predicted in the top predictions. We also normalize the sum by the number of relations in the TSG to have a value between 0 and 1.

In the following, we compared the performance of different baseline SGG models using both InformativeRecall@k (IR@k) and traditional Recall@K metrics. We evaluated the baseline models Motifs-TDE [38], VCTree-TDE [37] and Transformer [19] on the task of SGG (predicting

Models	R@20/50/100	IR@5/10/20/50
Motifs-TDE [38]	13.45/17.65/20.76	9.19/10.68/15.28/22.73
VCTree-TDE [37]	13.74/18.26/21.39	9.56/10.86/15.35/22.72
Transformer [19]	24.2/30.6/34.02	11.75/13.26/17.78/24.69

Table 4.3: Comparison between traditional Recall@K metric used in SGG and our newly introduced IR@K for benchmarking models' ability to generate informative relations.

object bounding boxes, labels, and relations) using the codebase provided in these references. We compute the IR@k the same way as R@k from the original papers, except for key differences introduced previously. Results are shown in Table 4.3 where we can see that the Transformer model was the best one to predict informative relations with high confidence. However, there is still room for improvement as the best model had only an average recall of 13.26 for IR@10. IR@20 and IR@50 were also presented allowing a comparison with traditional Recall@K computed on standard ground truth annotations. We observe a correlation between performance in predicting accurate and informative relations. However, predicting informative relations was shown to be more challenging than predicting accurate ones as the difference between the worst and best model in R@K was more than 5 times that between the worst and best model in IR@K (i.e. there is an 11% difference between R@20 of Motifs and Transformer but only a difference of 2.5% between IR@20).

Using the IR@K metric to benchmark SGG models is a first step toward democratizing the usage of SGG in downstream tasks. But we can push our approach further by introducing a new selection process of relations based on *intrinsic* and *extrinsic* information score to further boost the performance of those models. In fact, as we have seen before, SGG models typically predict a very large amount of relations that are either false positives or uninformative triplets. By using our *intrinsic* and *extrinsic* information score, we can re-rank prediction and filter out uninformative triplets. In the next section, we will evaluate our approach first qualitatively and then quantitatively on a set of downstream tasks relying on Scene Graph inputs.

4.5 Informative Inference

Measuring the *informativeness* of relations is a crucial step towards understanding the importance of relations in the context of the scene. However, our ultimate goal is to use this information to improve the quality of SGG models. As we have seen before, one of the main drawbacks of current SGG models is that they generate a large number of relations, most of which can be true but uninformative. During inference in SGG, each relation is ranked with the following formula:

$$\theta_{rel} = \theta_{obj} * \theta_{pred} * \theta_{subj} \tag{4.12}$$

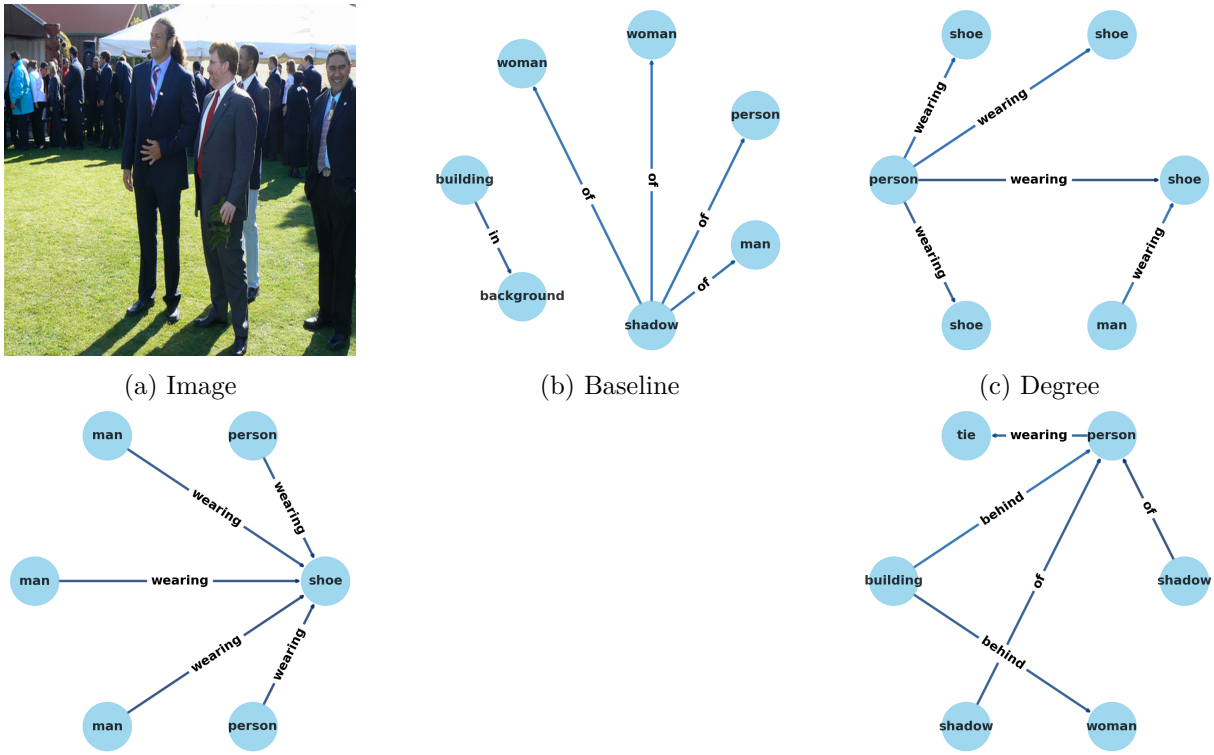
With θ_{pred} being the confidence score of the predicate, given the $\langle subject, object \rangle$ pair, and $\theta_{obj}, \theta_{subj}$ are the respective confidence score of the object detector given to each bounding box. When doing inference, we typically select only high-confidence bounding boxes above an α threshold (for instance $\alpha = 0.8$) so the $\theta_{obj}, \theta_{subj}$ are usually very similar and thus negligible. Because SGG datasets are biased over the most common relations, θ_{pred} is usually high for common relations and low for uncommon relations, which does not necessarily correlate with the *informative* score of the relation. This is why we propose to use the combination of the *intrinsic* and *extrinsic* information value of relations τ_{rel} to re-rank the relations after the predicate prediction stage as follows:

$$\theta_{rel} = \tau_{rel} * \theta_{obj} * \theta_{pred} * \theta_{subj} \tag{4.13}$$

Because τ_{rel} is the average of the *intrinsic* and *extrinsic* information value of the relation, we need to compute the edge betweenness centrality of the entire graph to give a value to each relation. This is computationally expensive for large graphs and is not optimal in the case of SGG because the predicted graph is fully-connected (i.e. all nodes have a relation with all other nodes). To solve this problem, we pre-select the top k relations ranked by θ_{pred} before computing the edge betweenness centrality of the graph. Here we choose $k = 100$ to be consistent with previous works [12], [37] which assume that relations above this threshold are mostly false-positive in the computation of the Recall@K metric. We then compute the edge betweenness centrality of the subgraph containing only the top k relations and re-rank the relations based on the *informative* score τ_{rel}. To demonstrate the approach, we applied it to the scene graphs predicted by the Neural-Motifs model trained on VG150 [38]. Figure 4.9 shows an example of different edge selections for the same set of predictions, based on different centrality measures. The top-5 relations were extracted for 4 different settings: (b) relations ranked by prediction's confidence given by the SGG model (our baseline); (c) relations ranked by centrality using the average of in-degree and out-degree of nodes; (d) relations ranked using the PageRank algorithm and (e) relations ranked using our method based on betweenness centrality. We observe that our method is the one that best describes the scene with homogeneity of nodes and edge types. This example also emphasizes the problem of current SGG models that tend to predict vague and spurious relations with high confidence, see the (b) baseline settings.

4.5.1 Evaluation

In this section, we compared the performance of a few models in SGG with and without our Informative Inference method. We used the same models as before, Motifs-TDE [38], Transformer [19] and PE-NET [49]. The results are shown in Table 4.4 where we can see that the performance in Recall@20 and Recall@50 drops significantly when using Informative Inference while Recall@100 stays similar. This is expected as we are re-ranking relations in the top 100



(a) Image (b) Baseline (c) Degree

(d) PageRank (e) Betweenness

Figure 4.9: Different relations selection method based on edge centrality and importance measures.

Model	Settings	R@20	R@50	R@100	IR@5	IR@10	IR@20
Motifs-TDE [38]	Baseline	7.20	9.93	12.32	4.97	15.89	21.98
	Informative	3.17	8.44	12.32	14.24	18.95	25.86
Transformer [19]	Baseline	19.32	24.97	28.83	12.35	16.08	21.28
	Informative	5.66	14.77	28.83	13.64	19.29	25.64
PE-NET [49]	Baseline	18.38	24.37	27.87	12.49	16.53	21.12
	Informative	5.21	13.56	27.87	18.63	23.91	30.30

Table 4.4: Comparison of different different SGG models with and without Informative Inference selection.

Settings	R@20	R@50	R@100	IR@5	IR@10	IR@20
Informative	5.21	13.56	27.87	18.63	23.91	30.30
Ext. Only	8.39	18.33	27.87	11.39	16.64	22.93
Int. Only	2.80	9.86	27.87	12.79	16.42	22.02

Table 4.5: Performance of the PE-Net model [49] using only *extrinsic* or *intrisic* information for re-ranking.

predictions which does not impact Recall@100. However, the performance in IR@5, IR@10, and IR@20 increases significantly, from an average of 5.4 points for Motifs-TDE to 7.6 points for PE-NET. This shows that our method is able to correctly re-rank relations, making the top predictions much more informative than before. These results also clearly demonstrate the shift between ground truth annotations in SGG datasets and the actual informative relations that should be predicted by SGG models.

Our Informative Selection method is based on *intrinsic* and *extrinsic* information values of relations. It is thus interesting to compare the actual impact of both values on the performance of the model. To do so, we conducted an ablation study where we compared the performance of the model using only the *intrinsic* information value, only the *extrinsic* information value, and both values combined. The results are shown in Table 4.5 where we can see that the performance of the PE-NET model using only the *extrinsic* information value is better than using only the *intrinsic* information value. This is expected as the *extrinsic* information value is computed using the graph structure and thus gives a better representation of the importance of relations in the graph. However, the best performance is achieved when combining both values, confirming that our Informative Selection method is indeed appropriate for the task of SGG. We also observe that the *intrinsic* information-only setting is deteriorating the Recall@K performance more than the *extrinsic* information-only setting. We hypothesize that *extrinsic* information selection tends to re-rank relations that match the ground truth, whereas *intrinsic* information selection is more likely to re-rank other relations due to the distributional shift between the two.

When applying our Informative Selection method, we sample a set of K relations from the top predictions of the model to re-rank them based on their *informative value*. We do not take

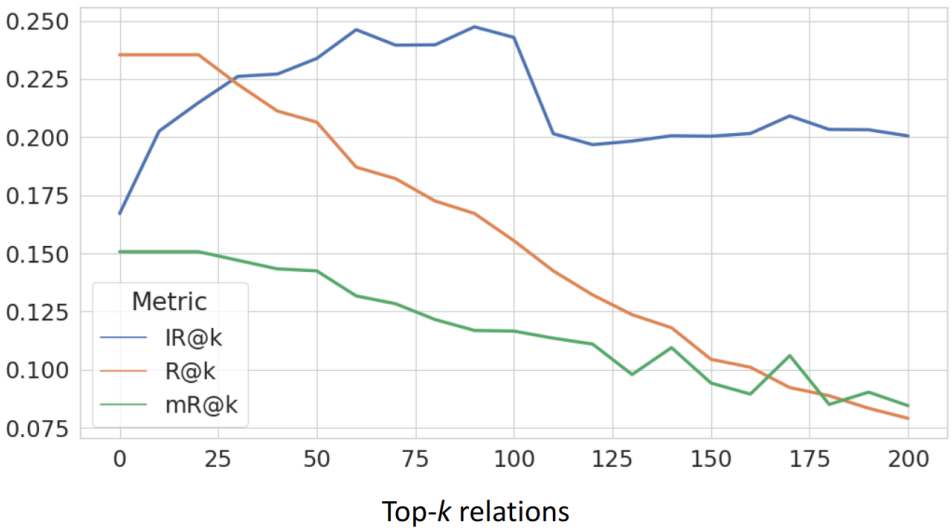


Figure 4.10: Impact of the number of relations to select for re-ranking on the performance of the PE-NET model. For Recall and meanRecall, we average the results for Recall@20, Recall@50, and Recall@100. For InformativeRecall, we average the results for IR@5, IR@10, and IR@20.

into account all predicted relations because their number can grow significantly with the number of objects in the image, hindering a computational overload. In the next section, we will evaluate the impact of the number of relations to select for re-ranking on the performance of the model.

4.5.2 Impact of K

The number of relations to select for re-ranking is an important hyperparameter of our method. Previously, we chose $K = 100$ to be consistent with the Recall@100 metric but other values can be picked. To evaluate the impact of K on the performance of the model, we conducted an ablation study where we compared the performance of the PE-NET model using different values of K, ranging from 0 (no informative selection) to 200. Note that 200 is the maximum value, but a lower amount of relations can be selected in images that contain less than 15 objects ($\sqrt[2]{200} = 14.1$). Results are displayed in Figure 4.10 where we can see that the performance of the model in IR@K increases with the number of relations selected for re-ranking until 100. After that, the performance drops and stabilizes, showing that relations below 100 are likely to be false-positive, as it has been observed in previous work [12]. We also observe a consistent decrease in both traditional Recall@K and meanRecall@K metrics, re-ranking relations in the top 20 of the predictions has no impact on Recall and meanRecall because both metrics start at a value of 20. For cases where very good performance on those metrics is needed, an optimal trade-off of $k = 30$ can be chosen. Otherwise, $k = 60$ or $k = 90$ can be chosen for optimal performance in Informative Recall with this particular model.

As we hypothesized previously, selecting relations based on their informativeness rather than correctness could be beneficial for downstream tasks that rely on Scene Graph inputs. In addition, we hypothesize that non-informative relations hurt the performance of downstream tasks as they introduce noise in the graph and models could wrongly focus on it. In the next section, we will evaluate the impact of our method on three downstream tasks: Image Captioning, VQA, and Image Generation from Scene Graphs. These three tasks were chosen as they are the most common tasks that rely on Scene Graph inputs and are representative of the different types of tasks that can be conducted with Scene Graphs. The biases of SGG models have also already been pointed out in the literature for some of these tasks [14], [60].

4.6 Experiments on downstream tasks

To evaluate our approach, we compared performance on several downstream tasks using (1) ranking based on confidence and (2) ranking based on our Informative Selection method. Three distinct downstream tasks were chosen to this end: Image Generation from Scene Graphs (SG2IM) [65], Visual-Question Answering with Scene Graphs (VQA) [15], and Image Captioning with Scene Graphs [14]. For comparison purposes, results from previous works were reproduced by retraining and evaluating each model using the given codebase from the original references, by using the hyperparameters and settings reported.

The Visual Genome (VG) dataset [8] was used to conduct this investigation, as it has been adopted by the majority of approaches in the Scene Graph-related literature [93]. The following distinct splits were used for downstream tasks, to respect comparison with respective baselines:

- For SG2IM, the VG178 split [65] (178 object and 49 predicate classes) was used.
- For Image Captioning, Scene Graphs have been generated for the COCO-captions dataset using the original NeuralMotifs model [38] trained on VG150, we call this split COCO-sgg.
- Finally, for VQA, the GQA dataset [15] (which is based on a refinement of VG) was used.

4.6.1 Image Captioning

The goal of Image Captioning with Scene Graphs is to generate a short textual description of an image, given the corresponding scene graph and input image. Here we used the TFSGC model [133] for the tests. After retraining the model using authors codebase and parameters, it was evaluated using the predictions generated from the Motifs model [38], in the Original and Informative settings. In Image Captioning the size of the input graph matters as the model equally attends all relations. Thus, we compared our approach by selecting the top 5 relations using the Informative settings to the original full predictions (average length of graphs = 21 relations). We retrained the model for 15 epochs using the codebase provided by the original

Settings	B@4	M	R	C	S
Orig. / Full	34.23	27.21	55.43	109.41	20.33
Inform. / Top5	34.25	27.52	55.55	110.42	20.59

Table 4.6: Results for Image Captioning on the test set of MS-COCO [45].

references, and we performed the evaluation using the standard metrics for the task, namely BLEU (B@4), METEOR (M), ROUGE (R), CIDEr-D (C), and SPICE (S) [133]. For all metrics, higher is better. Results are presented in Table 4.6 where we can see that, by only using the top 5 relations ranked by informativeness, we were able to generate better captions than using the full graph. This shows that our method was able to remove the noise in the graph and focus on the most informative relations, which is beneficial in this case. It also shows that, for the task of Image Captioning, the quality of the relations is more important than the quantity of relations.

4.6.2 Visual Question-Answering

Apart from Image Captioning, the task of Visual Question Answering (VQA) can also benefit from Scene Graph representations [15]. The VQA task aims at answering a set of complex questions given an image. In the settings of VQA from Scene Graphs, we are using the graph to abstract the image and give it as the only input to the neural network that we want to evaluate. Then, the goal is to select the valid answer to the question, usually in a set of four proposals. In the present case, the GraphVQA model [4] was used as a baseline. We retrained the model with the original GQA dataset [15] and then generated graphs for the validation set using the compositional approach [134]. The paradigm of VQA from scene graphs is different from Image Captioning in the sense that the GraphVQA model will attend specifically to relations that match the question keywords. The performance will only increase with more relations, even of bad quality. However, the resources and time consumed to process the graph will grow with the number of relations, thus applying our method here can still be beneficial. To demonstrate this hypothesis, we selected distinct sets with the top k relations as $k \in [10, 20, 30]$ using the Original ranking and our Informative one. We used traditional metrics for the task [135]: answer Accuracy, Consistency, Validity, and Plausibility. Accuracy is the exact matching of answers to the ground truth, consistency, validity, and plausibility are all metrics that evaluate the coherence of the provided answers. Results are shown in Table 4.7, for all metrics, higher is better. We observe that our Informative selection outperformed the Original one by a small margin for all k. With only 10 relations, our settings outperformed the original one with 30 relations. By using the top 30 relations in our settings we are almost matching the accuracy obtained by using all relations, where the average number of relations per graph was 143.23 (almost 5 times more). This shows that selecting informative relations first is indeed beneficial to remove the noise in the VQA task. It also shows that the predictions of the SGG models are of poor quality. In fact, we observed

	Settings	Accuracy	Consistency	Validity	Plausibility
	Ground truth	60.78	90.42	92.62	87.5
Original	Full	38.54	79.61	88.79	80.31
	Top 10	37.45	78.42	87.42	78.36
	Top 20	37.94	79.65	87.79	79.17
	Top 30	38.19	80.38	88.02	79.50
Inform.	Top 10	38.23	81.22	88.0	79.35
	Top 20	38.41	80.72	88.26	79.79
	Top 30	38.48	80.89	88.39	80.02

Table 4.7: Results on the validation set of the GQA dataset [135].

that relations after the top 10 only slightly impact the performance of the model, meaning that they are likely to be false positives or uninformative relations. The gap in all metrics between the predictions and the ground truth can be explained by the size of the GQA dataset. In fact, the GQA dataset contains more than 1,700 object classes and 300 different predicate classes, making it very challenging for the SGG model to predict both objects and relations precisely.

4.6.3 Image Generation

Image Generation from Scene Graphs (SG2IM) [65] aims at generating corresponding representations solely from input SGs. To benchmark our approach we used the latest state-of-the-art model, SGDiff [36], which is based on Latent Diffusion [136]. We first trained the model using the original VG178 dataset by following the authors' codebase. Then, we generated scene graph predictions for the test set of VG178 using three SGG models, Motifs [38], VCTree [37], and Transformer [19]. We named the ranking of predictions based on confidence as the original setting and our new ranking as the Informative setting. As the average number of relations in the VG178 dataset is around 6 per image, we selected the top 5 predictions for both settings to be consistent with the training set of the SGDiff model. All experiments used an image size of 256x256 pixels and a sampling size of 200 for the DDIM sampler (i.e. number of steps in the denoising process). The results obtained are shown in Table 4.8 and the metric used is the Fréchet Inception Distance (FID) [137], which evaluates the distance in the latent space of an InceptionV3 model between the distribution of the ground truth images and the generated ones. An FID of 0 represents an exact matching between original images and generated ones. The Inception Score (IS) evaluates the diversity and quality of generated images alone and was only reported here for comparison with other image generation approaches. We can see that for all predicted scene graphs, selecting relations based on informativeness led to better results than using confidence, outperforming the latter by an average of +1.38 on the FID metric. These results also show that our Informative Selection method can get consistent improvements

Model	Settings	FID ↓	IS ↑
Ground truth		23.54	18.02
Motifs [38]	Original	26.37	17.25
	Informative	25.48	17.35
VCTree [37]	Original	26.6	16.88
	Informative	24.32	17.49
Transformer [19]	Original	25.22	17.11
	Informative	24.26	18.83

Table 4.8: Performance of the SGDiff model [36] for 256x256px settings with original and informative predictions from different SGG models.

over the predictions given by different models, making it general and applicable to any SGG model. Finally, we validate the usage of the IR@K metric for selecting a good backbone for this task as the results obtained in SG2IM are consistent with values of IR@K computed for each model previously (i.e. the best model in IR@K performs the best here and the worst in IR@K is the worst here), see Table 4.3. Figure 4.11 shows a qualitative example of our approach for Image Generation from Scene Graphs. We can easily see that the top predictions from the original model (left graph) are not informative at all, misleading the diffusion model to generate a blurry background (left image). By selecting informative relations instead (right graph), the model generated a background that resembles that of the original image (right image). This example, and quantitative results from Table 4.8, show how the current SGG models are biased into generating uninformative relations first as those relations are more likely to be true, hindering the performance in downstream tasks. Here, we also want to emphasize the importance of *Topological* relations in the performance of SG2IM models. Because diffusion models reconstruct the image pixel by pixel, relations that are more likely to impact the distribution of pixels in the final image are topological relations between important regions of the image (right graph) rather than fine-grained details of a specific region which could be predicted with high importance by an SGG model (left graph).

The success of our approach in the task of Image Generation is especially important because it is a task that, in contrast to VQA or Image Captioning, relies solely on scene graph inputs. This is why we are observing a greater improvement in the different metrics here than in the previous two tasks. For instance in VQA, inputs also contain the question and for Image Captioning the image in itself, where both can impact the performance of models.

4.7 Concluding Remarks

In this chapter, we have presented a new concept related to the semantic importance of relations in scene graphs, which we call *informativeness* of relations. We have shown that the

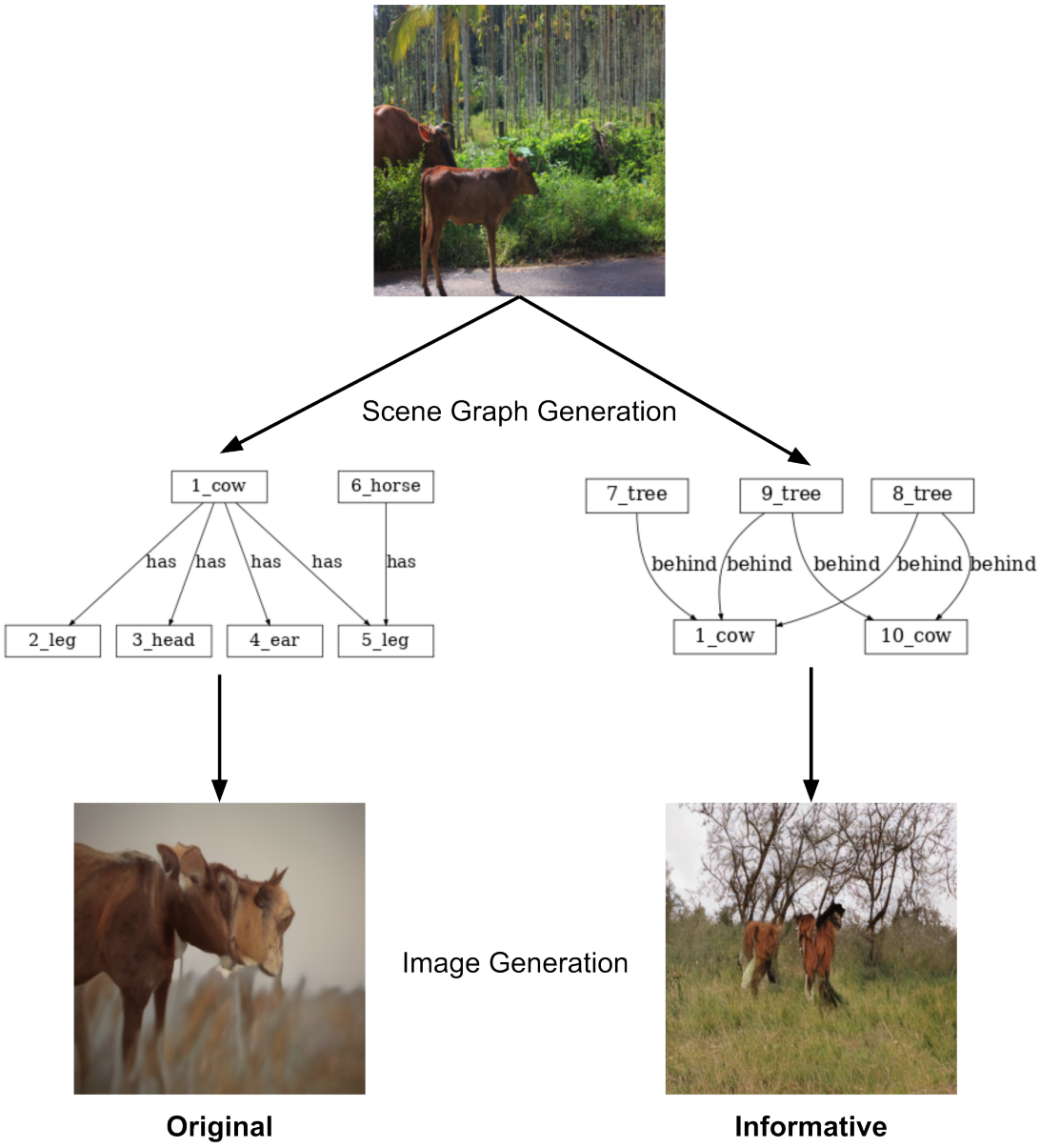


Figure 4.11: Comparison of generated scene graph images from original Motifs predictions (left), and Motifs predictions refined with Informative Selection (right). Final images (bottom) were generated with the SGDiff model [36].

current metrics used in the evaluation of SGG models are not representative of the actual importance of relations in the graph. We defined the importance of relations as a combination of *intrinsic* and *extrinsic* information values. For both values, we have introduced new methods to compute them efficiently: similarity with image captions for *intrinsic* information and edge betweenness centrality for *extrinsic* information. Based on *intrinsic* information value, we proposed a new metric: Informative Recall@K (IR@K) to evaluate the performance of SGG models to be used in downstream tasks. We showed that this new metric is more representative of the actual importance of relations in the graph than traditional Recall@K. We then proposed a new inference method to re-rank predictions from an SGG model based on their *informativeness* and showed that this method can improve the performance of SGG models in downstream tasks. Our approach has been evaluated on three different tasks: Image Captioning, VQA, and Image Generation from Scene Graphs. We showed that our method can consistently improve the performance of models in those tasks, demonstrating the importance of selecting informative relations in the context of SGG. This approach is general and can be applied to any SGG model, without retraining or fine-tuning, making it a powerful tool to improve the performance of SGG models in downstream tasks. Furthermore, we provided additional insights into the informativeness of the relations categories introduced in Chapter 3 by linking them to our definition of *intrinsic* information value. We showed that the *Topological* and *Functional* relations are more informative than the *Part-Whole* and *Attributive* relations, which is consistent with the results obtained in the task of Image Generation from Scene Graphs.

Our contributions in this chapter can also be applied to other tasks and contexts beyond those we have evaluated here. For instance, the Informative Selection method can be used to improve the performance of SGG models used in service robotics. Service robots and robots in general require strict constraints regarding quality of data but also inference time, as new predictions need to be made frequently to account for changes in the environement. The current paradigm of SGG, as we have seen, is not adapted to this context as it does not take into account any notion of time or resource efficiency. In the next chapter, we break down current architectures of SGG models and propose further solutions to adapt them to the real-time constraints of service robotics.

Chapter 5

Real-Time SGG

> Resource efficiency - it's about only taking what we need.
>
> Hilary Benn

Real-world applications, and especially robotics applications, require robustness and efficiency to meet real-time constraints. In previous chapters, we have tackled the challenges related to robustness and performance of SGG models for downstream tasks. In this chapter, we will focus on the efficiency of SGG models, and their deployment in applications that require real-time inference. Despite the recent advances in SGG , the task remains computationally expensive and is not yet suitable for real-time. In fact, efficiency in terms of computational resources or time is not a concern at all in the field of SGG as only a few approaches are reporting latency metrics [26], [138]. In this chapter, we aim at solving this gap by proposing a new architecture for SGG that is efficient in terms of latency and computational resources.

It is difficult to define a proper real-time constraint for a SGG task, i.e. if we want to embed a SGG algorithm onboard a robot, what would be the desired latency of the SGG model? This question is bounded by two parameters: (1) the expected frequency of changes in the environment and (2) the expected frequency of updates in the downstream task. Regarding (1) we can assume that in domestic contexts, relations changes are frequent when we are identifying complex actions, possibly involving multiple humans. Traditionally, in object detection, a rate of 10Hz+ is considered as taking into account most of these changes. Regarding (2), a desired rate of update for planning systems is usually between 5 to 50hz (20 to 200ms per detection). As a consideration, we will assume a goal of at least 20Hz in our approach here (50ms per detection), benchmarked on standard hardware (a unique laptop with a standard GPU). Having a lower latency than 50ms on a laptop should ensure adequate latency on an edge computing device. In this chapter, we will focus on fundamental changes to SGG architectures and not on implementation details to meet these requirements. Actual optimizations for real-time inference of deep neural networks (e.g. quantization, model pruning etc...) are outside the scope of this work.

We first analyzed the real-time performance of different SGG models which are considered

Model	Model Latency (ms)	Params (M)	FLOPs (G)
Motifs [38]	398.63	274.51	1030.70
VCTree [37]	519.53	357.89	1083.62
Transformer [19]	381.64	327.98	1025.58
GPS-Net [47]	377.51	392.81	1212.14
PE-NET [49]	277.62	425.99	2218.91

Table 5.1: Real-time performance of SGG models equipped with the Faster-RCNN backbone. Model latency measures the overall latency of the forward pass of the model (object detector + relation prediction) with batch size 1. FLOPs is the Floating Point Operations per Second of the forward pass of the mode, computed with the PyTorch profiler.

baseline models in the field in table 5.1[1]. We can see that none of them meet our requirements for real-time (i.e. latency < 50ms). The lowest latency is the PE-NET [19] model with 277.62ms and the higher latency is the VCTree-TDE [37] model with 519.53ms, which are respectively 5.54x and 10.38x slower than our higher bound. The number of parameters of these models is also very high, considering the difficulty of the task. In addition of latency, we measured the complexity of each models using Floating Point Operations per Second (FLOPs) which is a measure of the number of floating-point operations that a model performs in a second. The FLOPs of the models are also very high, ranging from 1025.58 GIGA FLOPs to 2218.91. This is a very high number of operations for a single forward pass of the model, which is likely to fail on edge devices.

Defining the current bottleneck of those models in terms of latency is complex, as approaches often rely on the aggregation of diverse modules. The standard approach that most models are using is called the "two-stage approach" as it relies on the aggregation of a frozen object detector and a custom model for scene graph prediction. The two-stage approach can be decoupled into four different modules, paired as the following pipeline:

1. The Feature Extraction step, which generates different visual features and proposals from the image ;
2. The Feature Refinement step, which aggregates those features to form nodes and edges representation ;
3. The Context Learning step which refines the representation of nodes and edges with contextual information at the graph level ;
4. The Scene Graph Prediction module, which takes as input the bounding boxes and associated features to predict the final graph composed of objects, bounding boxes, and

[1]Hardware used: 11th Gen Intel™ Core™ i9-11950H @ 2.60GHz x 16, NVIDIA GeForce RTX 3080 Laptop GPU 16GB VRAM, 2 x 16GB 3200 MHz RAM.

relations.

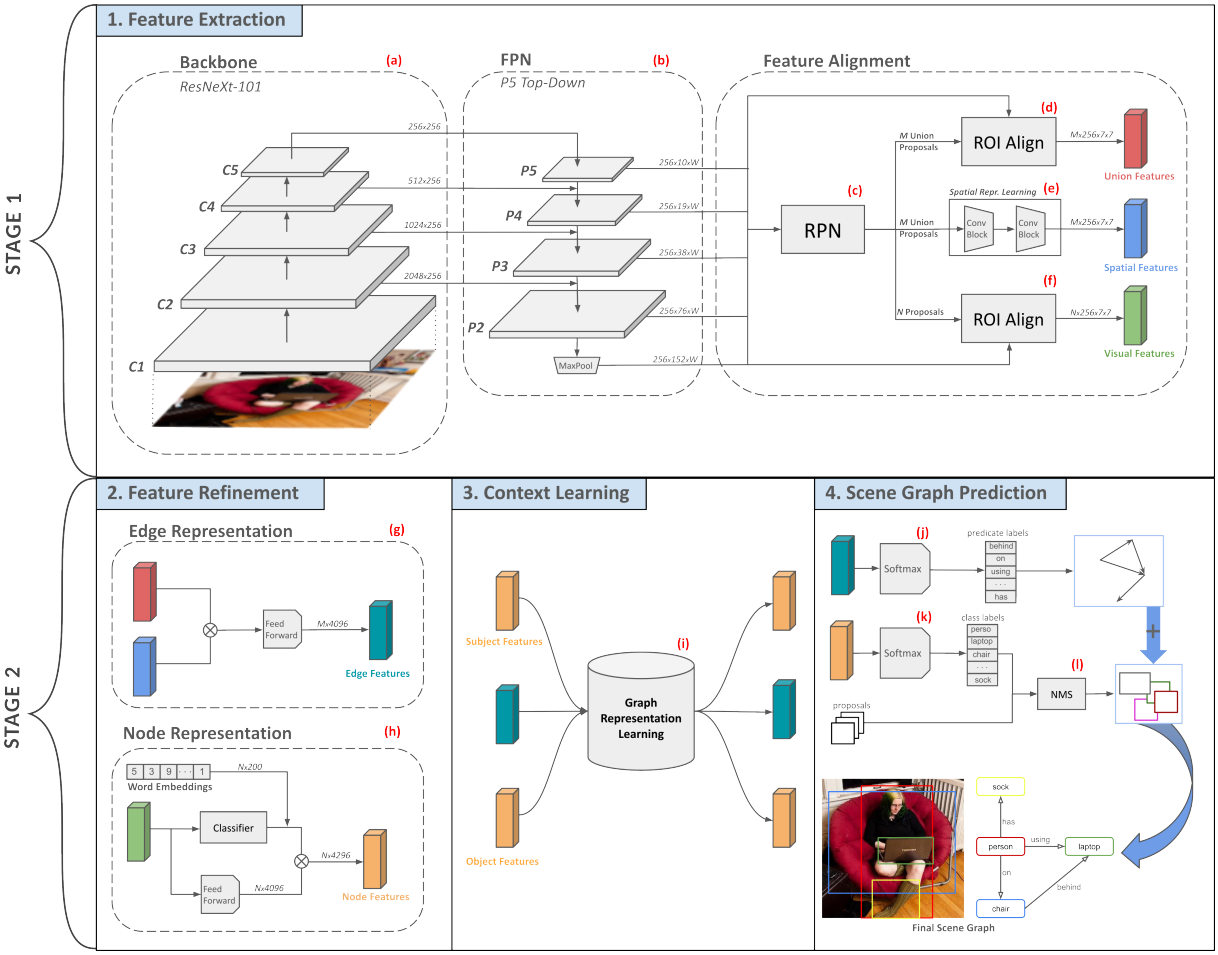


Figure 5.1: Pipeline of a typical SGG model. Stage 1 correspond to bounding box regression (see **(c)**) and features extraction (**(d)**, **(e)** and **(f)**). Stage 2 is responsible for features aggregation and refinement (**(g)** and **(h)**) as well as context learning (see **(i)**) and final decoding (**(j)**, **(k)** and **(l)**). $\bigotimes$ denotes element-wise concatenation.

In Figure 5.1, we display the pipeline of a typical SGG model [12], [19], [37], [47], [49]. The first stage (first row, module "Feature Extraction") has been proposed by Tang et al. [37] based on the Mask-RCNN implementation [139] of the Faster-RCNN object detector. We can break-down this stage in three different blocks: the ResNeXt-101 backbone for features generation (Figure 5.1**(a)**), the Feature Pyramid Network (FPN) [140] (Figure 5.1**(b)**) for features enhancement and the Region Proposal Network (RPN) [141] (Figure 5.1**(c)**) for bounding box proposal generation. The main differences between this implementation for SGG and the original Mask-RCNN implementation are the second ROI Align performed on the union of proposals (Figure 5.1**(d)**) and the addition of a spatial feature module that generates features from box

coordinates (Figure 5.1**(e)**). Another step of ROI Align (Figure 5.1**(f)**) is performed on every proposal, as in the original Faster-RCNN approach [141]. ROI Align is a technique that extracts features from a fixed-size feature map for a set of bounding boxes, by aligning the feature map with the bounding box coordinates. In this particular implementation, the feature map is composed of the five different feature dimensions extracted by the FPN (Figure 5.1**(b)**) at five different scales. For each bounding box, the ROI Align algorithm selects the appropriate layer of the feature map from which to sample the features. Bigger bounding boxes will tend to be matched with the last layer of the feature map (P2) while smaller bounding boxes will be matched with the first layer (P5) for smaller resolution. Then, the features are aligned with the corresponding coordinates of the bounding box using bi-linear interpolation. Features are extracted as a fixed dimension of 7x7 for each box (small or big). For the ROI Align performed on the union of proposals (Figure 5.1**(d)**), the input bounding box coordinates are computed on the union of every proposal, by taking into account the top-left corner of the left-most box and the bottom-right corner of the right-most box. The output of the ROI Align on the union of proposals (Figure 5.1**(d)**) and the spatial features (Figure 5.1**(e)**) are then fed to the edge representation model (Figure 5.1**(g)**). On the other hand, the output of the ROI Align on every proposal (Figure 5.1**(f)**) is fed to the node representation module (Figure 5.1**(h)**).

In the second stage of the pipeline, we have three different modules: the Feature Refinement, Context Learnings and Scene Graph Prediction modules. In the Feature Refinement, the union and spatial features are concatenated to represent the edge features (Figure 5.1**(g)**). At the same time, the visual features are decoded using a classification head to predict the object labels (Figure 5.1**(h)**). These object labels are then used to retrieve corresponding word embeddings for each object. These embeddings, concatenated with the original visual features, will serve as the node features for the graph. In the Context Learning module, the edge features are aggregated with the node features to form a graph representation (Figure 5.1**(i)**). It is in this step that most of the learning is done, as the model will learn to predict the relations between pairs of objects. Most of the approaches in SGG are only modifying this step to improve the performance of the model [19], [47], [49]. Finally, in the Scene Graph Prediction module, the predicate labels are decoded using a standard softmax layer (Figure 5.1**(j)**). The object and subject class labels are also decoded using softmax (Figure 5.1**(k)**) and then matched with corresponding proposals before a step of Non-Maximum Suppression (NMS) (Figure 5.1**(l)**). NMS is a technique to merge overlapping bounding boxes, which are then used to generate the final scene graph.

We identified two main limitations in the two-stage architecture: (1) the use of Faster-RCNN as a feature extractor is not optimal for real-time applications and (2) the feature alignment and feature refinement step are overly complex and not efficient in terms of latency. This last point mainly result from the usage of an RPN which requires performing object detection on every proposal, which is computationally expensive (see Figure 5.1**(c)** and Figure 5.1**(d)**). To solve

these challenges, using a single-stage object detector such as You Only Look Once (YOLO) [27] seems promising. YOLO is an object detector which does not use an RPN and ROI Align, as it is done with Faster-RCNN. Instead, YOLO decode both bounding boxes and object labels in a single head of the network. The latest versions of YOLO, such as YOLOV8 [39], are heavily optimized for a trade-off between latency and accuracy, making them a very good choice for real-time SGG. However, integrating YOLOV8 in the SGG architecture is not straightforward as YOLOV8 does not use box candidates and does not have a traditional feature extractor component. Thus, our goal in this chapter is not only to modify the feature extraction stage of the SGG architecture but also subsequent modules which may depend on it.

Before describing our new architecture, we will first review the state of the art in real-time SGG and the limitations of current approaches in Section 5.1. Then, in Section 5.2, we will describe the modifications made to the feature extraction stage of the SGG pipeline to use YOLOV8 as a feature extractor. To make these changes efficient, we need to also modify the feature alignment and feature refinement steps of the SGG pipeline, as these steps goes hand-in-hand. We will detail the changes made to these steps in Section 5.3. Improvements can also be made to the context learning and final graph prediction steps of SGG models, as we will see in Section 5.4, by lowering the complexity of the relation learning without sacrificing performance. Finally, we will discuss the overall challenges of the task of SGG for its adoption for real-time and real-world applications and the limitations of our approach in Section 5.5.

5.1 State of the Art in Real-Time SGG

The term Real-Time SGG has not been widely adopted by the community, to our knowledge only a single approach uses the term “real-time SGG” [138]. However, a set of recent approaches are reporting latency metrics in their work, showing a growing interest in efficient implementations. Specifically, we can separate approaches in SGG into two categories: two-stage approaches and one-stage approaches. The former uses a two-stage pipeline with the Faster-RCNN backbone, while the latter uses a single-stage pipeline, to infer both relations and object proposals directly from the image features. As a result, this second category of approaches is often more efficient for real-time processing than the first. However, a few challenges remain, especially concerning the accuracy of such approaches for object detection. In the following, we review those limitations and challenges for both two-stage and one-stage approaches.

5.1.1 Two-Stage Approaches

In a recent work, Jin et al. [26] introduced an approach for real-time SGG based on contextual information. In contrast to every other SGG approaches, this work does not use visual features to infer relations. Instead, it uses object bounding box coordinates to learn the correlation

between respective positions in the image and predicate classes. To generate the bounding boxes, the use of YOLOV5 is reported for real-time object detection. The reported results show a latency of 29ms with 33.5 Frames Per Second ()FPS) for a state-of-the-art F1@K metric in the task of SGG. However, these results are difficult to put in perspective due to the choice of using only box coordinates as inputs. In fact, it is a highly questionable choice for the overall purpose of the task which is to understand fine-grained compositional relations at the image level. In its present form, the model would never be able to disambiguate between different relations that may have similar bounding box coordinates such that $\langle person, holding, bottle \rangle$ and $\langle person, drinking\ from, bottle \rangle$. By removing the dependence on visual features, the model can learn very efficiently the statistical co-occurrences of the dataset and thus achieves very good F1@K performance but is very likely to fail to generalize to other datasets or out-of-distribution images. In another work [138], Jin et al. propose to achieve real-time SGG by leveraging a Relation-aware YOLO structure (RYOLO). RYOLO is composed of two parallel branches, one for classical object detection using YOLOV5 and one for relationship prediction using anchor orientation on the visual feature maps. The idea is to predict the coordinates of relations directly from the feature maps and then do a step of matching with relatively spatially close objects in the scene that have been detected by YOLOV5. Due to both operations being performed in parallel, the approach is able to attain a low latency of 28ms or 25.7FPS. However, performance in the task of SGG is poor and more importantly, the performance of YOLOV5 for object detection is also poor. The poor performance in relation prediction may be due to the design of the model, which needs to match relations detected with the closest object coordinates, which may not be efficient for cluttered scenes with a lot of similar objects.

5.1.2 One-Stage Approaches

The first one-stage approach to the task of SGG is Sparse-RCNN [142]. This approach employs a strategy of triplet querying to generate object proposals and relations. Once visual features have been extracted from a CNN backbone, boxes and relations are decoded altogether using a cascade-RCNN scheme. This approach reports state-of-the-art Recall and meanRecall@K metrics for a latency of 190ms (5.26FPS), which is not real-time. RelationTransformer (RelTr) [143] and SGTR [79], [144] are two Transformer-based one-stage approaches to the task that report low-latency with 13.4FPS and 6FPS, respectively. RelTr uses a DETR-based approach with a ResNet-50 backbone and SGTR is a custom Transformer-based approach with a ResNet-101 or ResNet-50 backbone. Both approaches generate a sparse set of relations between a selected set of pairs of interest, in contrast to traditional two-stage approaches which will predict relations for all possible pairs of objects. Due to this design, these approaches maintain a good trade-off between performance in SGG and latency, however, they still suffer in object detection. In fact, because object proposals and labels are decoded all together with relations, the models will tend

to only predict objects that can be easily paired with a relation. This design is not optimal for real-world applications as it is likely to miss relations between important objects in the scene [144]. Also, we want to point out the lack of a clear benchmark for latency, as all approaches are benchmarking their model using different hardware and settings. Some approaches are even not reporting the hardware used for latency [142], which is an issue for fairness. In our work, we report all latency with the same hardware and settings.

Authors of RelTr have compared their work with two-stage approaches such as Motifs [38] and VCTree [19] in terms of latency. The comparison shows a clear disadvantage for any two-stage approaches as none of them succeed in achieving more than 6FPS. However, as we have seen previously, this may be biased by the usage of Faster-RCNN for object detection. Specifically, we strongly believe that lower latency and better accuracy can be attained by combining the best of the two-stage approaches with a real-time object detector such as YOLOV8. In the following, we will prove this last point by modifying the pipeline of two-stage approaches to be more efficient in real-time, with no loss of accuracy. We will demonstrate the benefits of this solution by combining the YOLOV8 model with a set of different relations prediction heads used in two-stage approaches. Nonetheless, this is not straightforward and requires first a review of the architecture design of two-stage approaches.

5.2 Feature Extraction

Extracting high-quality visual features for SGG is critical for the overall performance of models. Visual features will be used during two stages: the object detection stage and the relation prediction stage. One could use a different pre-trained feature extractor for each stage, however, this will be computationally expensive. Instead, in traditional approaches, the features extracted from a unique backbone are shared by the object detection head as well as to the relation head. In the case of Faster-RCNN in SGG, approaches traditionally use a ResNeXt-101 architecture paired with a Feature Pyramidal Network (FPN) [140] to extract features from different depths to construct a multi-scales feature map (traditionally features from 5 stages are extracted). This design ensures the efficient extraction of the maximum of information from the backbone. It is known that, in ResNet-type models, deeper layers are responsible for coarse information about the image whereas shallow layers are responsible for fine-grained information. This can be used to efficiently model the representation of small to large objects. These feature maps are then fed to a RPN to generate proposals. The proposal coordinates are then used to align and "crop" the feature from corresponding layers to extract a deep representation of each Region Of Interest (ROI) in the ROI Align stage. It is after this stage that the corresponding proposal and visual features are fed to the relation prediction head. In classical SGG, the relation head is not only responsible for predicting relations between pairs of objects but also the classes of the objects,

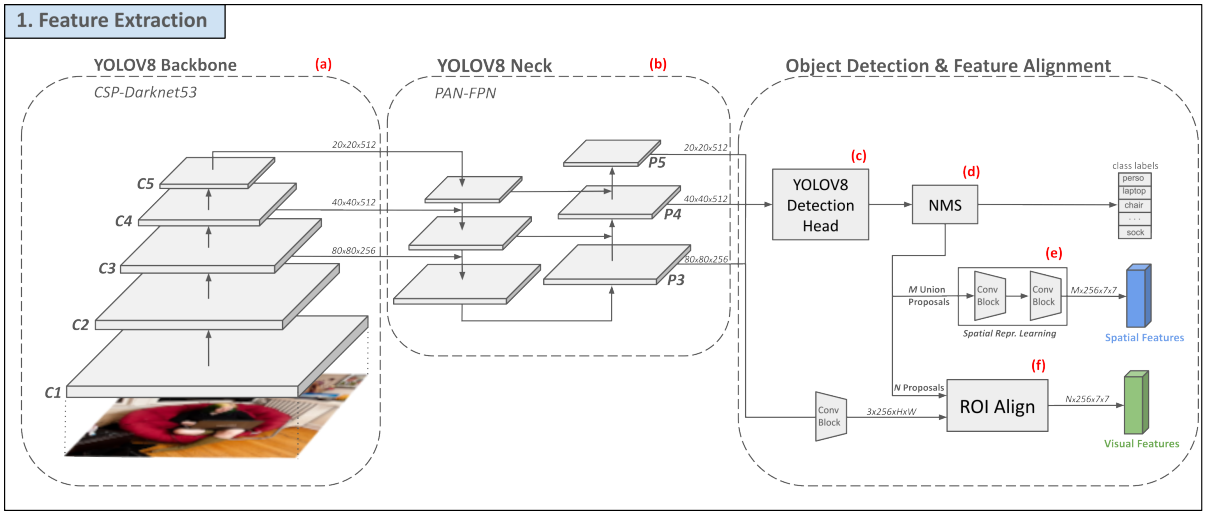


Figure 5.2: Modified SGG architecture with YOLOV8 as feature extractor.

a task traditionally performed by a dedicated classification head after the ROI align stage.

On the other hand, real-time object detectors such as YOLO [27] are what we call one-stage detectors because they do not possess an RPN and process features straight out of the backbone (sometimes using an FPN but not always) with regression and classification heads. In addition to extracting features from interesting layers, the FPN is responsible for upsampling the features by integrating context from other layers for better representation [140]. In YOLOV8, the FPN is followed by a Path Aggregation Network (PAN). The combination of those is called the "neck" which performs a top-down and bottom-up pass of features at three different scales instead of five for Faster-RCNN, see Figure 5.2 **(b)**. The three different feature maps are then fed to three different "decoupled heads" which each comprise a regression and a classification module, see Figure 5.2 **(c)**. To respect the original two-stage approach of SGG models, we extracted the upsampled features from YOLOV8 after the Neck, which corresponds to the P3, P4 and P5 layers of the overall architecture. Features could also be extracted directly after the Backbone and post-process by the original FPN of Faster-RCNN, however, this would lead to extensive overhead. Because YOLOV8 does not use an RPN and ROI Align as in Faster-RCNN, we needed to modify the original ROI Align algorithm of Faster-RCNN to extract corresponding features for each object proposal. After the forward pass of the YOLOV8 model, we feed the three feature maps and bounding boxes coordinates from the regression head to the original ROI Align algorithm [139]. At the difference of Faster-RCNN, YOLO's feature maps do not possess the same number of channels, P3 has 256 channels, and P4 and P5 have 512 channels. We then add one 1x1 Conv layer to downsample P4 and P5 feature maps to 256 channels before performing Region of Interest (ROI) Alignement (see Figure 5.2 **(f)**). ROI Align is a technique to accurately extract features from proposed regions within an image. It improves upon ROI

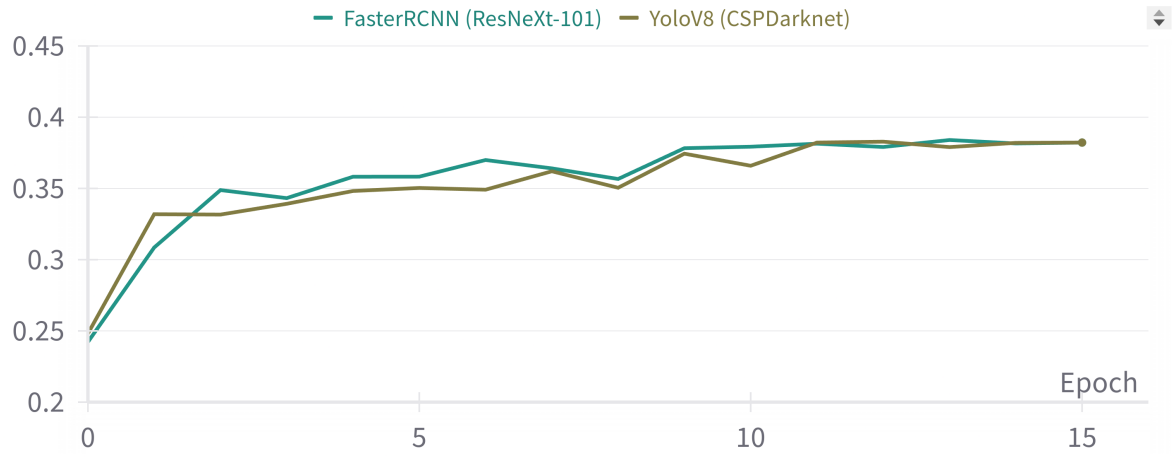


Figure 5.3: Performance in F1@K for relation prediction of different models [19], [37], [38] equipped with Faster-RCNN and YOLOV8 backbones.

Pooling by avoiding quantization and using bilinear interpolation to compute precise feature values. ROI Align works by dividing each candidate bounding box as a fixed grid size (in our case a 7x7 grid) and then computing interpolation points with the feature vector for each grid cell. This results in a set of $n \times 7 \times 7$ features which can be used for further processing.

We evaluated the impact of these changes by training different relation prediction models for the task of SGG. For all experiments, we keep all modules except the feature extractor unchanged. To reproduce the training strategy employed in SGG, we first trained the YOLOV8 model, froze its weights, and then trained the rest of the SGG model upon it. During training and evaluation, we purposely input ground truth bounding boxes and pairs to evaluate only the prediction of relations. This type of evaluation is known as predicate classification (PredCls). In Figure 5.3 we display a comparison of the performance of three different baseline models, Motifs-TDE [38], VCTree-TDE [37] and Transformer [19] equipped with the feature extractor of YoloV8 (CSPDarknet-53) and Faster-RCNN (ResNeXt-101). For each model, the metric used is F1@K which is the harmonic average between the commonly used metric Recall@K, and meanRecall@K. We display the average obtained by all models for better visualization. We observed almost no difference between the two feature extractors, which can signify that either the visual features are not important in the learning process of relation prediction or that the quality of the features generated by the YoloV8 backbone and the Faster-RCNN ones are very similar. This is an interesting finding, as the size and architecture of both backbones are very different. However, the implication of such findings is left for future work.

Generating good feature representations is a critical step in the SGG pipeline. In the next

section, we will focus on the object detection step and how to improve it by using YOLOV8.

5.3 Object Detection

The traditional approach in real-time object detection employs a bounding box regression and a classification head after the feature generation step. In the original Faster-RCNN implementation, regression and classification are done in a sequence, and then a final step of Non-Maximum Suppression (NMS) is applied to merge overlapping bounding boxes. In the case of the two-stage approach in SGG with a Faster-RCNN backbone, the strategy employed is to decode object classes a first time before the context learning, and a final time after the context learning which will be used for the final prediction. While in the original Faster-RCNN implementation, a simple softmax is applied to the class logits, SGG approaches are introducing their method to decode classes, for instance, a TreeLSTM for VCTree [37]. This strategy has been employed as we assume that the prediction of a relation could improve the performance of the object classification. For instance, for a given pair of proposals, if the detected relation is *wearing* and the subject label is *man*, there is a high chance that the object label will be *shirt*. The first approach to report this strategy is Neural-Motifs [38] where a contextual representation of parent nodes is used to decode the child node labels using an LSTM network. This strategy has been employed with a TreeLSTM in VCTree [37] and then used various times with different contextualized decoding in further approaches [19], [47]. It is very important to notice here that, to our knowledge, none of these approaches performed ablation studies to confirm if this strategy is indeed beneficial for the task (i.e. comparing the performance in object detection with and without contextualized decoding). This makes the performance of SGG models different from each other on the task of Object Detection, even though they all report using the same Faster-RCNN checkpoint. This can lead to confusion in the comparison of models for the task of SGG, as the difference in performance for object detection may hinder the comparison of relation prediction modules.

On top of that, in SGG mode the step of NMS is performed after the relation prediction stage (see Figure 5.1**(l)**) which will also influence the predicted objects. To measure this shift, we draw a comparison of the performance of the Faster-RCNN model trained on the IndoorVG dataset with different classification heads in table 5.2. In this table, we can see that the performance in object detection drops significantly, from 1% for Motifs-TDE to almost 10% in mAP for GPS-NET by comparison to the original Faster-RCNN implementation [141]. These results show a true dependence between the two stages of SGG methods, in contrast to what is usually accepted in the community. The ability to accurately detect objects in the scene directly impacts the performance of models to generate the graph, as relations are first evaluated by the correspondence of the $\langle subject, object \rangle$ pair with the ground truth. As a result, the performance

Backbone	Detection Head	Classification Head	$\mathbf{mAP^{50}}$	$\mathbf{mAP^{50\text{-}95}}$
ResNeXt-101 [145]	Faster-RCNN [141]	*Faster-RCNN [141]*	27.2	11.5
		Motifs-TDE [38]	26.2	11.6
		VCTree-TDE [37]	25.5	11.1
		PE-NET [49]	25.2	11.2
		Transformer [19]	24.2	10.6
		GPS-NET [47]	17.4	7.1

Table 5.2: Performance on Object Detection of Faster-RCNN before and after the Relation Prediction stage with different SGG models on the IndoorVG dataset. The baseline (top row) is the Faster-RCNN implementation with a ResNext-101 backbone as described by He et al. [139]. In this implementation, the Classification Head is a simple average pooling followed by a linear layer.

of SGG models in object detection becomes slightly worse after the relation training stage than before, biasing a fair evaluation of the task.

To solve this problem, we propose to completely make independent the first and second stages of the SGG process. To do so, we freeze the regression but also the classification head of YOLOV8 and perform Non-Maximum Suppression (NMS) before the relation prediction stage, see Figure 5.2 **d**. The objective of the relation prediction stage becomes then to predict only correct predicates, in contrast to also predicting the class labels of objects. This significantly lowers the complexity of the relation prediction stage and thus the computational load of the model, while maintaining similar performance in mAP for all SGG models.

To evaluate the performance of this new architecture, we first trained the YOLOV8 object detector on the IndoorVG dataset. YOLOV8 comes in different variants, ranging from nano to x-large. Each variant is a different scale from the original model, both for depth and width, making optimal trade-offs between accuracy and latency for a wide range of use cases. In our experiment, we used the medium version YOLOV8m. We trained the model for 50 epochs with a batch size of 32 and a learning rate of 0.001. We used other default parameters of the original implementation of YOLOV8 [39]. To make the model converge faster, we fine-tuned the provided checkpoint pre-trained on the COCO dataset [45] by authors of the original implementation [39]. With this method, we obtained a mAP@50 of 31.2 and a mAP@50-95 of 17.1 on the IndoorVG test set. When used in the relation prediction stage of an SGG model, this stays strictly similar after any full model training (see Table 5.1), which also helps to fairly evaluate the quality of relation predicted by SGG models. We also observed a significant drop in latency for YOLOV8 against Faster-RCNN, as seen in Table 5.1 and Table 5.2. This is encouraging as it shows that significant improvements can be made to reduce the latency of SGG models. In the next section, we will evaluate the impact of these changes on the relation prediction stage of SGG models, in

Backbone	Heads	mAP^{50}	mAP$^{50\text{-}95}$	Latency (ms)	Params (M)
CSPDarknet [39]	YoloV8 [39]	31.2	17.1	9.28	25.91

Table 5.3: Performance on Object Detection of YOLOV8 on the IndoorVG dataset. Latency is computed with similar settings as in Table 5.1.

terms of overall performance in Recall@K and meanRecall@K but also InformativeRecall@K.

5.3.1 Comparison with two-stage approaches

In the following, we compared the performance in latency and performance metrics between traditional two-stage approaches with Faster-RCNN equipped with a ResNeXt-101 and our new architecture which uses YOLOV8 exclusively as an object detection backbone. We used IndoorVG as a dataset for the experiments. To make the comparison as fair as possible, we kept all parameters similar during training and inference for all models, except of course parameters related to the object detection backbone in itself.

We experimented with our approach for five different relation prediction models: Neural-Motifs [38], VCTree [37], Transformer [19], GPS-Net [47] and PE-NET [49]. For all approaches, we retrained the original implementations with both the Faster-RCNN backbone and our modified YOLOV8 backbone. All models have been trained for 20 epochs with a batch size of 8 and a learning rate of 0.001 with the SGD optimizer with a momentum of 0.9. These settings are similar to previous approaches [19], [49]. At the difference of Faster-RCNN, we used fix size for the image inputs as it is required for YOLOV8. In the original implementation of Faster-RCNN, the input size of the image is a minimum of 600 pixels for the height and 1000 pixels for the width. Images are then pre-processed to fit the first of those requirements by keeping the original aspect ratio. For instance, an image of size 400x500 will be reshaped to 500x600, an image of 900x450 will be reshaped to 1000x500, etc. For YOLOV8, all images need to be reshaped to a square size, we chose 640x640 as this is the default size used for YOLOV8. In addition, to fairly evaluate all approaches, we set a fixed seed of 42 for all random number generators and made use of deterministic operators in PyTorch [2]. Finally, we benchmarked latency on one Nvidia RTX3080 GPU with batch size 1 and input image size 640*640px. The latency reported is the combination of object detection and relation prediction stages for all models. In Figure 5.4 we display the difference in latency and F1@k for the five different models. Using YOLOV8, we experienced an average improvement of 62.17% in F1@k compared to Faster-RCNN. This improvement is obviously due to the overall gain of the accuracy of YOLOV8m in object detection, which performs at 35.1 mAP@50 versus 17.4 to 27.2 for Faster-RCNN on the IndoorVG dataset, see Table 5.2. We computed an average of 67.62% improvement in mAP@50 for all models, which is different

[2]For more information, please consult `https://pytorch.org/docs/stable/notes/randomness.html`

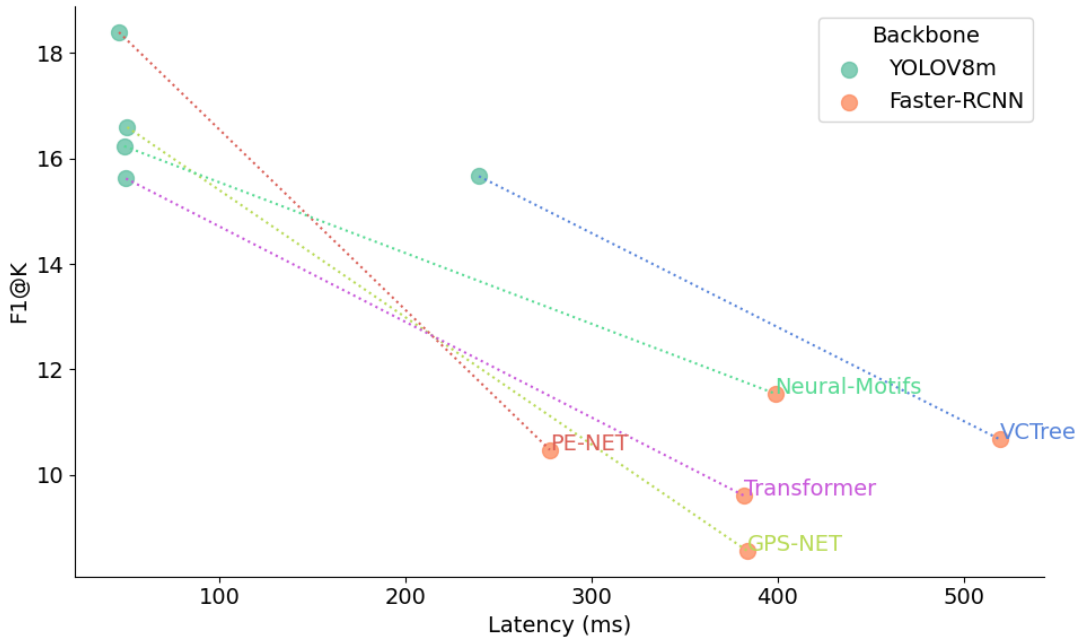


Figure 5.4: Latency versus F1@k for different models equipped with the Faster-RCNN backbone and our modified YOLOV8 backbone.

from the gain in F1@k which shows that there is not a strong correlation between mAP@50 and F1@K performance in general. In fact, we measured a Pearson correlation coefficient of 0.70 between the two gains with a p-value of 0.19, which is not significant under the condition $p < 0.05$. Regarding latency, we observed an average improvement of 77.82% for using YOLOV8 instead of Faster-RCNN, which is a considerable gap. The difference in latency between the five models tested is also lower with YOLOV8 than Faster-RCNN, this is due to the step of object classification which has been removed in our implementation. As relation heads do not need to compute object classes, computational complexity becomes lower and their average execution time for relation prediction becomes very similar. The best model in F1@K, PE-NET [49], is also the fastest one with a latency of 48.5ms which is a good choice for real-time constraints.

As shown in previous chapters, the meanRecall@k and Recall@k metrics do not efficiently represent the performance of a model to produce useful graphs for real-world applications. We used the InformativeRecall@K metric introduced in Section 4.4 with $k = [5, 10]$ to evaluate the impact of YOLOV8 on the performance of SGG models. Here we purposely chose low values of k as the main goal of this metric is to represent the ability of models to output informative predictions with confidence, such that the top predictions can be directly sampled for reasoning. We assume that in real-world applications a few high-quality relations are more beneficial than a high number of medium to low-quality relations to describe a scene. We compare our approach with YOLOV8m and traditional Faster-RCNN in fig. 5.5, the metric reported is the average of

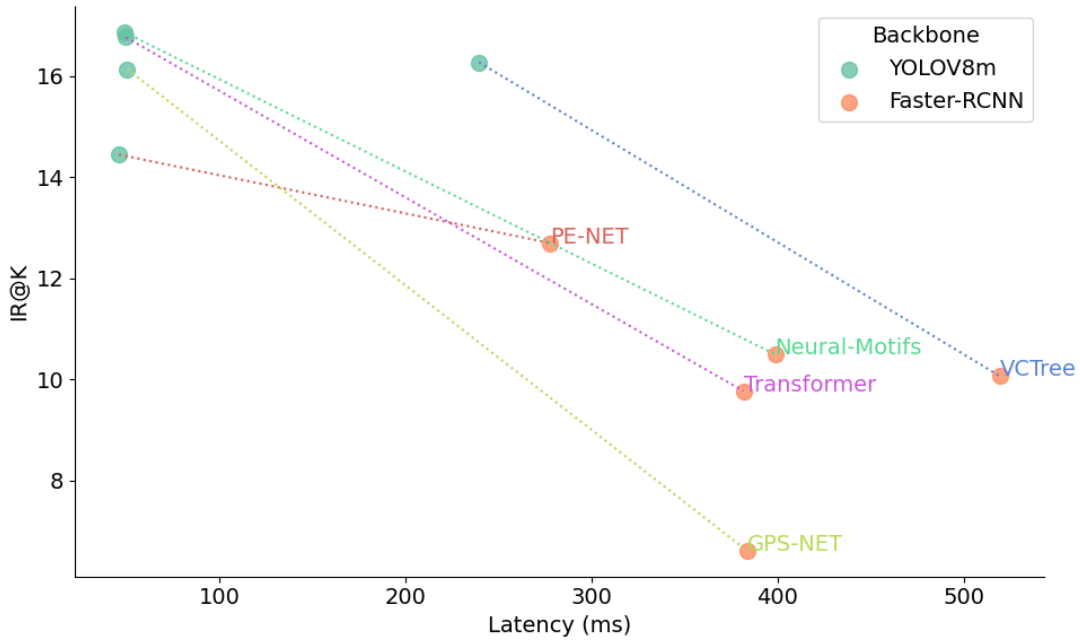


Figure 5.5: Latency versus IR@k for different models equipped with the Faster-RCNN backbone and our modified YOLOV8 backbone.

IR@5 and IR@10. Surprisingly, we observe a net improvement using YOLOV8 for all models, with an average improvement of 62.08%. In contrast to F1@k, the gain in IR@K is strongly correlated to the gain in mAP@50 with a Pearson correlation coefficient of 0.93 and a p-value of 0.02, which is significant under the hypothesis $p < 0.05$. This is a very interesting result as it shows that the quality of the object detection backbone has a direct impact on the quality of the relations predicted by the model. The best model overall seems to be PE-NET with the better trade-off between F1@k and IR@K and the lowest latency. This result is consistent with the performance of the model reported by authors on the VG150 dataset in their original paper [49]. In the next sections and chapters, we will use the PE-NET model for all our experiments.

5.3.2 Scaling YOLOV8

The results that we obtained with the YOLOV8 object detector clearly show that object detection has a consequent impact on the performance of relations predictions. However, it is still unclear if this improvement comes from the quality of the features extracted by YOLOV8 or the performance of the regression and classification heads of YOLOV8. To investigate this issue, we decided to run a last experiment by scaling down and up the YOLOV8 model with its different variants. Similarly to YOLOV8-medium, we trained the nano, small, large and x-large variants with the same hyperparameters as before on the IndoorVG dataset. In Table 5.4 we first display the key differences between every variant. The YOLOV8-Large is the base model

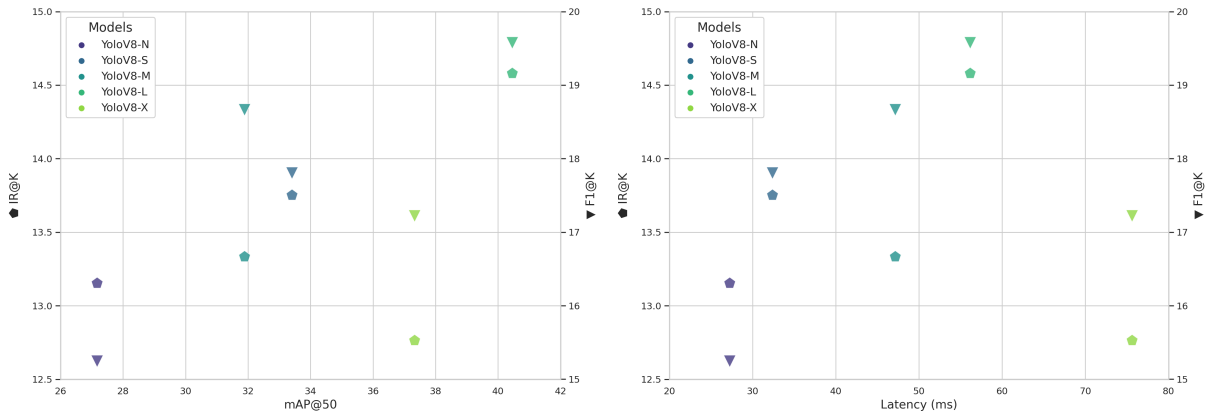
Variant	Width	Depth	Channels	Params	$\mathbf{mAP^{50}}$	$\mathbf{mAP^{50\text{-}95}}$	Latency
Nano	0.33	0.25	64	3.2	26.6	14.5	11.46
Small	0.33	0.50	128	11.2	32.8	18.2	11.94
Medium	0.67	0.75	192	25.9	31.2	17.1	14.33
Large	1.0	1.0	256	43.7	39.8	22.9	16.80
X-Large	1.0	1.25	320	68.2	36.7	20.6	22.49

Table 5.4: Scales of the different YOLOV8 variants and their respective performance on the IndoorVG dataset.

and then every variant scales in depth or width from this model. The depth scale is the number of layers in the model, implemented as the number of repeated Conv layers inside each Pyramide block (the C1-C5 and P3-P5 blocks depicted in Figure 5.2). It is important to notice that the overall architecture of the network stays the same for all variants. Due to differences in width and depth, the number of out channels for the Neck of YOLOV8 is different from every model. This corresponds to the number of dimensions that the features extracted from the backbone are reduced to, before being fed to the Head of YOLOV8 and the relations prediction stage in our case. For comparison, the number of out channels in Faster-RCNN is 256, which corresponds to YOLOV8-Large. Results in mAP and latency are also displayed in Table 5.4. Surprisingly, the performance in mAP@50 and mAP@50-95 is not increasing as the size of the model increases. We observe a slight decrease in accuracy from the medium to the small version and a similar effect from the x-large to the large version. This could be due to the bad quality of IndoorVG bounding box annotations or the small number of images available for training (11,433 images). On the other hand, the latency does increase linearly with the size of the model, which is consistent with reported metrics by the authors of YOLOV8 [3].

To evaluate the impact of the model size on the relation prediction stage, we ran a set of experiments using the PE-NET model [49] for SGG. Here, we trained the model with the same hyperparameters as before, only modifying the input size of the visual features used for relation prediction to match the different output channel sizes for each YOLOV8 variant. For each model, we measured the InformativeRecall@K with $k = [5, 10]$ and averaged the results in the IR@K metric. For the F1 score, we measured the F1@K with $k = [20, 50, 100]$ and averaged the results in the F1@K metric. The results are displayed in Figure 5.6a. In this figure, we can observe no correlation between the mAP@50 of the different object detector backbones and the overall performance of the PE-NET model, for both F1@K and IR@K. Specifically, the Pearson correlation test gives a p-value of 0.141 for the correspondence between mAP and F1@K and 0.390 for the correspondence between mAP and IR@K, which is not statistically significant under the $p < 0.05$ hypothesis. While the nano and large versions have the worst and better performance in both F1@K and IR@K, the other models with small, medium, and x-large

[3] See `https://docs.ultralytics.com/models/yolov8/#performance-metrics`, accessed on 12/08/2024



(a) (b)

Figure 5.6: Average F1@k and IR@k performance for the PE-NET model [49] on relation prediction against the corresponding mAP@50 for object detection using different variants of the YOLOV8 model.

variants seem to have a random performance with respect to their mAP. In Figure 5.6b we also display the latency of the PE-NET model for the different YOLOV8 variants. We observed a linear increase in latency with the size of the model, which is consistent with the results of the object detection stage. This is an important finding as it shows that the performance of the relation prediction stage is not directly correlated with the performance of the object detection stage. We hypothesize that the out-channel sizes of the features extracted from the backbone are also an important factor in the relation prediction stage. Regarding the underperformance of YOLOV8-X, we believe that the higher number of dimensions in the features extracted from the backbone could lead to more difficulties for the relation head to converge.

In terms of the overall performance, the large versions seem to be the best. It has a good trade-off between IR@K, F1@K, and latency, which is important for real-time applications. However, the latency is still slightly above our desired threshold of 50ms with a value of 56ms on average. To address this problem, we decided to run a new set of experiments on another variable that could impact the performance of the relation prediction stage: the number of proposals per image.

5.3.3 Candidate Selection: Quadratic Complexity

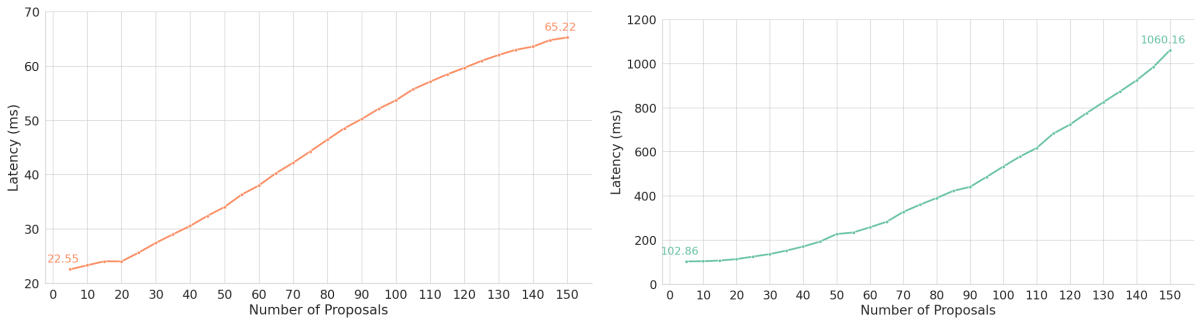
During the step of relation prediction, all proposals detected by the object detector are considered valid node candidates for the graph refinement process. To not blow up the memory usage during training and inference, previous works [12], [37] have chosen to keep only a fixed number of proposals (i.e. 80 proposals) from the higher confidence proposals predicted by the object detector. The idea here is that by sampling enough proposals, it is easier to find valid pairs

and relations. However, 80 proposals per image is unrealistic and will considerably enlarge the number of computations required to generate a good graph, which is a bottleneck for real-time applications. In fact, in the Context Learning step (Figure 5.1**(i)**) strategies such as Iterative Message Passing [12], [47] are used to learn inter-dependencies between nodes. This type of learning is scaling in terms of computational complexity quadratically with the number of nodes in the graph, as the model needs to learn the relation between all possible pairs of nodes. This issue also raises concerns about the objective of the task, as it could be considered unfair to look for candidate nodes of the graph based on their likelihood of having a relation rather than the true confidence of the object detector. For instance, a low-confidence bounding box `chair` (which could be ranked bottom 80 in the detected proposals) could be selected as a node candidate in the final graph instead of a higher-confidence bounding box `kite` if the two objects are in the presence of a third one, `table`, as `chair` is more likely to have a relation with `table` than `kite`. This paradigm will tend to improve the performance of models for relation prediction but with the drawback of lowering the accuracy in object detection and limiting the generalization to new or unseen relations. This could be one explanation of the phenomenon observed in Table 5.2. The question here is: *do we want models to predict relations from a small set of highly confident objects in the scene or do we want models to predict the most likely relations from a larger set of objects, even though some of those objects are of low confidence?* Outside the real-time constraint, we believe the first approach is more beneficial for real-world applications and extends to the usage of SGG on the edge. We also hypothesize that the gain in latency is more important than the gain in accuracy by using a smaller number of proposals, such as 10 or 20 per image and that an optimal trade-off can be found. For the relation prediction stage, as we are evaluating models to predict entire triplets and not solely the predicate, the relation triplets score is computed with the following formula:

$$\theta_{rel} = \theta_{obj} * \theta_{pred} * \theta_{subj} \tag{5.1}$$

With θ_{pred} being the confidence score of a predicate given $\langle subject, object \rangle$ pair as candidate and $\theta_{obj}, \theta_{subj}$ are the respective confidence score of the object detector. This formula gives more weight to the confidence of the subject and object than the predicate, which makes the model's overall performance rely more on the object detector than the relation predictor. Thus, one would rather want to input only highly confident proposals to the relation prediction stage. There could be a maximum of $n * (n-1)$ possible pairs in the graph, as a result when doing matching the computational complexity is supposed to scale accordingly. To demonstrate this hypothesis, we evaluate the performance of the PE-NET model [49] in both latency and accuracy for different numbers of input proposals, ranging from 10 to 150 per image, with a step of 10. For all experiments, we ranked proposals by confidence of the backbone after a step of Non-Maximum Suppression (NMS) and selected the top n ones.

We display the results of those experiments in fig. 5.7. By using 150 proposals with YOLOV8,



(a) (b)

Figure 5.7: Latency for the PE-NET model [49] with YOLOV8 backbone (a) and Faster-RCNN backbone (b) using a different number of proposals per image, with batch size = 1.

the latency is 65.22 while the same number of proposals with the Faster-RCNN backbone will result in a latency of 1060.16. We also observe a 3x gain from 10 to 150 proposals with YOLOV8 whereas, for Faster-RCNN, the gain is 10x. We believe this is due to the label decoding step added in the PE-NET model with Faster-RCNN backbone, which adds a significant overhead to the model. The latency of the Faster-RCNN-based model is also significantly higher than our upper bound for real-time inference, even with only 10 proposals. However, when using YOLOV8, we can find a satisfying latency (i.e. $>$ 50ms) with 90 proposals and below. To find the best trade-off between latency and performance, we also measured the average F1@k for different numbers of proposals, see fig. 5.8. In this figure, we can see that for both models, an optimal F1@k (left axis, plane line) is obtained at around 40 proposals per image. This means that the top 40 proposals returned by YOLO or Faster-RCNN are of good enough quality to be valid node candidates for the graph refinement step. In the same plot, we also measure the average IR@k for the same set of settings (right axis, dotted line). Interestingly, we observe a high IR@k for the lowest number of proposals, 10, even if the average F1@k is not maximal. These results confirm the assumption that to obtain informative graphs, the quality of object proposals is of utmost importance. By crossing those numbers with latency per number of proposals (see fig. 5.7), we can find a threshold of $n = 40$ for an optimal implementation for F1@k and $n = 18$ for an optimal F1@k to IR@k trade-off for both YOLOV8 and Faster-RCNN.

This concludes our experiments on the Feature extraction and object detection components of the SGG pipeline (see the first stage in Figure 5.1). However, as soon as latency is involved, we believe that further improvements can be made by also optimizing the relation prediction stage (the second stage in Figure 5.1). In the next section, we will tackle this problem by proposing a new method to extract relation features from the backbone and reduce the complexity of the relation prediction stage for the PE-NET model.

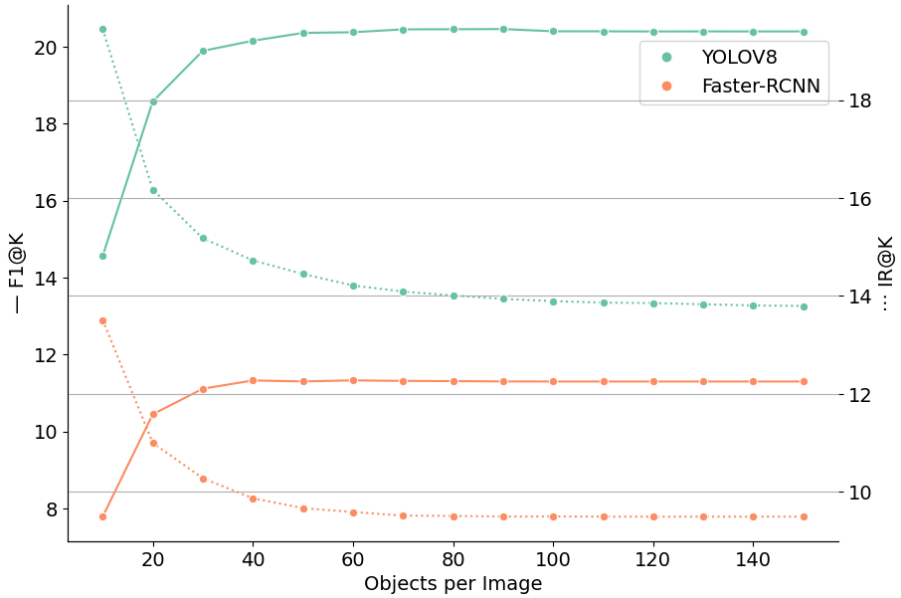


Figure 5.8: Average F1@k and IR@k performance for the PE-NET model [49] using a different number of proposals per image, with batch size 1.

5.4 Relation Prediction

The feature extraction and object detection modules were the most obvious and easiest components to optimize in the SGG pipeline. Taking the PE-NET as a case study, we now focus on the relation prediction stage. During the relation prediction stage three steps are performed in a sequence: (1) features refinement to generate node and edge representation, (2) context learning, and (3) final decoding of relations and object classes. In this section, we will focus on steps (1) and (2), which are the most computationally expensive and the most critical for the overall performance of the model.

5.4.1 Features Refinement

In most SGG implementations, the relation prediction stage takes as input visual features for each bounding box (which serve for subject and object representation, see Figure 5.1**(f)**) but also visual features aggregated from each $\langle subject, object \rangle$ pairs of bounding boxes (which will be used later on as the internal representation of the relation, see Figure 5.1**(d)**). These union features are usually aggregated with spatial features to form the edge representation in step **(g)** of the pipeline. However, the union features are computed using a second step of RoI Align, which is computationally expensive (see Figure 5.1**(d)**). We could remove this step and instead directly merge the features from every possible pair of subject and object after the first RoI Align step (Figure 5.1**(f)**), which would save a significant amount of computation. Another possibility is to

Head	IR@K	F1@K	Latency (ms)	Params (M)
Baseline	14.13	20.03	45.26	151.8
Union	14.25	21.05	42.61	64.18
Spatial	14.40	20.67	20.89	55.42
Spatial+Union	14.35	19.03	57.41	65.23

Table 5.5: Performance on SGG of the PE-NET model [49] using different feature extraction methods for relation prediction.

remove the use of union features, and create the edge representation based on spatial features only. The hint here is that the visual representation of the relation is already embedded in the visual features of the subject and object nodes, which are used to predict the relation. To test this hypothesis, we ran a set of experiments with the PE-NET model [49] using different features as input to the relation prediction stage. We tested the use of union features, spatial features and a combination of both for the relation prediction stage. For these experiments, we also reduced the number of dimensions of the node and edge features. In the original implementation, node and edge features are upsampled 8x and 4x respectively to a fixed size of 4096 by a feed-forward layer (see steps **(g)** and **(h)** in Figure 5.1). We believe that this is not necessary and that the model will perform similarly without upsampling. In fact, 256 and 512 dimensions are already enough information to represent the visual features of the subject, object, and relation. The results from these experiments are displayed in Table 5.5. First, we can observe that by not upsampling the features we do not lose performance, but we gain a significant amount of latency, compared with the baseline used in previous sections. More interestingly, we observe a small gain in F1@K and IR@K by not upsampling features. In terms of the features extraction, we observe that the use of spatial features only is the best choice for the PE-NET model, as it has a lower latency while keeping a similar performance in F1@K and IR@K than other settings. Interestingly, by combining spatial and union features with no upsampling (last row of Table 5.5), we observed a decrease in performance in F1@K and IR@K, which shows that upsampling is beneficial in this particular case to take advantage of the large amount of information extracted.

5.4.2 Prototype Embedding Network

The Prototype Embedding Network model (PE-NET) [49] is a state-of-the-art model for SGG. It is based on the idea of learning prototypes for each predicate class in the dataset and using those prototypes to predict relations between objects. Entity prototypes are also used to model the node representation and learn efficient node-edge representation. Entity (or node) prototypes are formed using a combination of linguistic features (using GLoVe word embeddings [46]) and the visual features extracted from the ROI Align step. Next, a gate mechanism is employed to remove class-irrelevant information from the representation. To compute the edge prototype,

the prototype representation of the subject and object are merged. Specifically, in the original representation the fusion is done as shown below:

$$ReLU(s + o) - (s - o)^2, \tag{5.2}$$

with s and o being the corresponding prototypes for subject and object. Then, this representation is combined with the union features and spatial features obtained in step **d** and **e** of the pipeline. Finally, a gate mechanism is employed to select relevant features from the multi-modal fusion. We proposed to modify this step in two ways: first, we added a linear layer to replace the fusion of subject and object prototype (Equation (5.2)). Second, we removed the union features for the edge representation, as demonstrated previously union features are of no use for the relation prediction stage of PE-NET. Next, prototype-guided learning is applied as in the original work [49]. Prototype-guided learning will consider a loss function as the combination of cosine and Euclidean distances to push away dissimilar prototypes and pull closer similar ones. The final loss function is defined as:

$$\mathcal{L} = \mathcal{L}_{r\_cos} + \mathcal{L}_{e\_cos} + \mathcal{L}_{r\_euc} + \mathcal{L}_{e\_euc}, \tag{5.3}$$

with $\mathcal{L}_{r\_cos}$ and $\mathcal{L}_{e\_cos}$ being the cosine distance between relation prototypes and edge prototypes and $\mathcal{L}_{r\_euc}$ and $\mathcal{L}_{e\_euc}$ being the Euclidean distance between relation prototypes and edge prototypes. We ran a set of experiments with the PE-NET model using the modified edge representation and removing the union features. The modified PE-NET architecture is displayed in Figure 5.9. As we are now performing object class prediction in the object detection stage, we can remove the object class prediction head in the Scene Graph Prediction module, which also saves some computation (see Figure 5.1**k** and Figure 5.1**(l)**). The edge representation refinement (see Figure 5.9**(g)**) is now dependent on the node representation refinement (see Figure 5.9**(h)**), in contrast to the baseline implementation (see Figure 5.1**(h)**).

The results of new experiments with this architecture are displayed in Table 5.6. We observed a slight improvement in F1@K and IR@K by using the modified edge representation, which shows that the original fusion of subject and object prototypes is not optimal for the task. We also observe a significant decrease in latency by removing the union features, which is consistent with the results of the previous section. This is a very important result as it shows that the relation prediction stage can be optimized to be more efficient and faster without losing performance. When we compare the final model with the original Faster-RCNN implementation, we see a consequent improvement in performance but also in latency, with almost a 10x gain in latency. This is surely exciting news for the SGG community and especially the robotics community, as it shows that the task can be performed in real time with a good trade-off between accuracy and latency. We also observe a significant decrease in the number of parameters and GFLOPS, with

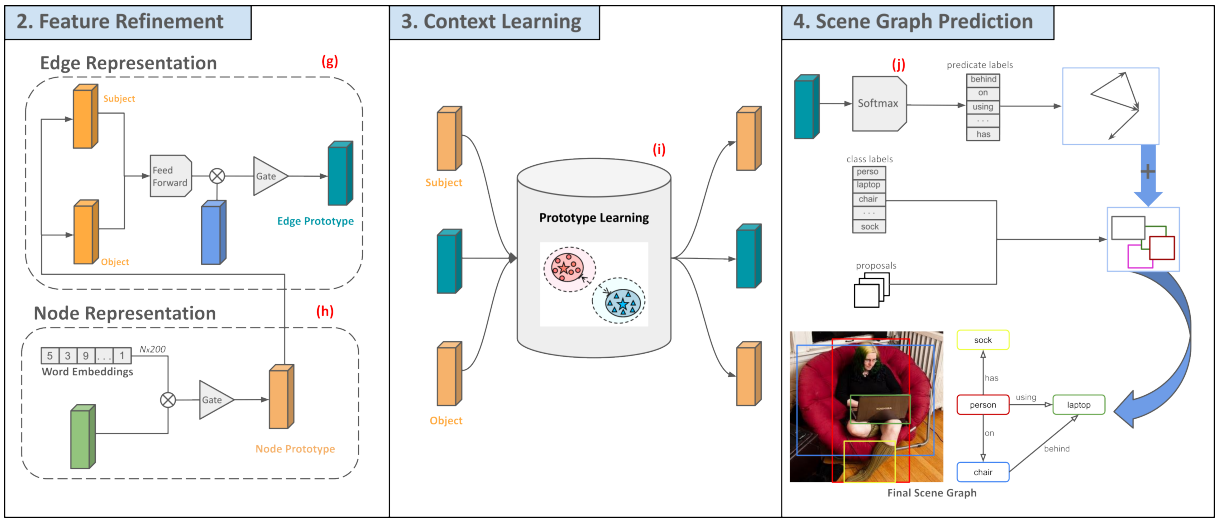


Figure 5.9: Modified PE-NET model for SGG.

Backbone	Head	IR@K	F1@K	Latency (ms)	Params (M)	FLOPs (G)
Faster-RCNN	PE-NET	10.7	10.47	277.62	425.99	2218.91
YOLOV8-L	PE-NET	14.13	20.03	45.26	151.8	164.42
YOLOV8-L	M-PE-NET	14.75	21.48	29.36	65.75	140.05

Table 5.6: Performance of the modified PE-NET model for SGG on the IndoorVG dataset.

15x less GFLOPS require for our Modified PE-NET than the original model. This demonstrates that the model is more efficient and can be deployed on edge devices.

5.5 Discussion

Even with improved latency and performance, the Real-Time SGG task is still far from being solved. In this section, we discuss the limitations of SGG models and the challenges that remain to be addressed for their deployment on edge devices.

5.5.1 Performance in SGG

The performance of SGG models that we obtained in our experiments is not very convincing. For the Recall@K or meanRecall@K, the best model we have tested does not reach more than 27% in Recall@100 and 18% in meanRecall@100. This means that out of the top 100 relations predicted, merely a fifth of them are correctly identified on average for each class (meanRecall metric). If we put this information in perspective with other similar tasks such as VQA (VQA), state-of-the-art models in VQA attain today more than 70% accuracy on traditional benchmarks. Of course, this performance has to be mitigated by the fact that the task of SGG is bounded by the performance of the object detector, which is not the case for VQA. However, this is still a significant gap in performance that needs to be addressed. As we have seen, a significant increase in the performance of the object detector does not necessarily translate into a similar increase in performance for the SGG model. This is due to the inherent complexity of the task, which can be pinned to one major issue: the intrinsic polysemy of natural language. In Figure 5.10 we display the confusion matrix of the PE-NET model for relation prediction on the IndoorVG dataset. We can see the confusion created by the predicate "on" which is highly polysemous. This issue has been pointed out by multiple previous works [18], [146] under the umbrella of long-tail learning. As predicate annotations are very sparse for some confused classes (such as *laying on* or *mounted on*), disambiguation can be proposed by artificially boosting the performance on those tail classes. This can be done by using the Logit Adjustment method [147] or other balance adjustment strategies [148], [149]. These methods can significantly boost the meanRecall performance for SGG, however, their potential for real-world deployment is still to be evaluated. In fact, there is often a significant shift between in and out-of-distribution data in Computer Vision [150], which can lead to a decrease in the performance of the model when evaluated on different data. Due to the sparse annotation of SGG datasets, we argue that this bias can be even more important in the case of SGG models. When used in real-world settings, SGG models will tend to predict true spurious relations or invalid fine-grained relations most of the time. We display two examples of the M-PE-NET model predictions in Figure 5.11. In the first prediction (b), we see that most of the relations predicted are correct, but spurious. The object

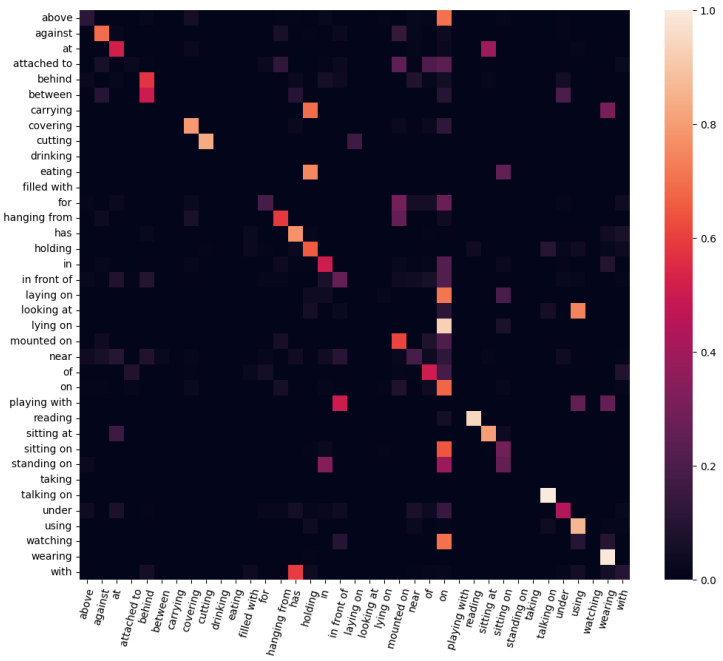


Figure 5.10: Confusion matrix for the M-PE-NET model with YOLOV8-L backbone on the IndoorVG dataset.

detector backbone is also not able to detect a wide range of objects in the scene. In the second example (d), the model is predicting $\langle person, sitting\ at, counter \rangle$ which is not correct but very likely to be true given the other relations (specifically the relations $\langle person, holding, bottle \rangle$ and $\langle bottle, on, counter \rangle$) as well as language priors. We hypothesize here that the visual features are not taken into account to predict this relation, leading to failure. This shows that the task of SGG is still far from being solved and that more research is needed to address the issue of generalizing to out-of-distribution data.

5.5.2 A Multi-Modal Problem

As we have seen in the previous sections, not all modalities involved in the relation prediction stage play a significant role. We pointed out the usage of the Union features for the relation representation which seems to confuse more the PE-NET model than it helps it to converge. This addresses the question of which modality plays the biggest role in the context learning of SGG models. As we have seen, a lot of issues in the performance of SGG models come down to the ambiguous nature of natural language, it is then possible that the linguistic features extracted from the text embeddings are the most important modality for the task. In a new experiment, we decided to change the PE-NET model one last time by removing all dependence on visual features. This means removing the visual features and gate operation in step (**h**) (see Figure 5.9)

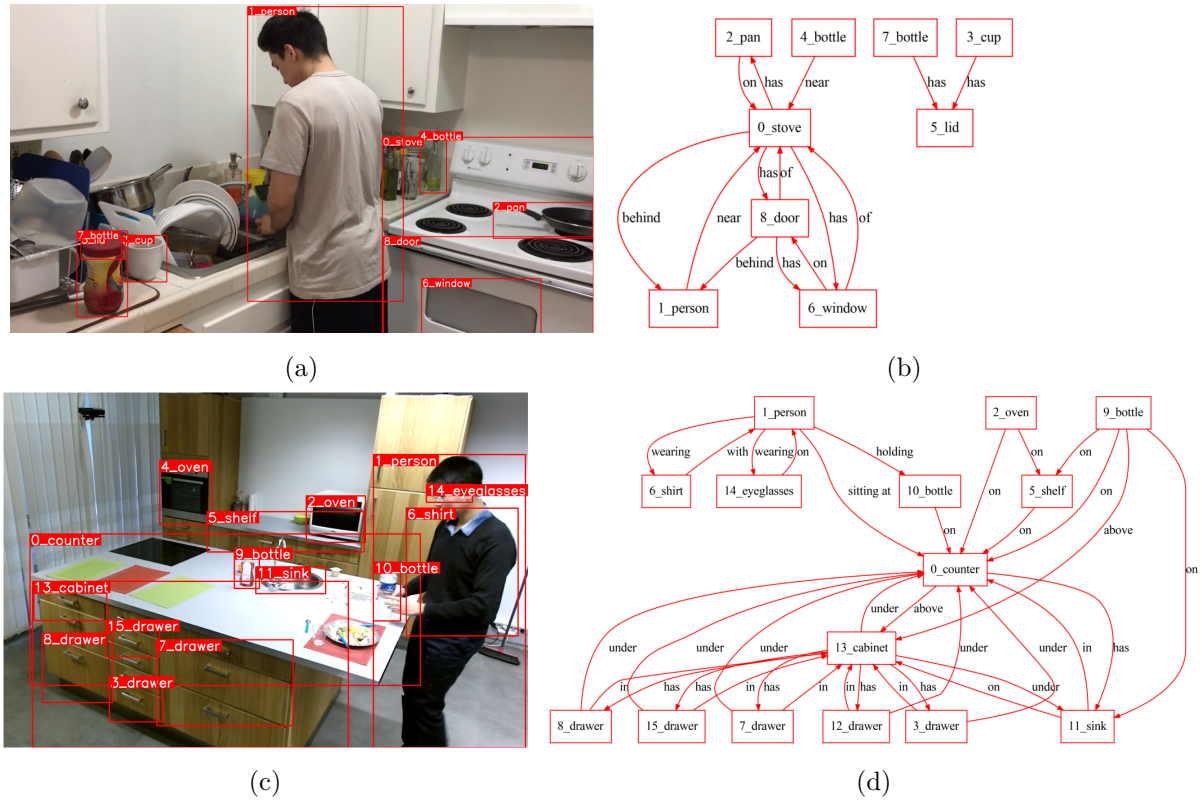


Figure 5.11: Examples of SGG in real-time with the YOLOV8-L object detector and the PE-NET model.

Model	IR@K	F1@K	Latency (ms)	Params (M)
M-PE-NET	14.75	21.48	29.36	65.75
M-PE-NET w/o Visual	14.33	19.21	17.84	55.94

Table 5.7: Performance of the modified PE-NET model (M-PE-NET) with no visual guidance on the IndoorVG dataset.

as well as removing the spatial features and the gate operation in step (**g**). By doing so and re-training the model with the same hyperparameters as before, we obtain a performance of 19.21 in F1@K and 14.33 in IR@K (see Table 5.7) which is very close to the performance of our modified PE-NET model with visual features. By removing the dependence on visual features, the latency drops significantly to a low 17.84ms per inference, taking into account that YOLOV8 alone takes an average of 12.43ms, this means that the relation prediction stage takes only 5ms. This can of course benefit the real-time implementation of SGG models but with the disadvantage of losing all possible generalizations to out-of-distribution images. This shows the enormous bias of the PE-NET model towards linguistic features and the misuse of visual features for the task. Our findings are consistent with previous work which has shown that SGG models can attain good performance without visual guidance [26]. However, not using any visual features is not profitable for the task, and especially for generalization on out-of-distribution data. A model without visual information will not be able to predict rare relations and will rely solely on statistical information from the training data. In the past few years, a consequent amount of work has been proposed based on the same two-stage architecture proposed in 2019 which, by its structure, helps this bias to happen. We believe that a new architecture that does not use linguistic features and a better fusion of spatial and union features could be more efficient for the task. In fact, using linguistic features helps the model to converge very quickly but does not necessarily help it to generalize to new data and may overfit the training data distribution easily.

5.6 Concluding Remarks

In this chapter, we proposed the first implementation of a true real-time SGG procedure by leveraging the YOLOV8 object detector. Our main finding is that the performance and latency of SGG models can be greatly improved by optimizing the feature extraction and object detection steps. However, the performance of SGG models is not strictly correlated with the quality of objects detected by the object detector, as greater mAP does not necessarily imply greater F1@K or IR@K performance for the task. We also found that the number of proposals used as input to the relation prediction stage is a critical parameter for the performance of the model, for both latency and accuracy. We observed that the quality of the proposals is more important than the

quantity of proposals for the task. An optimal trade-off between latency and accuracy can be found by using 40 object proposals as input to the relation prediction stage. Using the IR@K metric, we observed that a low number (i.e. 10 or below) of high-quality proposals produces a more informative graph even though relations for this number of proposals are deemed of less quality by standard metrics (F1@K). This last point illustrates the limits and challenges of current evaluation metrics and benchmarks used in the task which may not be appropriate to evaluate real-world use cases of SGG.

Furthermore, we conducted additional experiments on the relation prediction stage of the PE-NET model and found that the use of union features is not beneficial to form the initial edge representation. By removing union features and optimizing the PE-NET model, we obtain a final performance of 21.48 in F1@K and 14.75 in IR@K with a latency of 29.36ms per inference on our IndoorVG dataset. This is an improvement of +105.02% in F1@K and +37% in IR@K compared to the original PE-NET model with Faster-RCNN trained on the same dataset with similar hyperparameters. Compared to the original model, the gain in latency is 955.17%, going down from 277ms to 29ms which is close to the performance of the object detector alone. We also found that the PE-NET model is highly biased towards linguistic features and that the use of visual features is not beneficial for the task. By removing all visual features from the PE-NET model, we obtain a performance of 19.21 in F1@K and 14.33 in IR@K with a latency of 17.84ms per inference. This shows that the PE-NET model, and potentially other models in the SGG task, still possess today severe limitations for their deployment in real-world applications.

In summary, in this chapter, we successfully leveraged the state-of-the-art real-time object detector YOLOV8 as a feature extraction backbone for the task of SGG, and we proposed a new implementation that achieves a competitive runtime latency of 29.36ms with an improved performance of 62.17% in average on a large set of baseline models. Our approach is generic and can be easily implemented in any two-stage SGG architecture to boost performance and lower latency. This approach can be specifically beneficial for real-time constraint applications such as during Human-AI collaboration in Activity of the Daily Life (ADL). We believe that our approach for real-time SGG can be used to power the representation capabilities of autonomous agents in these scenarios and enable them to understand and interact with the environment in a more human-like way. In the next chapter, we will introduce a new architecture which takes advantage of Scene Graphs to power the internal representation of the world of an autonomous agent. We call this new type of representation *Continuous SGG*.

Chapter 6

Continuous SGG

> We are drowning in information but starved for knowledge.
>
> John Naisbitt

Autonomous AI agents evolve in a dynamic world where the environment is constantly changing. Representing such changes in a structured and comprehensive manner is a long-lasting challenge in the field of Artificial Intelligence. In this chapter, we aim to tackle this issue from the perspective of compositional relations. Compositional relations can be aggregated into Scene Graph representations thanks to the task of SGG. In previous chapters, we tackled the task of SGG from still images. However, in the context of autonomous agents, the environment is not static and the agent should be able to reason over time. In this chapter, we introduce a new paradigm for SGG in the context of autonomous agents. We propose a new representation based on Scene Graphs that is continuously updated over time and that serves as the internal abstract memory of the agent, i.e. a symbolic *World Model*.

Scene graphs are currently not tailored for temporal reasoning. Current approaches for this purpose are limited because they replace the entire graph with new predictions, potentially erasing previous detections that could still be valid [30], [151]. This is not suited for an autonomous agent as it does not memorize past relations. Here, we define a new Continuous SGG paradigm for autonomous agents: given generated scene graphs from each timestamp t, we update a single *Global Scene Graph (GSG)* representation that encompasses all *informative* and *plausible* relations gathered by the agent on the visual environment from timestamp t_0 to t_n. Here, *informative* refers to our definition of informativeness in Chapter 4. In addition, we define *plausibility* as the likelihood of a relation existing in the real world, with respect to past relations in the graph. As we are updating the relations in our *Global Scene Graph* with new predictions at each new timestamp, some relations can be inconsistent with each other. For instance, a new relation can be predicted stating that $\langle person\_2, is\ sitting\ on, chair\_1 \rangle$ when another relation $\langle person\_1, is\ sitting\ on, chair\_1 \rangle$ already exists in our representation. This is not plausible in the real world, as it breaks basic common sense and needs to be handled before updating the representation to avoid wrong interpretations of the scene. To tackle this issue, we applied *Constraint Optimization* [152] to our representation via a set of commonsense rules to ensure

consistency.

Our *Global Scene Graph (GSG)* representation serves as the internal memory of the agent and is used for further reasoning. Downstream tasks include Activity Recognition [153], Automated Planning [154], or even HRC [29]. In this chapter, we evaluate our approach on the task of Activity Recognition and Automated Planning for Symbolic Learning by Demonstration. For Activity Recognition, we provide a learning-free approach that can predict human activity from our Global Scene Graph representation, without the need for any visual features. For Automated Planning, we presented a new approach to generating planning domains from our Global Scene Graph representation. We show that our approach can generate a PDDL [40] description from our Global Scene Graph representation with no further processing. Activity Recognition and Automated Planning are two key tasks in HRC. Our approach is based on a simple scenario where a human is conducting various Activities of the Daily Life (ADL) in a domestic environment with the help of an autonomous robot. However, in contrast to previous work, we do not assume any specific knowledge of the environment or the human. Our approach is based on the continuous learning of the environment and the human activities through time. Our system needs to be able to (1) detect the activity being performed after a few demonstrations and (2) generate a comprehensive plan of the actions which composed the activity to be able to reproduce them. To achieve this goal, we will use objects and relations detected in the scene to generate a symbolic representation of the environment and the human activities.

This chapter is organized as follows: in Section 6.1 we review the state-of-the-art on symbolic representations in robotics, their limitations, and the need for a new paradigm. Then, we review the latest trends and popular approaches for Video SGG and their challenges for real-world applications. In Section 6.2 we describe the modifications done to the previously introduced SGG architecture (see Chapter 5) to adapt it to videos. Next, in Section 6.3 we present our Global Scene Graph (GSG) representation and the rules we defined to ensure common sense consistency. In Section 6.4 we present a quantitative evaluation of our approach for the task of Activity Recognition. Finally, in Section 6.5 we present a qualitative evaluation of our approach for the task of Automated Planning for Symbolic Learning by Demonstration.

6.1 Related Work

While the task of SGG aims at representing all kinds of relations, their applications to robotics are mostly tied to spatial relations [67], [68]. These approaches are modeling spatial relations between objects to help autonomous agents navigate the environment. On the other hand, Video SGG is a new task that aims at generating scene graphs from a sequence of images. In the next sections, we review related work on temporal reasoning and knowledge representations in robotics, as well as the latest trends in Video SGG and the challenges to adapt these approaches

to robotic constraints.

6.1.1 Temporal Reasoning and Knowledge Representations in Robotics

Temporal Reasoning. Temporal reasoning have been used in robotics for a long time. From the early works of Allen [155] to the more recent work of Beetz et al. [156] on knowledge management, temporal reasoning has always been a key aspect of robotics. Theoretical methods include different types of temporal logics such as Allen's Interval Algebra [155], the Situation Calculus [157] or the Event Calculus [158]. These methods are based on a set of rules that define the temporal relations between events which are traditionally handcrafted. Implementations of temporal reasoning in robotics is often associated to task planning. The usage of Answer Set Programming (ASP) [159] or PDDL [40] for planning in robotics is popular. These methods are used to generate plans for autonomous agents to perform tasks in the environment. However, these methods assume a complete knowledge of the environment and the tasks to be performed. They are not suited for real-world applications where the environment is dynamic, and the tasks are not known in advance.

Knowledge Representations. Temporal reasoning requires extensive knowledge of the environment, which is usually represented in a structured manner. Ontologies and taxonomies are popular methods to represent knowledge in robotics [160]. Popular ontologies include the RoboBrain [161] or the KnowRob [71], [72] projects. These projects aim at representing knowledge in a structured manner to be used by autonomous agents. However, these methods are not easily generalizable to new data and are not robust to new scenarios and applications. They require a lot of manual work to define the rules and cautious updates to adapt to new data. Semantic graph and scene graph representations have gained popularity in robotics in recent years through their applications in autonomous navigation [162], [163] and object manipulation [68]. In these representations, relations are defined by spatial properties between objects and agents, often in a 3D space. These representations do not represent other types of relations such as functional or attribute and are thus limited in their applications. A second challenge of traditional Knowledge Representation in robotics is the acquisition of knowledge [164]. Each robotic platform has its own sensors and perception system, which makes it difficult to transfer knowledge from one platform to another and generate the same representation. In this chapter, we present a new approach to knowledge representation and reasoning in robotics based on Multi-Layer Scene Graph representations (the different layers corresponding to **Topological**, **Functional**, **Part-Whole**, and **Attributive** relations) that are continuously updated over time. In contrast to 3D scene graphs, our approach uses only 2D images from RGB sensors which can be deployed in almost any robotic platform. This representation combine the flexibility of deep learning methods through an SGG backbone and the robustness of symbolic reasoning through commonsense rules and heuristic methods.

6.1.2 Video SGG

Models. When generating scene graphs from a video flow, a complex structure is needed to model relationships between images. The task of Video SGG was initiated by Shang et al. [165] with a first approach at modeling relations through time as object tracklets and predicate. In contrast to object bounding boxes, object tracklets represent the trajectory of an object in the video flow. Relations are typically predicted by frames and then aggregated to corresponding tracklet pairs over the video sequence. Subsequent approaches, such as VSGG-Net will take inspiration from this architecture and the current work in classical SGG [38] to propose a context learning module to refine the graph representation before the final classification, improving efficiency. Instead of a separated context learning module, Wang et al. [166] uses a Temporal Convolution Network (TCN) paired with a Graph Convolution Network (GCN) for modeling within-image dependencies to create coherent stories across video clips. Other works include the Target Adaptive Context Aggregation (TRACE) architecture [31]. Here, the authors use a temporal fusion module to relate information from consecutive frames to each other in the latent space. Finally, Li et al. [167] proposed to add a step of pre-training to their spatio-temporal architecture. This pre-training aims at predicting the Scene Graph from the next frame given past frames. During fine-tuning, this temporal context is combined with information retrieved from the current frame to output the current graph. Authors claim a significant improvement over previous work [31] with this method on Action Genome. Unfortunately, the aforementioned approaches are not yet mature enough to be used in real-life applications. For instance, the TRACE architecture needs to relate all frames from a video clip to each other, generating relations between past and future frames to generate predictions. In real-world scenarios, predictions should be made in real-time and the model should not have access to future frames. Finally, these approaches are not tailored for the task of Continuous SGG. They are designed to generate a scene graph for every new frame in the video, without keeping track of past predictions. This is not suited for our approach as we need to keep a memory of past relations to ensure consistency in the representation and allow further reasoning.

Datasets. In a first attempt at leveraging the power of compositional relations for video understanding, Shang et al. [165] proposed VidVRD in 2017. VidVRD is a dataset of 1,000 videos with 35 object classes and 130 predicate classes for a total of 55,631 annotated triplets. This dataset is focusing on temporal actions with relation triplets being annotated between agents (person, animals, etc...) and objects. At the same time, VidVRD contains predicates which are compositions of multiple relations, such as *walk behind* or *walking faster than*. The video clips are short and depict activities being shared by multiple agents and their interactions (such as someone walking their dog on the beach). This dataset is a good starting point for Video SGG but is limited in the number of relations and objects annotated. It is also not representative of real-world scenarios as the actions are limited to a few classes and the dataset is not large

enough to cover all possible relations between objects, especially in the domestic context. In 2019, Shang et al. proposed a new dataset, VidOR, which encompasses more realistic video clips for the task of Video SGG. VidOR is a large-scale dataset of 10,000 videos with 80 object classes and 50 predicate classes. 42 of the 50 predicates are actions-related and the other 8 are spatial relations. Like VidVRD, the dataset is mostly centered on representing actions between agents and objects. The dataset is more diverse than VidVRD and contains more complex actions such as *shake hand with* or *playing (an instrument)*. However, the dataset is still limited because it does not cover relations between objects in cluttered scenes. The ActionGenome dataset [85] is another dataset annotated with scene graphs in videos. It represents a set of 10,000 short video clips of an average 30s with HOIs annotations. Even though this dataset is used to benchmark the most part of recent approaches in Video SGG, it is in fact a video HOIs (HOI) dataset as it does not represent object-object relations. To our knowledge, there is no large-scale Video SGG dataset that covers all relations in the frames, not only relations involving human subjects in a comprehensive way. This is a major drawback for our approach as we need background annotations between all entities to accurately model the context of the scene. Video SGG datasets focus on modeling dynamic relations between moving objects and agents, forgetting static relations which can still be relevant to describe the *gist of the scene*.

6.2 Informative SGG From Videos

In the previous sections, we showed that Video-SGG methods and datasets are not yet mature enough to be used in real-world Human-AI Collaboration. Building on our real-time SGG approach introduced in Chapter 5, we propose to extend this architecture to videos. However, using our Modified PE-NET model for inference in real-world images requires ensuring to filter out False Positive detections. We cover this paradigm in Section 6.2.1. To capture the history of relations from consecutive frames, we add a new module for Multi-Object Tracking (MOT) to the SGG backbone to track similar visual entities in the video flow, see Section 6.2.2.

6.2.1 SGG Backbone

To generate informative graphs in every frame in real-time we used the Modified PE-NET (M-PE-NET) model previously introduced which is optimized for real-time edge computing (see Chapter 5). Inference in real-world images requires setting two confidence thresholds to filter out (1) low-confidence objects detected and (2) low-confidence relations detected. For the object detection, we set a threshold $\alpha = 0.194$ by taking into account the average best performance in the F1-score of all classes, see Figure 6.1. In this image, we can see that the average the F1-score of all classes is maximized at a confidence threshold of 0.194 (strong blue line). For object detection, the F1-score is computed as the harmonic combination of precision and recall,

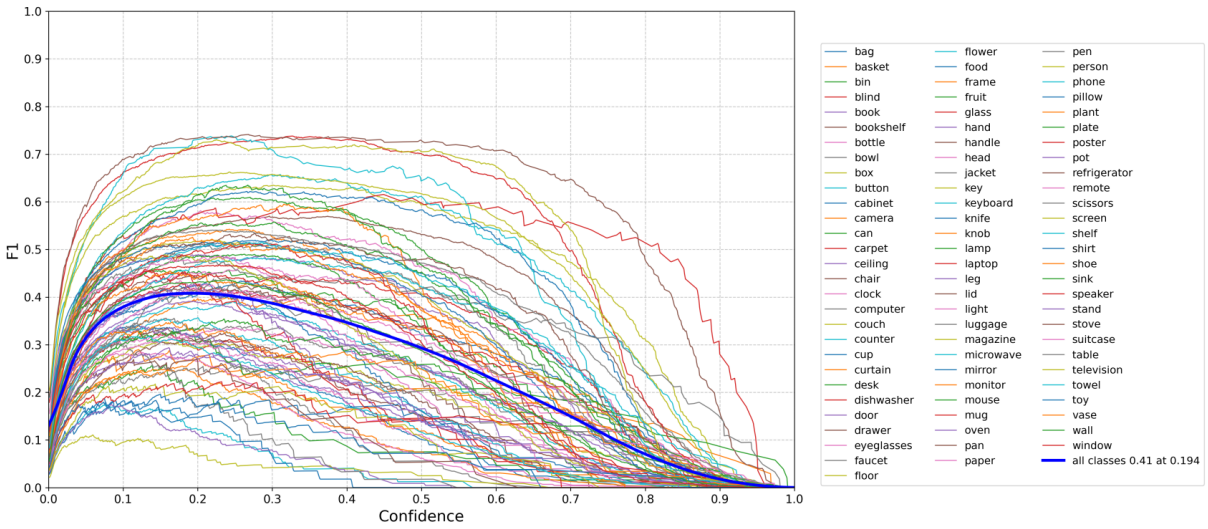


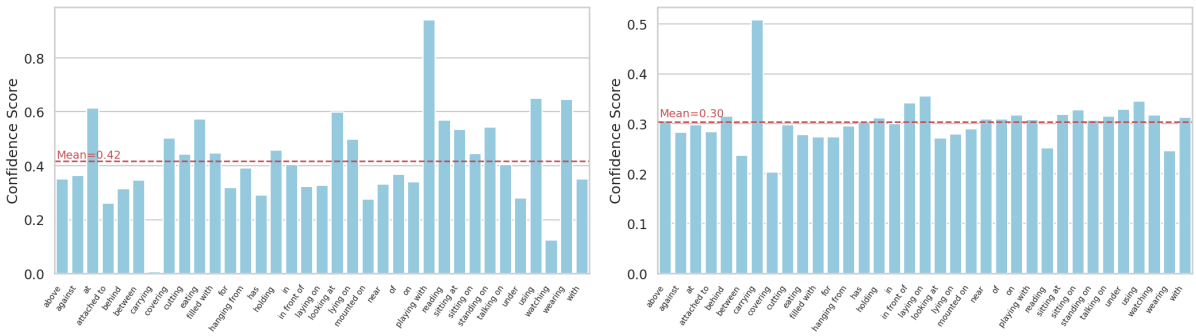
Figure 6.1: F1-score of the YoloV8-m model for all classes in the IndoorVG dataset for different confidence thresholds. The average F1 is maximized at a confidence score of 0.194.

as follows:

$$F1 = 2 \times \frac{precision \times recall}{precision + recall} \tag{6.1}$$

This metric is different from the F1@K score for relations. The best performance is for the class *refrigerator* with an F1-score of 0.74 at 0.28 confidence threshold. The worst performance is for the class *key* with a maximum F1-score of 0.11 at 0.06 confidence threshold. The performance is particularly low for this class because both *keyboard key* and *door key* are annotated with the *key* label, which confuses the model.

For relation prediction, we post-process predictions using the Informative Inference algorithm introduced in Section 4.5 to ensure that only informative relations are kept. This algorithm re-ranks relations based on a combination of their informativeness score and confidence score. As we cannot compute the F1-score for relation prediction, we choose instead to use the Recall@K metric to find the best confidence threshold for relations. After the Informative post-process, we use the combined score as the new confidence score for each predicate class and average this score for each true prediction in the top 100 relations (which correspond to Recall@100). In Figure 6.2a, we show the average confidence score per class for the Recall@100 metric. We compared the performance of the Informative Inference algorithm with the baseline algorithm that only uses the confidence score of the model in Figure 6.2b. We can see that the Informative Inference algorithm is able to increase the confidence score for all classes, especially for rare classes. This is particularly important for our approach as we want not only to use confidence to select true predictions but also to use confidence as an initial weight for the relations in the Global Scene Graph. Edge weights in the GSG are used to remove low-confidence relations after



(a) Baseline

(b) Informative Inference

Figure 6.2: Average confidence score per class for relations prediction in the IndoorVG dataset for the M-PENET model (Recall@100).

Settings	R @20/50/100	mR @20/50/100	IR @5/10	F1@K (avg)	Latency (ms)
Baseline	17.81/19.10/19.21	15.66/16.79/16.96	14.98/20.41	17.52	26.57
Informative	12.11/17.84/19.21	8.58/15.48/16.96	16.32/21.32	14.88	30.87

Table 6.1: Performance of the M-PE-NET SGG model on the IndoorVG dataset, with $\alpha = 0.194$.

a certain number of timestamps without update, as explained in Section 6.3.1.

In Table 6.1 we display the final performance of our model with an active selection of informative relations and a confidence threshold for object detection set at $\alpha = 0.194$. We can observe that the performance in Recall@K and meanRecall@K is decreased compared to the baseline without informative selection. However, the InformativeRecall@K has increased, which is the most important metric for our approach. The post-processing by informative selection is also increasing the latency of the model, which is expected as we are adding a new step in the inference process. The latency only increases by an average of 4ms, staying under our threshold of 50ms (see Chapter 5).

6.2.2 Object Tracking

To model compositional relations through time, object and subject nodes need to be persistent. To solve this issue, we proposed to use an Object Tracking approach to track similar visual entities in the video flow. We added a new module for Multi-Object Tracking (MOT) to the two-stage SGG architecture presented in Chapter 5. This module takes as input the bounding boxes and class labels predicted by YOLOV8 (see Figure 5.2 **(d)**) and generates a corresponding identifier (ID) for each of them.

Here, we specifically used the OC-SORT approach [168] for real-time MOT. OC-SORT is a state-of-the-art approach in MOT that is able to track objects in real time with high accuracy.

The approach is based on the popular Simple Online and Realtime Tracking (SORT) algorithm [169]. SORT uses a Kalman filter [170] combined with a Hungarian matching algorithm [171] to estimate the position of objects in the video flow and match new detections with known objects. SORT-based approaches can be inefficient for tracking objects during occlusions as a Kalman filter typically assumes linear trajectory during occlusions. To solve this problem, OC-SORT introduces a virtual trajectory estimation based on pre- and post-observations before and after occlusion to update the Kalman filter parameters. Combining this with traditional Kalman-based SORT approaches makes the overall tracking more robust to occlusions and noisy detections, without significantly impacting running time [168].

To further boost the performance and reduce memory consumption, we used OC-SORT with the class-based association. For multi-class object tracking, objects can be tracked independently for each class. This process lowers the matching cost as the Hungarian algorithm will only be applied to objects of the same class. If no more than one object of each class is present in the image, the tracking will be almost instantaneous because the only operation performed in this case is to save bounding box coordinates for the Kalman filter. We used a custom implementation of the OC-SORT algorithm based on the popular implementation *boxmot* for real-time object tracking [1]. We set the maximum age variable to 30: after this period of time with no update, past detections are removed from the tracking list.

Finally, we feed the OC-SORT tracker with the bounding boxes and class labels predicted by YOLOV8 and the confidence score of the model. The tracker outputs a unique ID for each detected object, which is then associated with the corresponding node of the scene graph and fed to the Global Scene Graph (GSG) representation.

6.3 The Global Scene Graph

We define a Global Scene Graph (GSG) structure that is continuously updated over time and that serves as the internal memory of the agent. This structure takes as input the scene graphs generated by the SGG backbone at every timestamp and maintains a continuous representation of the environment through time. We aggregated relations at every new timestamp, which led to a lack of consistency. To ensure consistency in such representations, traditional approaches use handcrafted ontologies [72], [160], [161]. Defining a comprehensive ontology of relations would be difficult in our case as our SGG model can predict up to $m \times n \times (n-1)$ different $\langle subject, predicate, object \rangle$ triplets, with m being the number of predicate classes and n the number of visual object classes. For the IndoorVG dataset, with $m = 37$ and $n = 84$, there are 257,964 possible relations which lead to a considerable amount of rules to define. Whilst we cannot manually define proper rules for every possible relation, we can define rules per

[1] `https://github.com/mikel-brostrom/boxmot`

relation category. In fact, we introduced in Section 3.2 a taxonomy of relation types, which can be autonomously inferred using a language model (see Section 3.4.3). Given this taxonomy, we split our *Global Scene Graph* into four different layers: **Functional**, **Topological**, **Part-Whole**, and **Attributive**. This multi-layer representation is called a *multiplex network*.

Definition 6.3.1 (Multiplex Network)**.** A *multiplex network* is a graph where nodes are connected by multiple types of edges [172]. In our case, the nodes are the entities detected in the scene and the edge types are the relations categories.

Our GSG is a multiplex network but keeps the same node attributes as the Scene Graph generated from the video, which are the bounding box coordinates, the class label and the corresponding ID. For a set of object classes O, a set of relations R, and a set of layers D we have:

$$\mathcal{G} = \{V, E, D\} \tag{6.2}$$

$$v = \{o, \{x, y, w, h\}\}, \quad o \in O \tag{6.3}$$

$$e = \{u, v, d, r, \tau, \omega\}, \quad \{u, v\} \in V, \quad d \in D, \quad r \in R, \tag{6.4}$$

with $\{x, y, w, h\}$ being the bounding box coordinates of the object o, τ the timestamp of the relation and w the weight of the relation. We detail the computation of ω and its usage in the next section. We display a representation of this multiplex network in Figure 6.3.

6.3.1 Edge Dynamics

To make edges continuous through time, we define edges as matrices of size $n \times m$ where n is the number of timestamps and m is the number of layers ($m = 4$ in our case). Each cell of the matrix represents a *state* of the relation between two nodes at a given timestamp and for a given dimension. To make this process more robust, we need to filter out wrong detections by ensuring time consistency with previous detections. Inspired by the work of Zhuo et al. [173], we deployed a state refinement mechanism for this purpose. State refinement goes as follows: we set a sliding window variable θ ($\theta = 3$) that represents the number of timestamps to consider for the state refinement. For every new relation detected, we compare it to its previous states and wait for future detections to confirm or infirm the relation. An example is depicted in Figure 6.4.

In addition, every new relation added to the GSG is given a weight value ω_r. ω_r is a value of confidence or certainty that the relation exists. When an existing relation is detected again, we update its confidence value as follows:

$$\omega_r = \omega_{(r-1)} + \sigma(\tau_c - \tau_r), \tag{6.5}$$

with $\omega_{(r-1)}$ the previous confidence value of the relation, σ a constant value ($\sigma = 0.5$), τ_c the

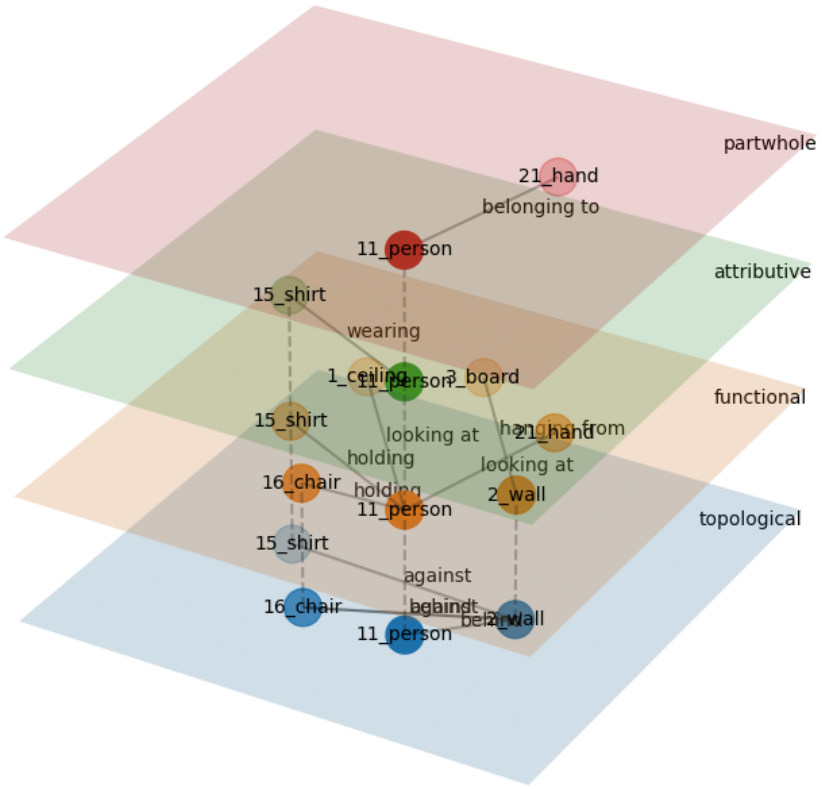


Figure 6.3: Example of a scene graph as a multiplex network with 4 layers, each layer representing a relation category (**Part-Whole**, **Attributive**, **Functional** and **Topological**). Dash lines connect similar nodes from different layers, they are added only for visualization. Plane lines represent relations between nodes.

current timestamp and τ_r the last timestamp of the relation. This ensures that relations detected multiple times in a row will have a strong confidence value. We use the original confidence value given by the SGG backbone as the initial weight of the relation.

6.3.2 Consistency

Inserting new relations in the GSG can lead to inconsistencies in the representation. We used *Constraint Optimization* [152] to refine the edges of the graph according to a set of commonsense rules. To be able to deploy our solution easily in a new environment and with new data, rules need to be set up only at the layer level. As a result, our approach can be easily adapted to new data and new scenarios. We defined a set of rules to ensure the consistency of the representation, as follows:

- Transitivity: an object node of a part-whole relation possesses the same topological rela-

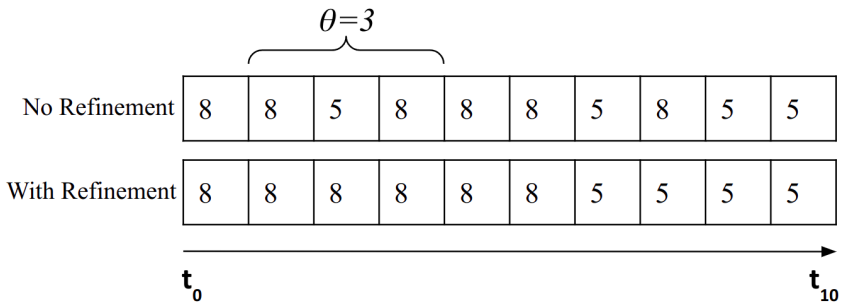


Figure 6.4: Example of state refinement for a relation between two nodes at a given layer. States are represented by the label of the relation, for instance 8 = *above* and 5 = *below*. The sliding window is set to 3 timestamps (i.e. $\omega = 3$).

tions as the subject node, example:
$\langle person, behind, table\rangle + \langle hand, part\ of, person\rangle \rightarrow \langle hand, behind, table\rangle$

- Belonging: A node cannot be the object of more than one part-whole relation. A new part-whole relation detected with a new entity will remove the current one.
- Functions: A functional relation will be removed if the object or subject nodes are not detected anymore at time t_n. This is to ensure that functional relations are only kept if the object or subject nodes are present in the scene.

Our Global Scene Graph representation is generated over time to represent contextual fine-grained information through **Functional**, **Topological**, **Part-whole**, and **Attributive** relations. This design allows the easy refinement of the generated graph according to basic semantic rules. Temporal consistency is also handled thanks to dynamic weight for each relation. In the following section, we detail how we use this new representation for commonsense reasoning in the context of Human Activities Understanding.

6.4 Evaluation: Activity Classification

To evaluate the relevance of our Global Scene Graph representation, we performed experiments with the downstream task of temporal action recognition from videos. Temporal action recognition aims at classifying every frame in a given video clip into a set of known human action classes. Typical approaches in this task use image features to train DNN models in a supervised manner [174]. In contrast to previous work, we use exclusively the generated Global Scene Graph to infer the action being performed in a given video clip, without the addition of any visual features. The hypothesis here is that contextual information from background relations will help to classify the current action, even when no object label is present in the action class (e.g. "Someone is eating something"). We used 200 videos from the Charades dataset [175] that are all

annotated with one or more human actions. We removed actions that are not based on relations with objects (such as <*Someone is smiling*> or <*Someone is standing up from somewhere*>) to keep 131 action classes. For every video, we generated a Global Scene Graph representation by applying our approach. At every new timestamp, we predict the activity being performed as follows:

1. We select all relations directly or indirectly linked to the *person* node, except the *part-whole* relations as they are not contextual (part-whole relations are still used indirectly in the *Semantic Consistency* refinement of other relations)

2. We retrieve the BERT [110] embedding of every $\langle subject, predicate, object \rangle$ triplet

3. All embeddings are averaged and compared to stored embeddings of activities using cosine similarity

4. Action candidates are ranked using the confidence weight ω_r of relations

5. The top-1 activity is selected

This process ensures that relations with higher weights are more important for the final decision. Such relations will often be related to foreground objects or persons that are essential to the understanding of the scene. We evaluate our approach using the mean Average Precision (mAP) metric on all action classes from all video clips. If no ground truth is available in one frame, prediction is not counted as a false positive but discarded instead. For efficiency, we processed the videos at 6fps. We used as a baseline the static scene graph generated by the *SGG backbone* in every frame to compare our approach. Results are shown in Table 6.2. Current state-of-the-art methods in this task report result up to 26.95% mAP on this dataset [176], but with slightly different evaluation protocol (evaluation on evenly-sampled 25 frames from each validation clip).

Even though our approach did not outperform previous work, our results are significant taking into account that we did not specifically train on the Charades dataset. In fact, our approach for this task can be associated with zero-shot settings. We qualitatively experienced better results with actions relying on specific objects that our backbone was able to detect. However, we also experienced good results in high-level actions (such as *Someone is cooking something*) relying only on background contextual relations from our graph (such as $\langle person, in\ front\ of, stove \rangle$).

In the Charades dataset, there is only one person in every video, however, our approach can predict actions related to n different humans at a time, taking into account that each human node possesses at least one relation in the graph, which is not necessarily the case of end-to-end learning approaches for the task [176]. In addition to temporal action recognition, our approach is also able to provide a spatially grounded *rational* for every prediction in the form of

	Continuous SG	**Static SG**
mAP	0.1292	0.02

Table 6.2: Results on Temporal Action Recognition from the Charades dataset [175] using (1) the M-PENET with our continuous implementation (see section 6.3) and (2) the static implementation (i.e. all previous relations are discarded at every new timestamp).

the top-related relations in the graph and associated bounding boxes. This is likely important information for an autonomous agent to understand the context of the scene and to be able to interact with the environment, for instance in a HRC scenario. In the next section, we explore such scenario with new experiments in the context of learning new tasks from observations.

6.5 Experiments: Learning From Observations

In natural interactions between humans and autonomous agents in domestic environments, a common behavior is the observation and reproduction of simple tasks. We can take for instance the task of cleaning a table after dinner. In such a scenario, we assume that the human has a definitive understanding of the task and can perform it without any additional information. In contrast, the observer (in our case the AI agent) has no priors of the task or the environment. This is a common scenario in HRC where the robot needs to learn a new task by observing a human expert. In such settings, the common approach is to learn low-level trajectories and transfer them to the autonomous agent [177]. However, this approach is not robust to (1) changes in the environment, (2) changes in the scheduling of the task, and (3) changes in the capabilities of the autonomous agent. To solve these problems, it could be convenient to learn a *symbolic representation* of the task, with correct scheduling, and apply basic low-level behaviors of the robot to reproduce the task.

Such behavior is related to the automated generation of robotic planning domains from demonstrations [178]. In a recent work, Andrea Maria Zanchettin [154] presents a new approach to demonstration learning which aims at learning the symbolic representation of human demonstrations. In their approach, authors proposed to use a graph representation of the scene which is then translated to a PDDL Planning Domain through various axioms. This approach is promising as it enables learning a symbolic representation of the action from a single demonstration, whereas usually other approaches often need more [179]. Inspired by the work of Andrea Maria Zanchettin, we propose to use our Global Scene Graph representation to learn a planning domain, which could be further used by the agent to reproduce the observed actions. We use the Planning Domain Definition Language (PDDL) [40] to model this planning domain.

6.5.1 PDDL Implementation

PDDL is a formal language for defining planning domains [40]. A PDDL domain is defined by a set of types, predicates, actions, and constants (optional). We define two types for our approach: *agent* and *object*. The nodes in the GSG with the label *person* will be defined as *agent* and all other nodes as *object*. Predicates are defined as relations between two entities in our graph, e.g. $behind(object, object)$. Actions are an important component of a planning domain because they define rules of changes in the environment. Actions in PDDL are specified by:

- Parameters: the entities involved in the action
- Preconditions: the conditions that must be true for the action to be executed
- Effects: the changes in the environment after the action is executed

To identify preconditions and effects, we define *transitional states* as actions of an *agent* which modify the state of at least one *object*. In our case, we define a transitional state as a relation in our graph that modifies one or more relations of another node at the next timestamp. For instance, the relation $\langle person, holding, bottle \rangle$ at time t_n can modify the relation $\langle bottle, on, table \rangle$ at time t_{n-1} to $\langle bottle, on, shelf \rangle$ at time t_{n+1}. We can then identify the preconditions and effects of this action by comparing the Global Scene Graph at two different timestamps. Instead of only looking at the previous and next timestamps, we set a sliding window of 10 frames to look back and forward for relation changes. This approach is necessary because actions can span long durations and may be interrupted by other actions. Setting this threshold will also cope with missing detections in the scene graph.

Some actions can directly modify the states of objects (such as *holding*, *drinking*) while some other will have more indirect effects (such as *sitting on* or *lying on*). For simplicity here, we will take the example of the *holding* action in the **Functional** layer to demonstrate our proposal. In the same way, for simplification, we will only consider the **Topological** layer of the Global Scene Graph to define the preconditions and effects of the action. In Algorithm 4, we describe the algorithm to find preconditions and effects for a given transition in the GSG. This algorithm first looks for preconditions for a given transition and then looks for effects. The algorithm is applied for every transition of interest in the GSG.

We propose to visualize one example of a PDDL action identified using the proposed algorithm. In a short video clip, one person (i.e. an *agent*) is moving a glass from a table to a shelf. Figure 6.6a shows the Global Scene Graph (left) before the action takes place. We see the relation $\langle glass, on, table \rangle$ identified. In the next frame, the person is holding the glass and moving it to the shelf, see Figure 6.6b. We can see the relation $\langle person, holding, glass \rangle$ in the GSG and the absence of $\langle glass, on, table \rangle$. Finally, the glass is placed on the shelf, see Figure 6.6c. The relation $\langle glass, on, shelf \rangle$ is identified in the GSG and $\langle person, holding, glass \rangle$ disappears. The

Algorithm 4 Finding preconditions and effects for a given transition in the GSG.

Inputs:
  Target relation rel
  Functional states $\mathcal{F}_{[0 \to n]}$
  Topological states $\mathcal{T}_{[0 \to n]}$
Outputs:
  Preconditions $\mathcal{P}$
  Effects $\mathcal{Q}$
temp_states $\leftarrow \emptyset$
$\mathcal{P} \leftarrow \emptyset$
$\mathcal{Q} \leftarrow \emptyset$
sliding_window = 10
for each index i in the range of len($\mathcal{F}$) **do**
  for each state in $\mathcal{F}[i]$ **do** ▷ Here we look for preconditions
    if state $\in$ rel **then**
      temp_states[state] = i
      obj = state[1] ▷ Find object of the transition
      $\mathcal{P}$[state] $\leftarrow$ empty list
      for each index j from $\max(0, i - \text{sliding\_window})$ to $i - 1$ **do**
        for each topo_state in $\mathcal{T}[j]$ **do**
          if topo_state[0] == obj **then** ▷ Find object in the topological layer
            $\mathcal{P}$[state].*insert*(topo_state)
          end if
        end for
      end for
    end if
  end for
  for each (k, v) in temp_states.items() **do** ▷ Here we look for effects
    if $v < i$ **then** ▷ Verify that the transition is over
      if k not in $\mathcal{P}$ **then** ▷ No preconditions so no effects
        continue
      end if
      $\mathcal{Q}$[state] $\leftarrow$ empty list
      for each index j from $i + 1$ to $\min(i + \text{sliding\_window}, \text{len}(\mathcal{T}))$ **do**
        for each topo_state in $\mathcal{T}[j]$ **do**
          if topo_state[0] == obj and topo_state $\notin \mathcal{P}[k]$ **then**
            $\mathcal{Q}$[state].*insert*(topo_state)
          end if
        end for
      end for
    end if
  end for
end for

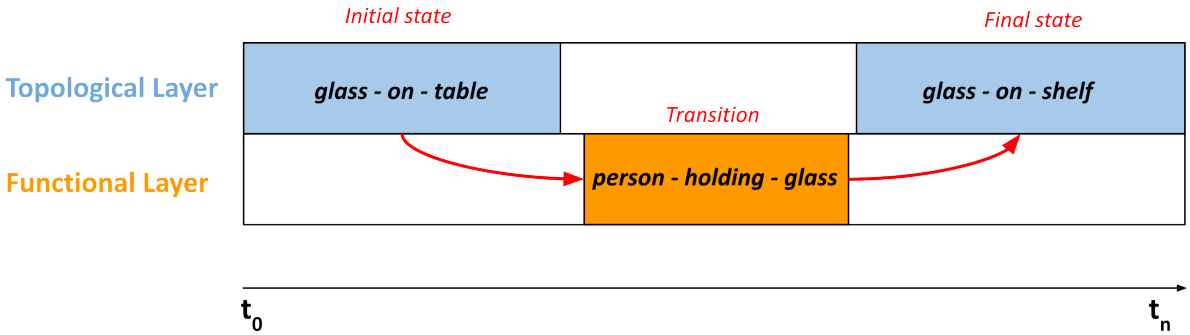


Figure 6.5: Example of a transition identified using the interactions of the *Topological* and *Functional* layers of the Global Scene Graph.

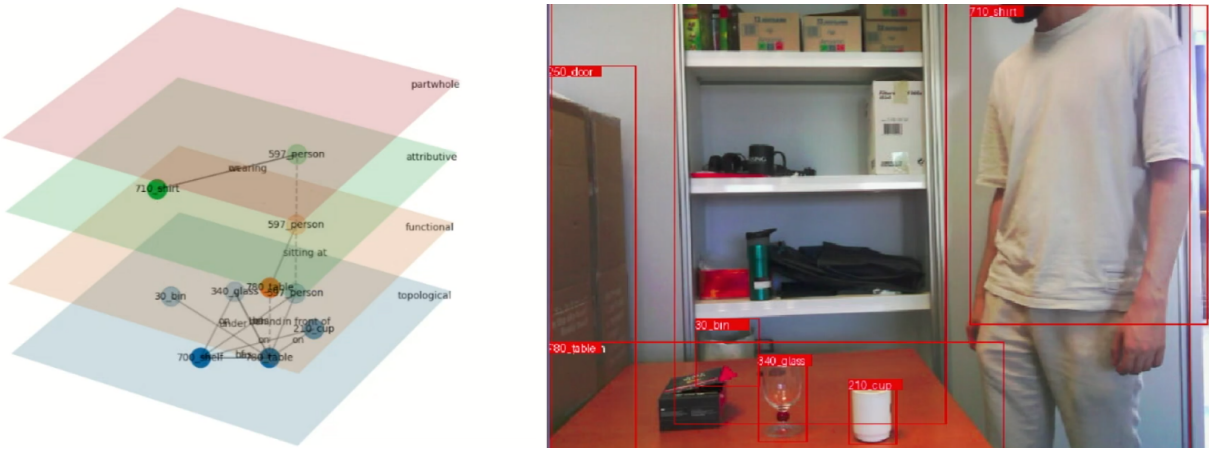
goal of our algorithm is to identify the *initial state* of the world, the *transition*, and the *final state* to generate a PDDL action. Initial states and the transition will serve as preconditions and the final states as effects, see Figure 6.5 for a schematic visualization. In Listing 6.1, we display the PDDL action generated by our algorithm in this example. We can see that the transition from the table to the shelf is correctly identified. Parameters are comprised of all entities involved in the action. The name of the action is generic, in this case, it is extracted from the effect of the action. The process of extracting PDDL actions from our Global Scene Graph can be used in real-world applications, such as during learning from observations.

Listing 6.1: PDDL "move on shelf" action.

```
(:action on shelf
  :parameters (
    ?person - agent
    ?glass, ?shelf, ?table - object
  )
  :precondition (and
    (on ?glass ?table)
    (holding ?person ?glass)
  )
  :effect (
    (on ?glass ?shelf)
  )
)
```

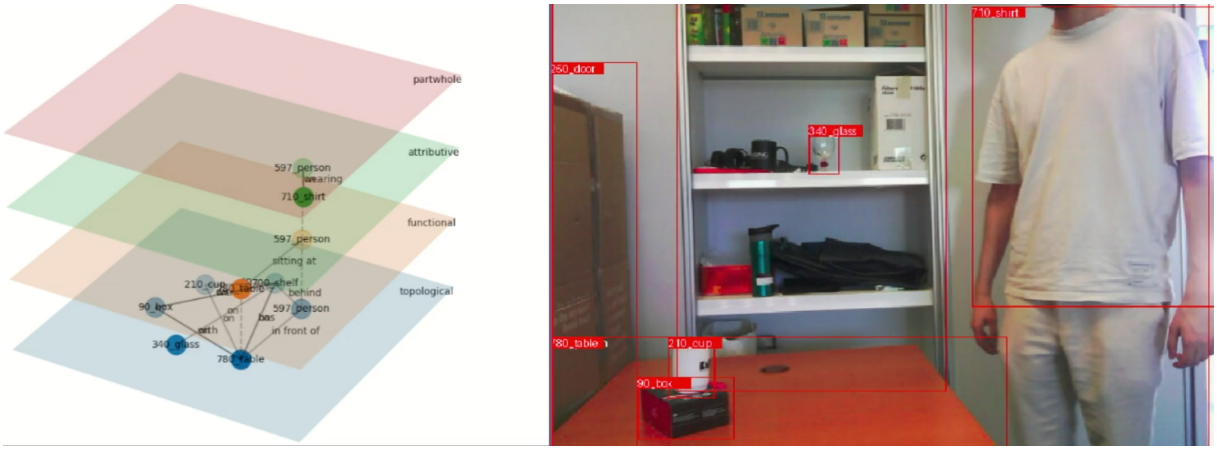
6.5.2 Implementation: ROS2 Integration

In order to test our approach in a real-world scenario, we integrated our Global Scene Graph as well as our Planning Domain Generation algorithm in a Robot Operating System 2 (ROS2)



(a) Initial State, $t = 0$

(b) Transition, $t = 5$



(c) Final State, $t = 10$

Figure 6.6: Global Scene Graph generated at each time step for the action "putting glass on shelf", for clarity representations in between key timestamps are not displayed.

[180] environment. The ROS2 framework is used to gather images from a robot or laptop camera and to control the behavior of the different modules. As per the ROS2 architecture, we defined four different nodes:

- **SGG Node**: This node is responsible for generating a Scene Graph from the RGB images. We used the M-PENET model coupled to the OC-Sort tracking algorithm for this task. In addition, this node performs Informative Selection to keep only relevant relations in the graph.

- **GSG Node**: This node is responsible for generating the Global Scene Graph by aggregating relations gathered by the SGG node. This node also performs the state refinement and adds edge dynamics to the graph. This node also output a dynamic visualization of the GSG for debugging purposes.

- **Reasoning Node**: This node is responsible for generating the Planning Domain from the Global Scene Graph. It queries the current states of nodes from the GSG node and uses the algorithm described in Algorithm 4 to identify preconditions and effects of transitions in the graph. This node can also perform Activity Recognition, as described in Section 6.4.

- **Manager Node**: This node is responsible for managing the communication between the different nodes. It can launch or stop other nodes by using service or action calls.

This architecture is summarized in Figure 6.7. The SGG node listen to the `/camera/image_raw` topic and outputs the Scene Graph on the `/sgg` topic. The GSG node listens to the `/sgg` topic and outputs the Global Scene Graph on the `/gsg` topic. The Reasoning node listens to the `/gsg` topic. It can be triggered by a service call to generate the Planning Domain. The Manager node listens to the `/gsg` topic and can trigger the Reasoning node by a service call. The Manager node can also trigger the Activity Recognition by a service call. Each PDDL actions generated is stored in an actions bank and can be access by the Manager node. This architecture can be easily extended with the addition of a Planning Node that would use the generated Planning Domain (actions bank) and the GSG to plan a sequence of actions to perform a given task (see *Planning Algorithm* in Figure 6.7). This architecture has been implemented on a Pepper robotic platform [2], with the scene graph generation node being deported on an external laptop for computational reasons. The robot is equipped with two RGB cameras, we used the front camera for getting RGB images at 30FPS. The overall latency of the system is around 5FPS (SGG + GSG + Reasoning) which is sufficient for real-time applications. Next, we evaluated the relevance of our approach on the DAHLIA dataset [181], a real-world dataset of daily life activities.

[2] `https://corporate-internal-prod.aldebaran.com/en/pepper`

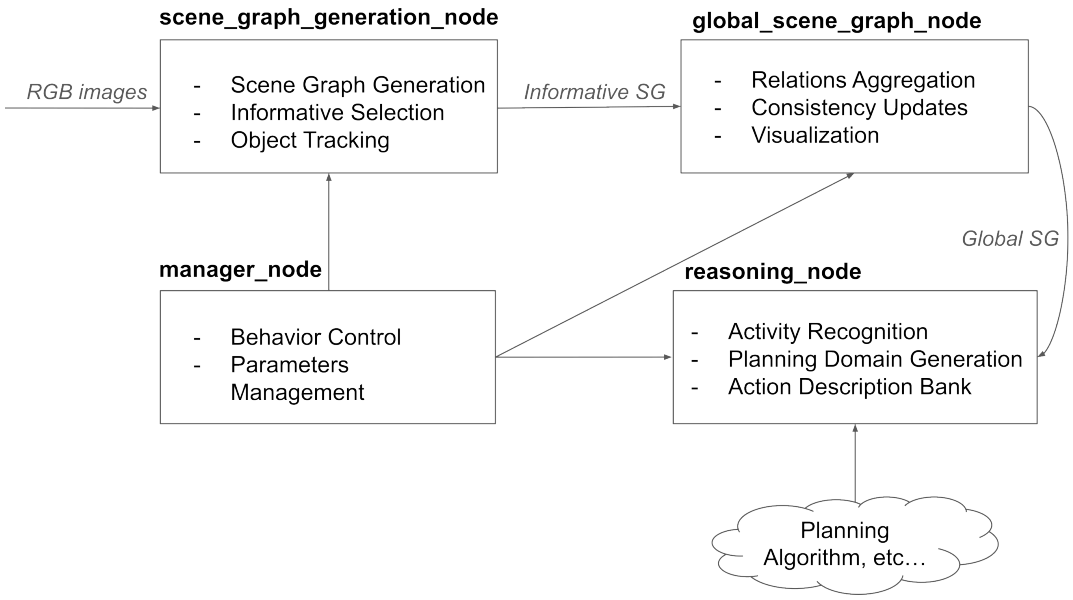


Figure 6.7: ROS2 Integration of the Global Scene Graph and Planning Domain Generation algorithm.

6.5.3 Evaluation: The DAHLIA Dataset

The DAily Home LIfe Activity (DAHLIA) dataset is a dataset of long-term human activities performed in home environments. The dataset is composed of 44 videos of 44 different subjects performing daily life activities such as cooking, cleaning, or working. The dataset contains 7 different annotated activities performed in a kitchen environment. In addition to RGB videos, the dataset contains depth maps and skeleton data collected by a Kinect sensor. Three different viewpoints are available for each video and each video lasts an average of 39 minutes. The dataset is annotated with the following activities: *Cooking*, *Laying Table*, *Having Lunch*, *Clearing Table*, *Washing Dishes*, *Working* and *House working*.

At the difference of other ADL datasets [175], DAHLIA introduces very long videos of daily life activities, which is more representative of real-world scenarios. Benchmarking our approach on the DAHLIA dataset is particularly valuable because our Global Scene Graph representation is designed for continuous updates over long period of time and can be used to learn multiple tasks with the same representation. Another advantage of DAHLIA is the high quality of the videos (lightning, resolution) and the diversity of objects and their interactions in the scene. We display an example of two activities from the DAHLIA dataset in Figure 6.8. The first activity (Figure 6.8a) is *Working* and the second activity (Figure 6.8b) is *Cooking*.

We used the DAHLIA dataset to evaluate our approach for automated planning. We used the same approach as previously described to generate a Planning Domain from the Global Scene Graph. First, we evaluate the relevance of our approach by measuring the number of correct ac-

(a) (b)

Figure 6.8: Example of two activities of the DAHLIA dataset.

Video	True Positive	False Positive
1	21	14
2	19	28
3	59	44
4	19	18
5	43	51
Recall	0.51	

Table 6.3: Performance of the Automated Planning approach on the DAHLIA dataset.

tions identified and translated into PDDL. We ran our approach on 5 different videos randomly sampled in the DAHLIA dataset. The length of the videos ranges from 32 to 51 minutes. Because the DAHLIA dataset does not contain any ground truth for the scene graphs or actions, we manually evaluated the relevance of the generated PDDL actions. For each PDDL action generated, we watched the corresponding video clip and evaluated if the action was correctly identified and translated into PDDL. We then computed the Recall of the approach as the number of correct actions identified divided by the total number of actions in the video. We display the results in Table 6.3. We can observe that our approach can identify correctly a significant number of actions in the scene, with less than 50% of false positives. This is encouraging as it shows that we can learn, end-to-end, a significant amount of actions performed in day-to-day activities from a symbolic representation. We observed that a lot of actions generated contained the same set of objects and relations. We hypothesize that the SGG backbone could be in fault here. Likewise, we can pinpoint two issues: (1) the default of the object detector to detect certain classes and (2) the lack of diversity in the predicate classes of the IndoorVG dataset. We believe that with a bigger and more diverse base dataset, our approach could be able to identify more actions in the scene. To test this hypothesis, the DAHLIA dataset could be directly annotated with object coordinates and relations, allowing to train our SGG model in better conditions.

We also display the results of the same approach without the Informative Selection process

Video	True Positive	False Positive
1	3	0
2	1	0
3	3	1
4	1	2
5	1	1
Recall	0.69	

Table 6.4: Performance of the Automated Planning approach on the DAHLIA dataset, SGG without Informative Selection.

in Table 6.4. We can observe that the number of actions identified is significantly lower than with the Informative Selection process. Even if the Recall is higher, with this few number of actions identified, measuring the recall is not relevant. This shows the importance of selecting only informative relations in the Global Scene Graph to generate actions in PDDL.

We tried to understand why there is such a significant gap between the number of actions identified with and without the Informative Selection process. For generating graphs without the Informative Selection process, we used the output of the SGG backbone with a confidence threshold of 0.3 for relations (see Figure 6.2a). By comparing the number of relations predicted in each setting, we were able to pinpoint the issue: the number of relations predicted by the SGG backbone is significantly higher when using the Informative Selection process, especially for action-related predicates. We display a comparison of the number of relations predicted by predicate class in Figure 6.9. For clarity, we display numbers in a log scale. We observed a higher count of relations predicted with our Informative Selection, especially for fine-grained predicates such as *sitting on*, *cutting*, *reading*, or *looking at*. We also observed a higher count of spatial predicates such as *in front of*, *behind*, or *under*. This shows that the Informative Selection process is very important to generate relations that could be used to generate a Planning Domain from the Global Scene Graph. Of course, it is very likely that more False Positive relations are predicted with the Informative Selection process, but this seems to be still beneficial for the performance of our PDDL parser.

6.5.4 Discussion

Our approach to automated planning discovery can be extended in many ways. For instance, we can learn hierarchical relations between object classes by aggregating actions generated by our PDDL algorithm, without any priors on the object classes. To follow up our previous example, we can learn a new class of entities called *movable* by aggregating all actions that involve moving an object from one place to another. The action of "moving" is determined by a change of relation in the **Topological** layer of our Global Scene Graph, as we have seen with the example of the

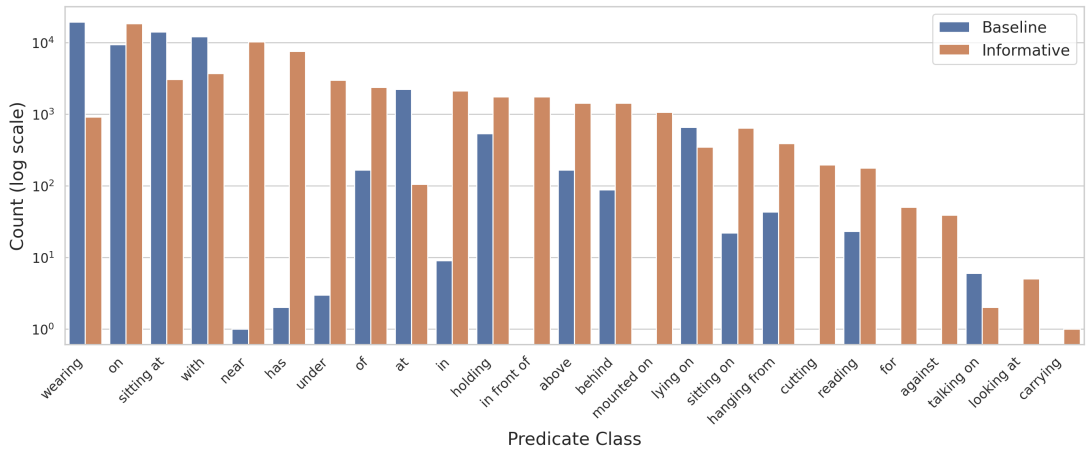


Figure 6.9: Number of relations predicted by predicate class per video in the DAHLIA dataset. Average over 5 videos.

Movable Objects	**Static Objects**
bottle, door, bag, basket, bowl, cup, knife, glass, plate	cabinet, microwave, counter, sink, faucet, floor, wall, table, shelf, stove

Table 6.5: Classes of entities learned from the DAHLIA dataset.

glass being moved on the shelf (see Figure 6.5). By looking at all objects involved in this type of action during the course of the observations, we can find patterns and define a new class of entities. On the other hand, objects involved in the preconditions but never in the effects of an action can be defined as *static* entities. By mining these two types of objects in the actions generated from the DAHLIA dataset with the "holding" action, we obtained a realistic list, see Table 6.5. This example demonstrates the perspectives of our approach for related tasks such as learning new concepts from continuous symbolic representations.

However, limitations are still important. As we have seen with the DAHLIA dataset, a significant amount of PDDL actions generated are wrong, due to wrong detections of the SGG backbone. For instance, the relation $\langle person, holding, microwave \rangle$ was almost systematically detected every time the person in the video was walking past the microwave, even when there was no contact between the person and the microwave. This shows the limit of SGG models in the real world and the need for more robust approaches. A second limitation that we noticed is the performance of the Multi-Object Tracking algorithm [168]. This type of tracker can struggle to keep track of objects which move a lot in very long time frames. We can observe that this is especially the case in the DAHLIA dataset where the camera is fixed, and the person is moving a lot in the scene, turning his back to the camera or moving out of the field of view. This is a common issue in real-world scenarios and a lot of research is still needed to improve the performance of these trackers [182]. One opportunity for our Global Scene Graph representation

is to augment the tracker by providing the history of relations for lost tracklets. For instance, if the person is turning his back to the camera and the corresponding tracklet is lost, it would be convenient to compare the relation of a new person's tracklet and the former one to find similarities. As an example, the **Part-Whole** layer could be used here to look for relations between the person and its parts (such as the hands) to find similarities, and then re-associate the tracklet. In this case, we assume that the tracklets for the parts have not been lost.

6.6 Concluding Remarks

In this chapter, we introduced a new approach for Continuous SGG in robotics. This approach models a new representation of the environment called the Global Scene Graph, which represents the evolution of compositional relations through time. This representation is powered by a SGG backbone and a Multi-Object Tracking module. In contrast to standard representations of the sort, we proposed to divide our representation into four different layers, each representing a category of relations: **Topological**, **Functional**, **Part-Whole**, and **Attributive**. This process can alleviate the lack of consistency in the representation with the use of dedicated commonsense rules. We presented four different axioms which are used to maintain the consistency of the representation via Constraint Optimization. Our Global Scene Graph is an end-to-end representation that can be generated directly from videos, furthermore, it is generic and can be used with any scene graph dataset or any SGG model. We successfully implemented our approach on a Pepper robot in a ROS2 environment.

We illustrated the opportunities of our approach in two downstream tasks: Temporal Action Recognition and Automated Planning. We showed that our approach is able to predict actions from videos without the need for visual features and that it can be used to generate Planning Domains in PDDL. We evaluated our approach on the Charades dataset for Temporal Action Recognition and on the DAHLIA dataset for Automated Planning. For the latter, we proposed a new approach to automatically generate PDDL actions from the continuous representation. This approach has been manually evaluated on the DAHLIA dataset as no ground truth was available. Results are promising with an overall Recall of 51% for the Automated Planning task.

Both approaches presented in this chapter open new perspectives for the usage of SGG in Human-AI collaboration scenarios. First, these types of representations are easy to visualize and understand by humans, which is important for trust in robotic agents [183]. Second, our approach can be used in real-time and on low-resource hardware thanks to an optimized SGG backbone (see Chapter 5), which can also be useful for real-world implementation on robotic hardware. However, challenges remain for robustness such as the limitations of the SGG backbone or the performance of the Multi-Object Tracking algorithm.

Chapter 7

Conclusion

What are the main causes of the limitations of current SGG in real-world applications? What are the opportunities for SGG in HRC? These two questions have been the main focus of this work: the former mainly emerged as a result of addressing the latter. At first, generating end-to-end comprehensive scene graphs from visual scenes seemed to be a promising approach for HRC scenarios. SGG methods can alleviate the need for hand-crafted rules and provide a more flexible and interpretable representation of the environment. However, the limitations of current SGG methods have become evident in the context of real-world applications. As an attempt to address these limitations, we have proposed a set of contributions that aim to bridge the gap between theoretical approaches to SGG and real-world implementations. The problems we addressed in this work are not only related to the usage of SGG in robotics but can also be seen as a general limitation of SGG methods. In this chapter, we summarize the main contributions of this work, discuss the limitations of our approach, and outline the opportunities for future research in this field.

7.1 Summary of Contributions

This work has resulted in four main contributions, spanning the field of Visual Understanding, Knowledge Representation, and Robotics Planning. This work covers all the main stages of the data annotation, model training, model evaluation, and deployment in real-world settings of a SGG solution. The main contributions of this work are summarized in the following.

In Chapter 3 we have addressed the problem of data annotation for Scene Graph datasets. Annotating compositional relations in images is a challenging task due to the polysemy of natural language and human biases in selecting relations of interest. As a result, common SGG datasets that have been proposed so far are limited in terms of the diversity and quality of relations annotated. We have quantified this issue by categorizing compositional relations based on their semantic meaning. We found that, for most SGG models, the over-representation of the **Part-Whole** category can lead to biased learning and evaluation. To address this issue, we have proposed a method for refining data annotations by removing irrelevant **Part-Whole** relations. In addition, we proposed a new method for selecting data classes based on their inter-dependence rather than overall frequency, which results in higher-quality annotations. We have evaluated this

method on the popular Visual Genome dataset, however, it can be applied to any other Scene Graph dataset. In particular, we can implement our method for data selection and refinement of in-domain data splits, which can be used in real-world applications. As a study case, we proposed a new data split, IndoorVG, which is a subset of the Visual Genome dataset that contains only indoor scenes. We have shown that the IndoorVG dataset has a higher quality of annotations compared to the original Visual Genome dataset. We have also shown that the IndoorVG dataset can be used to train a SGG model that generalizes better to indoor scenes throughout the following chapters.

If SGG datasets are limited in terms of quality, the actual evaluation process and inference method of SGG models are also limited for real-world applications. Current metrics in SGG are difficult to relate to actual needs in real-world applications. In Chapter 4 we have addressed the problem of informativeness of relations in Scene Graphs. Informativeness is a concept that (as we have defined it) has not been addressed in the literature on SGG. Inspired by the prevalence of informative relations in human descriptions of visual scenes, we have demonstrated that the informativeness of relations can highly impact the performance of SGG models in downstream tasks. We have proposed a new metric for evaluating the informativeness of relations in Scene Graphs, the InformativeRecall@K, which can be used to benchmark approaches for real-world usage. We have also proposed a new inference method, the Informative Selection, which can be used to re-score predictions of SGG models to better reflect the importance of each relation in the scene. This approach has been successfully evaluated in four downstream tasks, including VQA, Image Captioning, Image Generation, and later on in SGG scenarios (see Chapter 6). Our method can improve performance in those tasks, by a slow margin for certain (e.g. Image Captioning) and by a large margin for others (e.g. Image Generation).

In Chapter 5 we have addressed the problem of real-time SGG. Real-time SGG is a requirement for real-world applications, such as SGG, where the robot needs to understand the environment in real time. We have proposed a new method for real-time SGG, which is based on the popular two-stage approach. We have replaced the first stage of the SGG pipeline with the real-time object detector YOLOV8 and modified the second stage to be more efficient. In particular, we have removed computations and requirements that do not significantly contribute to the performance of models to attain real-time inference. We have evaluated our method with different baseline models on the IndoorVG dataset and shown that it can achieve real-time performance on a single GPU, with up to 10x gain in latency and no loss of accuracy.

Finally, in Chapter 6 we have successfully validated our previous contributions in downstream tasks related to robotics applications. To demonstrate the opportunities of SGG in new domains, we proposed a new architecture for Continuous SGG in SGG. In such scenarios, our SGG model serves as a backbone to build a continuously updated internal representation of the environment, a *World Model*. We called this representation the Global Scene Graph (GSG). A Global Scene

Graph can aggregate important relations through time and can be used to plan robotic actions. Here, we proposed a new method for automated planning domain generation from Scene Graphs, based on human observations. This method can be used to collect possible actions and their effects on the environment from human demonstrations, which can be further used by the agent to plan assistive actions for instance. We quantitatively and qualitatively benchmarked our approach on the real-world DAHLIA dataset.

The investigations conducted in this work have permitted to identify a few shortcomings of current SGG methods, which are limiting their applicability in real-world scenarios. In the next sections, we discuss the limitations of our approach and the opportunities for future research in this field.

7.2 Limitations

The limitations of our approach can be summarized in two main points: (1) the domain gap between annotated data and the needs of real-world applications and (2) the over-complexity of SGG models. Regarding the annotations of Scene Graph datasets, we have shown that the quality of annotations can highly impact the performance of SGG models in real-world applications. However, our proposed method is limited to the refinement of existing datasets [8]. A better approach would be to generate new data annotations from scratch, making sure that annotators are taking into account the relevance and informativeness of relations. Nevertheless, such an approach is complicated because SGG models need to be trained on a fixed set of m object classes and n predicate classes, which will always result in images where some relations are missing. To create a "perfect" dataset for SGG, one should carefully (1) determine a correct set of object and predicate classes that is semantically coherent (for instance avoiding polysemy as much as possible) and (2) select training images where all informative relations (i.e. related to the image gist) can be annotated with the provided set of classes. This process is hard to automate because there is no way of knowing which set of classes can cover the widest range of images in the best way possible. Recently, the PSG dataset [20] was released as an effort toward solving this challenge. However, the PSG dataset still lacks from a clear taxonomy of predicate classes, with some debatable choices such as the class "over" which can almost always be replaced by "on" or the class "biting" which can be replaced by "eating" etc. The problem of granularity in object classes is also important to be mentioned. In the PSG dataset, some classes are for instance "wall-stone" or "wall-wood" which are *attributes* of the object "wall" and not different classes. This example shows the need to also take into account the performance of the backbone object detector into account when designing a Scene Graph dataset. A similar problem can be spotted on the GQA dataset [15], where predicate classes such as "to the left" and "to the right" are annotated. However, in a natural scene, every object can be to the left or

to the right of another object, which makes the annotation of such relations irrelevant. If we take a step back, we can see that this problem comes from the initial definition of the task [7]. This definition is deliberately vague in order to be able to annotate as many relations as possible, and easily generate the large-scale amount of data required to train modern deep learning models. In this process, coherence and relevance of relation types has been overlooked. For instance, using ambiguous classes is a failure to consider the actual paradigm of supervised learning, and, more importantly, is a failure to consider the end applications of the task. One last consideration for building SGG datasets is the target domain. As we have observed with the IndoorVG dataset, it is way easier to construct a coherent set of object and predicate classes in a small domain (such as indoor scenes) than in an all-purpose dataset such as Visual Genome which comprises a wide range of scenes in different contexts (such as beaches, cities or nature). Solutions to this problem can be to annotate a very large number of classes, but this requires a lot of human effort and is not designed for models such as Object Detectors. Annotating too many classes will also augment the long tail problem of SGG datasets [78].

Regarding the complexity of SGG models, we have witnessed over the past few years the trend of designing new approaches which are adding new components to existing approaches. The best example of this is the VCTree model [37] with the two-stage Faster-RCNN-based architecture that we have reviewed in Chapter 5. This architecture has been re-used by almost every other approach since its first inception (to cite a few: [49], [92], [101], [142], [184], [185], [186], [187], [188], [189]). The reason for this choice is simple: the codebase provided by the authors of VCTree was the first documented codebase that supported recent versions of PyTorch. Re-using it for new approaches was then simply a matter of convenience and not efficiency or performance. We have demonstrated in our experiments with the PE-NET model that some components of the model can be removed without any loss of performance. These components have been kept in the model because they were part of the original VCTree codebase, and not because they were necessary for the specific presented approach. Or at least, if that was the case, there is no trace of it in the original paper [49]. By adding more and more components to existing approaches, it becomes tedious to disambiguate the actual contributions of each element to the final performance. This can be misleading to fairly evaluate approaches and compare them to each other. This issue is only clear when trying to reimplement a new approach given the provided codebase, as we have done with the PE-NET model. It may not be clear to the reader of the corresponding paper, who is not aware of the actual implementation details of the model. Some approaches do not explicitly state that most of their codebase is similar to the VCTree model, which is misleading for the reader. Last but not least, we believe that the over-complexity of SGG models is strongly misaligned with potential applications of the task. In fact, if SGG approaches are overly complicated and rely on useless components, their actual usage in the real world will be limited.

7.3 Perspectives

In the following, we detail opportunities for future research in the field of SGG, which can be used to address the limitations of current methods and to foster the development of new applications.

Open-Vocabulary SGG. To address the limitations of the annotation process in SGG, it would be reasonable to consider the task in open-vocabulary settings. Because it is very difficult to produce a non-biased dataset, it would be convenient to consider the task as a generative modeling task instead of a classification task. To do so, we can consider the task as a sequence-to-sequence problem, where the input is an image and the output is a sequence of objects and relations. This approach has been successfully applied to Image Captioning [60] and can be adapted to SGG. In this setting, we evaluate the model performance based on metrics such as BLEU [190], METEOR [191] or CIDEr [192] which are metrics that consider the *semantic similarity* with the ground truth. This approach could be used to match generated graphs with detailed descriptions of scenes for instance. Our work with the InformativeRecall metric is a first step in this direction, but it is still limited to the evaluation of the performance of SGG models and not to the training process itself. Training SGG models for Open-Vocabulary settings will be challenging because of the large number of possible relations in each scene. A few works have been proposed to address this issue, using large Vision-to-Language (VLM) models such as CLIP [193] or BLIP [194]. Although using VLMs could solve the aforementioned issues, this solution is not appropriate for real-time applications such as SGG. In such a context, low-cost and real-time solutions are preferred. In a recent work, Cheng et al. [195] proposed Yolo-World, a real-time Open-Vocabulary Object Detector based on YOLOV8. The key insight in Yolo-World is the alignment of a CLIP text encoder with YOLOV8 visual features through visual region-text matching. Cheng et al. modified the YOLOV8 architecture to incorporate a Vision-to-Language Path Aggregation Network (PAN) to align visual features with text features. By coupling this solution to a long training with weakly annotated data (millions of images from different object detection datasets), authors were able to propose real-time inference and open-vocabulary with state-of-the-art performance. This approach can be adapted to SGG by adding a relation detection head. We believe that using the Yolo-World architecture will also be possible for relation detection. By taking inspiration from the region-text matching, we can create a region-relation matching, where the relation detection head will be trained to match the visual features of object pairs with the text features learned. This approach can be integrated with the existing Box and Classification heads of Yolo-World, resulting in minimal computational overhead and enabling real-time inference. However, this would require a large amount of relations data to train the model, which is not available at the moment. We can draw inspiration from the data collection strategy of Yolo-World and collect a large amount of weakly annotated data using large pre-trained model such as BLIP-2 [116] or Grounding-DINO [196].

Category-Aware SGG. Current paradigm of SGG models all relations in the same embedding space. However, as we have seen through this work, relations can be of different types. To improve the performance of SGG models, it would be interesting to consider the task as a multi-task learning problem, where predicting each relation type would be a different subtask. In a recent work, Jiang et al. [197] proposed to use different Bayesian classification head for the three types of relations introduced in Neural-Motifs [38]: **Geometric**, **Possessive** and **Semantic**. The approach of Jiang et al. relies on manually defining the type of relations by predicate classes. As we have seen in Chapter 3, this is not a good approach because of the polysemy of natural language. As a result, defining multiple heads by predicate classes is not a good approach. Instead, one could split the original dataset into different sub-datasets, each containing relations of a specific type. During training, each head would be fed a different portion of the data and a context learning module will ensure the consistency of the predictions between each category-aware head. Here, the context learning will responsible for modeling the inter-dependencies between relation types (such as the fact that a **Functional** relation often implies a **Topological** relation of proximity, etc...). Other types of dependencies will be model inside each head, leading to better representations of each relation type. Finally, this type of architecture will be easy to pair with commonsense knowledge databases such as ConceptNet [97] to introduce external priors to the model and facilitate learning. The idea of bridging SGG models with external knowledge bases is not new, and has been proposed in a few works [111], [148], [198]. The issue with such work is that the external knowledge is infused systematically in the model, without taking into account the intrinsic type of the relation that we are trying to model. This often results in the insertion of irrelevant external knowledge, for instance when predicting the relation $\langle person, in\ front\ of, bike \rangle$ we will insert from the external knowledge base all relations related to bike or person such as $\langle bike, used\ for, riding \rangle$ which may confused the model. By using a Category-Aware SGG model, we can ensure that the external knowledge is only used when it is relevant to the type of relation we are trying to model, in this case a **Topological** relation.

Embodied SGG SGG is a field where real-time SGG can contribute significantly. The current paradigm of knowledge representation systems in robotics relies extensively on (1) handcrafted solutions [179], [199] or (2) non-symbolic approaches in the form of large Vision-to-Language models [200], [201]. As we have seen previously, none of these two approaches can be reasonable for low-resource applications which requires strong generalization capabilities. In fact, the first approach is not scalable and the second approach is not real-time. Here, we believe that the task of SGG can perfectly solve this gap, as we have shown in our Continuous SGG implementation for automated planning. The last missing step from our current implementation in HRC is the deployment of SGG models on embodied platforms. In a recent work [202], we proposed an implementation of YOLOV8 on the low-resource robotic platform Pepper from Aldebaran [1]. The Pepper robot is a humanoid robot that is equipped with a low-power CPU Intel ATOM® E3845 and no GPU. The memory of the robot is 4GB and the CPU is a quad-core 1.91GHz, which is very low by today's standards. We have shown that YOLOV8 can be deployed on Pepper with a latency of 300ms, which is reasonable for real-time applications. We believe that we can extend our approach to our proposed model M-PE-NET and deploy it on Pepper.

SGG for Service Robotics. Once our Continuous SGG approach is deployed onboard a robotic platform such as Pepper, we can use it in real-world tasks. In particular, we can use it for HRI in Service Robotics. Service Robotics is a subfield of Robotics targeting real-world applications of humanoid robots for daily care. As a study case, we explored the task of HRC in the context of the RoboCup@Home competition. The RoboCup@Home is the biggest international competition of Domestic Service Robotics [203]. The competition is divided into several challenges, including the *Final Challenge* where teams have to demonstrate the capabilities of their robots in a real-world scenario. This often includes HRC tasks such as setting a table, cleaning a room, or cooking a meal. In 2023, we participated in the RoboCup@Home competition, and especially in the Final challenge [202]. The task was to help someone prepare a meal using ingredients from the environment. The challenge took place in a randomly initialized kitchen-like environment. The robot had to understand the context of the scene, recognize the objects and ingredients, and instruct the human to prepare a meal. During this challenge, we used the Pepper robot to retrieve images from the scene and interact with the human. On a deported computer, we used our Global Scene Graph to analyze the video flow and predict relations. Communication with the robot is done through the Robot Operating System (ROS) framework. We created a custom algorithm on top of the GSG to retrieve objects of interest (food or ingredients) and their respective **Topological** relations with the human. We then used a simple rule-based system to instruct the human to retrieve dedicated objects. By looking at the **Functional** layer of our representation, we were able to monitor the person and especially detect if they were grasping the correct object through the *holding* relation. We were able to give

[1] `https://www.aldebaran.com/en/pepper`

a precise instruction on where to find objects (such as "the tomato soup is on the second shelf of the fridge") and to monitor the human actions precisely (such as "put the tomato soup on the table next to the plate"). The algorithm then needed to wait to detect the combination of the *on* relation between the tomato soup and the table and the *next to* relation between the tomato soup and the plate to give the next instruction. This was done by monitoring the **Topological** layer of the GSG. Thanks to this approach and other algorithms, our team RoboBreizh was able to win the RoboCup@Home competition in 2023 by scoring first [2]. This scenario was directed, as we knew in advance which object could be used by the human. However, we believe that our approach can be extended to more complex scenarios where the robot has to adapt to observed actions and objects present in the environment. This would require to develop a second module, which would work in parallel of the GSG to reason on the symbolic representation and infer the next actions to take.

SGG and Commonsense Reasoning. In another work [204], we have proposed a related architecture, where inferences are made on a similar scene graph representation thanks to the pairing with a commonsense knowledge database. In such architecture, we proposed to use the ConceptNet [97] and ATOMIC [205] databases to infer possible actions related to the current state of the Global Scene Graph and their effect on the environment. ConceptNet and ATOMIC contain relations of the type "X causes Y" or "X is a type of Y" which can be used to infer possible next states of the GSG. However, pairing a Global Scene Graph with external knowledge graphs is complicated. In a recent work, Agnese Chatti [206] takes inspiration from our work to propose a graph enrichment method for hazard detection. In this work, the scene graph is limited to spatial relations only in a static environment, without a continuous implementation. It would be then interesting to extend this proposal to our Continuous SGG approach and to use it in real-world applications such as in the RoboCup@Home competition. By extending such approach to human activities, we could detect possible hazards in the environment and prevent the human from doing dangerous actions. This will be the last perspective of our work.

[2] `https://github.com/RoboCupAtHome/Bordeaux2023`

Appendix A

Data curation and refinement for Scene Graph Generation

A.1 Annotations Comparison

In the following, we compare annotations obtained by removing irrelevant relations and using our class selection method by connectivity. Figure A.1 gives an overview of the difference in annotations between the original VG dataset (VG80K) [8], VG150 [12], our proposed data split VG150-connected and VG150-curated.

(a) VG80K (b) VG150 (c) VG150-connected (d) VG150-curated

Figure A.1: A few examples of the difference between annotations in the original dataset VG80K, VG150, VG150-connected, and VG150-curated. We can easily see that annotation from VG150-curated (right) only detail informative relations, while irrelevant annotations are heavily present in the other data splits. In addition, annotations from VG150-curated are preserving the graph structure by limiting the number of independent sub-graphs.

A.2 Triplet classification

To categorize relation triplets between **Topological**, **Functional**, **Part-Whole**, and **Attribute** categories, we fine-tuned the GPT3.5 model from OpenAI [107] on the IndoorVG dataset. Here we display the system prompt used for the fine-tuning of GPT3.5 for triplet classification:

> *You are an AI assistant with rich commonsense knowledge and strong reasoning abilities. You will be provided with a triplet formulated as (subject, predicate, object), where the predicate represents a relation between the subject and object. Your task is to categorize this triplet between the following four categories: topological, functional, part-whole, and attribute. Only answer with one of the four categories.*

Examples of triplets in the IndoorVG dataset successfully classified:

- **Topological**: (cup, on, table)
- **Functional**: (person, on, phone)

The model was fine-tuned for 1000 iterations using the OpenAI API, and the accuracy of the model during training is shown in Figure A.2 (numbers provided by OpenAI).

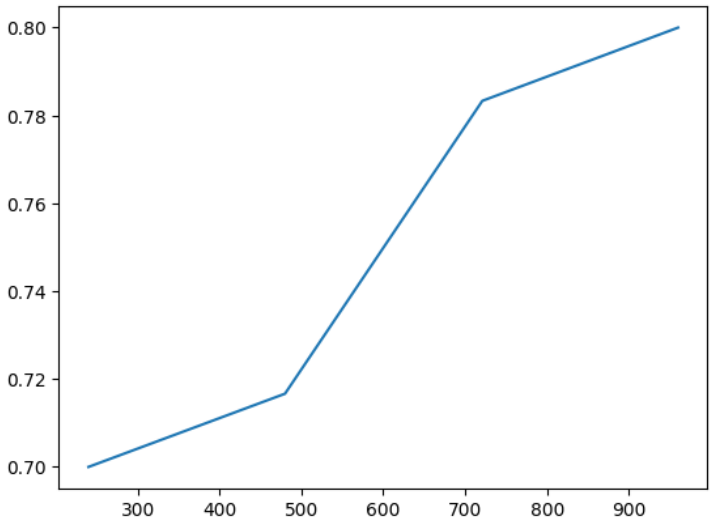


Figure A.2: Accuracy of the GPT3.5 model fine-tuned for 1000 iterations.

A.3 The IndoorVG Dataset: analytics

In the following we display the list of object and predicate classes in the IndoorVG dataset.

bag	basket	bin	blind	book
bookshelf	bottle	bowl	box	button
cabinet	camera	can	carpet	ceiling
chair	clock	computer	couch	counter
cup	curtain	desk	dishwasher	door
drawer	eyeglasses	faucet	floor	flower
food	fruit	frame	glass	hand
handle	head	jacket	key	keyboard
knob	knife	lamp	laptop	leg
lid	light	luggage	magazine	microwave
mirror	monitor	mouse	mug	oven
paper	pan	pen	person	phone
pillow	plant	plate	poster	pot
refrigerator	remote	scissors	screen	shelf
shirt	shoe	sink	speaker	stand
stove	suitcase	table	television	towel
toy	vase	wall	window	

Table A.1: List of object classes in the IndoorVG dataset.

above	against	at	attached to	behind
between	carrying	covering	cutting	drinking
eating	filled with	for	hanging from	has
holding	in	in front of	laying on	looking at
lying on	mounted on	near	of	on
playing with	reading	sitting at	sitting on	standing on
taking	talking on	under	using	watching
wearing	with			

Table A.2: List of predicate classes in the IndoorVG dataset.

In Figure A.3, we display the proportion of relation categories by predicate in the IndoorVG dataset. For each triplet, we predicted its category using the GPT3.5 model fine-tuned for triplet classification and then displayed the proportion by predicate. We can observe that fine-grained predicates (such as watching, cutting, playing with etc...) are untitled to a single category, whereas more general predicates (such as on, in, near etc...) are more versatile and can be categorized in multiple categories, depending on the subject and object of the relation.

As a comparison, we also display the proportion of relation categories by predicate in the VG150 dataset in Figure A.4. We can observe that the VG150 dataset contains more ambiguous predicates.

# Rels	# Objects	Triplets	Categories			
			Topo.	Func.	Part.	Attr.
112,804	708,409	9,095	66.15%	12.71%	11.93%	9.21%

Classes		# Rels/Image	Train/Val/Test	
Obj.	Pred.		Objects	Rels
84	37	8.18	64,098/5,069/29,159	9,538/733/4,403

Table A.3: Statistics of the **IndoorVG** dataset, Rels/Image is an average of all annotated samples. # Objects represent the total number of bounding boxes. The Train/Val/Test split is larger for objects to allow object detection training on the full original dataset.

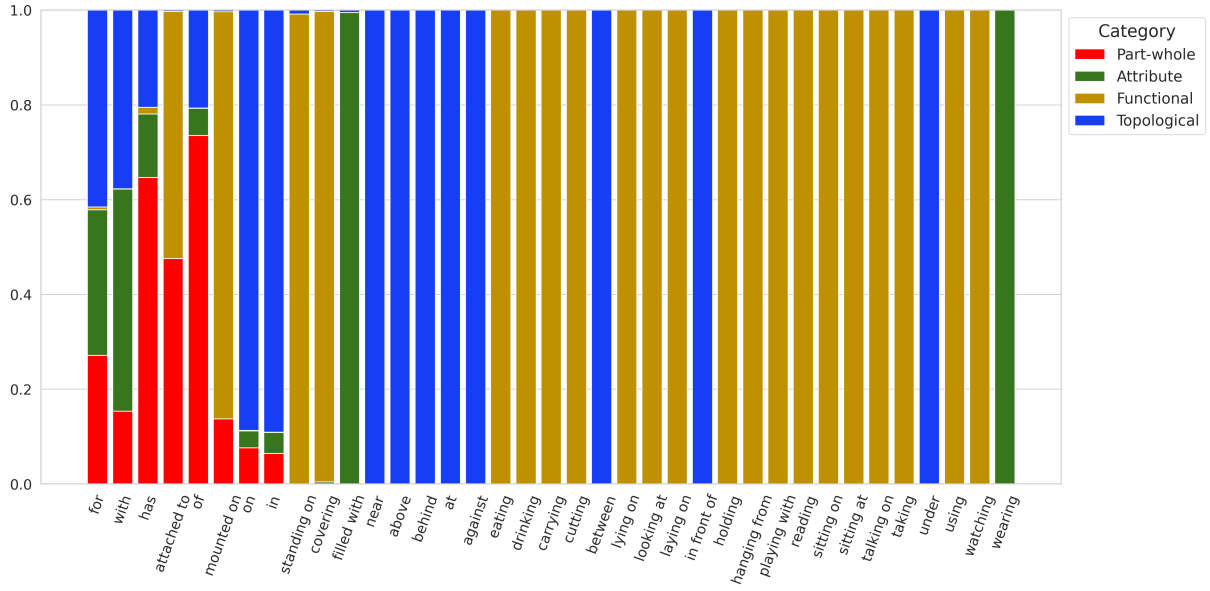


Figure A.3: Proportion of relation categories by predicate in the IndoorVG dataset.

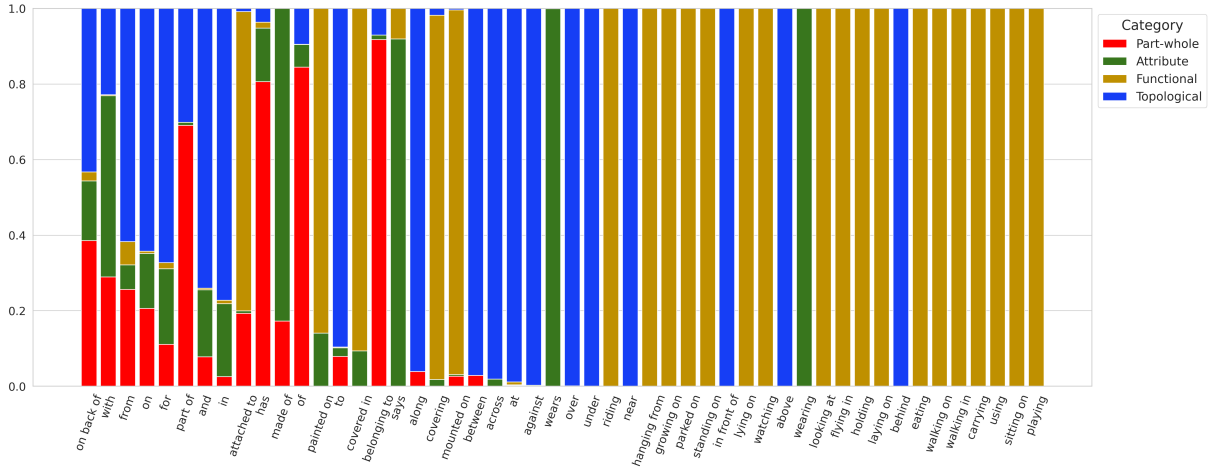


Figure A.4: Proportion of relation categories by predicate in the VG150 dataset.

Appendix B

Informativeness in Scene Graphs

B.1 Captions Generation

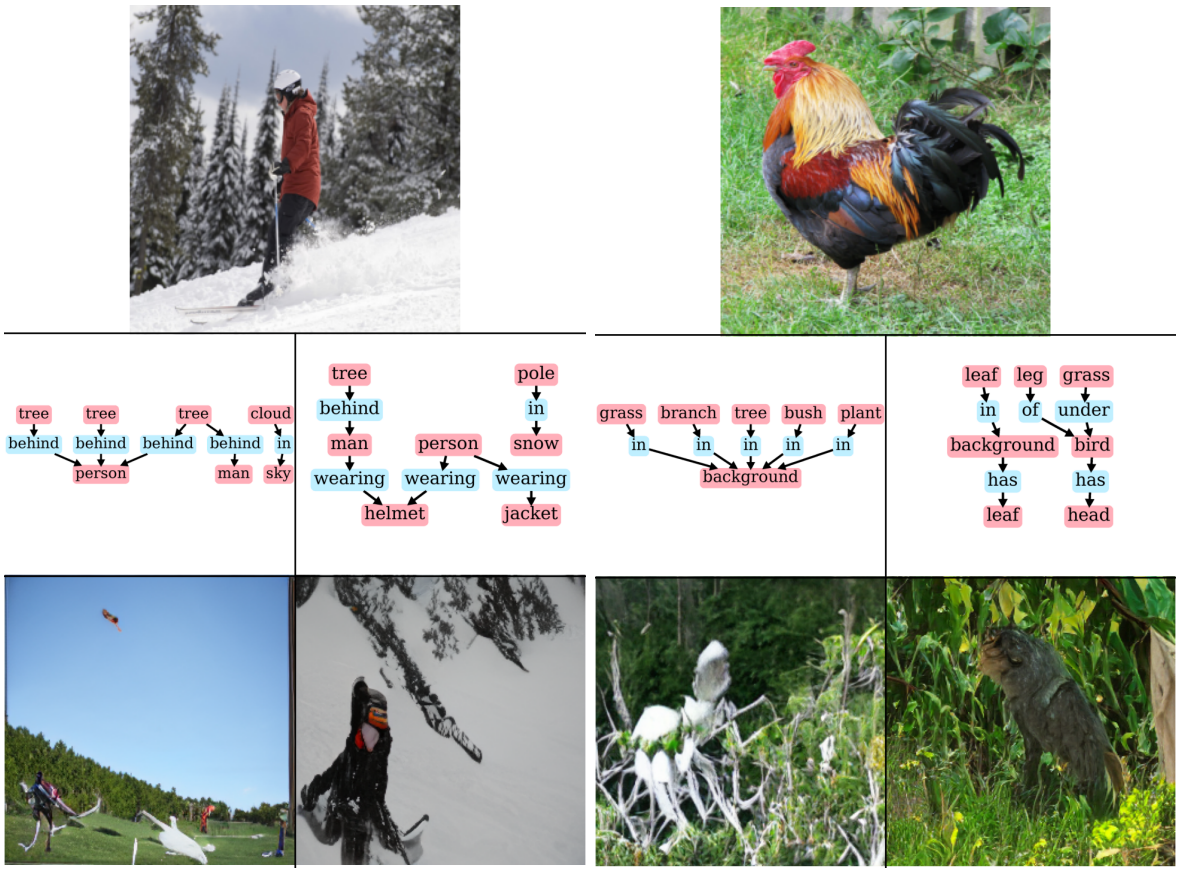
We used the *caption_coco_opt2.7b* model based on the decoder-only pre-trained model OPT with 2.7b parameters. The model was fine-tuned on the COCO dataset for image captioning. To generate good captions, we used the 5 following prompts, one by one for each image:

- "a photo of"
- "a picture of"
- "a photo showing"
- "a picture showing"
- "a photo with"

The temperature parameter was set to 0.9 to allow for more diversity in the generated captions and the minimum length to 15 for longer captions. Longer captions usually contain more relations which also prevents from having 5 very similar captions.

B.2 Image Generation From Scene Graphs

More visualization of the generated images from scene graphs can be found in this section. The layout is similar as the one from Figure 4.11 (top row: ground truth image; left: baseline graph and generated image; right: informative graph and corresponding generated image). We can see from these qualitative examples that the informative graph (right) and the corresponding generated image (right) are closer to the ground truth image (top row) than the baseline graph (left) and the corresponding generated image (left).



(a) (b)

Figure B.1: Generated images from scene graphs.

(c) (d)

Figure B.2: Generated images from scene graphs (con't).

(e) (f)

Figure B.3: Generated images from scene graphs (con't).

Appendix C

Real Time Scene Graph Generation

C.1 Experiments with YOLOV8

	Model	mR@20	mR@50	mR@100	R@20	R@50	R@100	F1@20	F1@50	F1@100
YOLOV8m	PE-NET	13.22	16.47	18.08	17.79	22.62	24.91	15.17	19.06	20.95
	GPS-NET	11.08	13.51	14.39	19.14	23.79	25.89	14.04	17.24	18.50
	Neural-Motifs	10.31	12.71	13.71	20.41	24.95	26.91	13.70	16.84	18.16
	VCTree	10.33	12.54	13.54	18.45	22.74	25.01	13.25	16.17	17.57
	Transformer	10.20	12.15	13.26	19.24	23.50	25.76	13.33	16.02	17.51
Faster-RCNN	PE-NET	7.56	9.48	10.25	10.22	12.68	14.15	8.69	10.85	11.89
	GPS-NET	5.42	6.92	7.70	9.27	12.41	14.14	6.84	8.89	9.97
	Neural-Motifs	8.16	9.95	11.11	12.30	14.45	15.79	9.81	11.78	13.04
	VCTree	7.59	9.13	9.86	11.37	13.76	15.12	9.10	10.98	11.94
	Transformer	6.48	7.71	8.76	10.87	13.27	14.64	8.12	9.76	10.96

Table C.1: Full results of our experiments with the YOLOV8 and Faster-RCNN models for SGG on the test set of the IndoorVG dataset. All results are in percentage.

	Model	IR@5	IR@10	IR@20	IR@50	IR@100	mAP@50	Latency (ms)	FPS
YOLOV8m	PE-NET	10.72	14.32	18.30	23.55	26.98	31.20	46.09 ± 1.3	21.69
	GPS-NET	11.94	16.11	20.36	25.61	29.42	31.20	50.54 ± 1.4	19.78
	Neural-Motifs	12.81	17.03	20.76	26.09	29.55	31.20	48.93 ± 1.8	20.43
	VCTree	12.23	16.31	20.28	25.16	29.11	31.20	239.51 ± 15.4	4.17
	Transformer	12.41	16.94	20.95	25.81	29.78	31.20	49.86 ± 1.1	20.05
Faster-RCNN	PE-NET	8.93	12.62	16.55	22.25	27.11	20.60	277.62 ± 25.4	3.60
	GPS-NET	4.45	6.49	8.92	13.37	16.83	14.17	383.51 ± 132.5	2.61
	Neural-Motifs	7.32	10.35	13.83	18.72	22.68	19.70	398.63 ± 133.5	2.50
	VCTree	6.96	9.99	13.25	18.24	22.06	19.20	519.53 ± 163.2	1.92
	Transformer	6.39	9.61	13.32	18.67	23.03	19.40	381.64 ± 126.7	2.62

Table C.2: (con't) Full results of our experiments with the YOLOV8 and Faster-RCNN models for SGG on the test set of the IndoorVG dataset. All results are in percentage except for Latency and Frames Per Second (FPS).

Hyperparameter	Value
Batch size	8
Learning rate	0.01
Optimizer	SGD
Loss function	CrossEntropy
Epochs	20
Learning rate scheduler	ReduceLROnPlateau
Learning rate scheduler factor	0.1
Learning rate scheduler patience	3
Learning rate scheduler threshold	0.001

Table C.3: Hyperparameters used for the experiments with the YOLOV8 and Faster-RCNN models.

Hyperparameter	Value
Batch size	32
Learning rate	0.01
Momentum	0.937
Optimizer	AdamW
Loss function	CrossEntropy
Epochs	20

Table C.4: Hyperparameters used for training the YOLOV8 model.

In this section, we display the full results of the YOLOV8 experiments for SGG on the test set of the IndoorVG dataset. These results correspond to the Figure 5.4 and Figure 5.5 in the main text. All results are in percentage.

In Table C.3, we display the hyperparameters used for the above experiments with the YOLOV8 model. To compare model with fairness, hyperparameters are the same for all the models using the YOLOV8 architecture and Faster-RCNN architecture.

Next, we display the hyperparameters used for training the YOLOV8 object detector in Table C.4. The hyperparameters are the same for all the YOLOV8 variants.

For the final training of our M-PE-NET model, we used hyperparameters tuning to find the best learning rate and momentum for the model. We used the Optuna library [1] to perform the hyperparameters tuning. We performed 50 trials to find the best learning rate and momentum with the ASHA scheduler. The grace period is 100 iterations and the maximum number of iterations is 500. The scheduler is configured to minimize the loss. The results of the hyperparameters tuning are displayed in Figure C.1. The best learning rate and momentum found are 0.016 and 0.23, respectively.

[1] `https://optuna.org/`

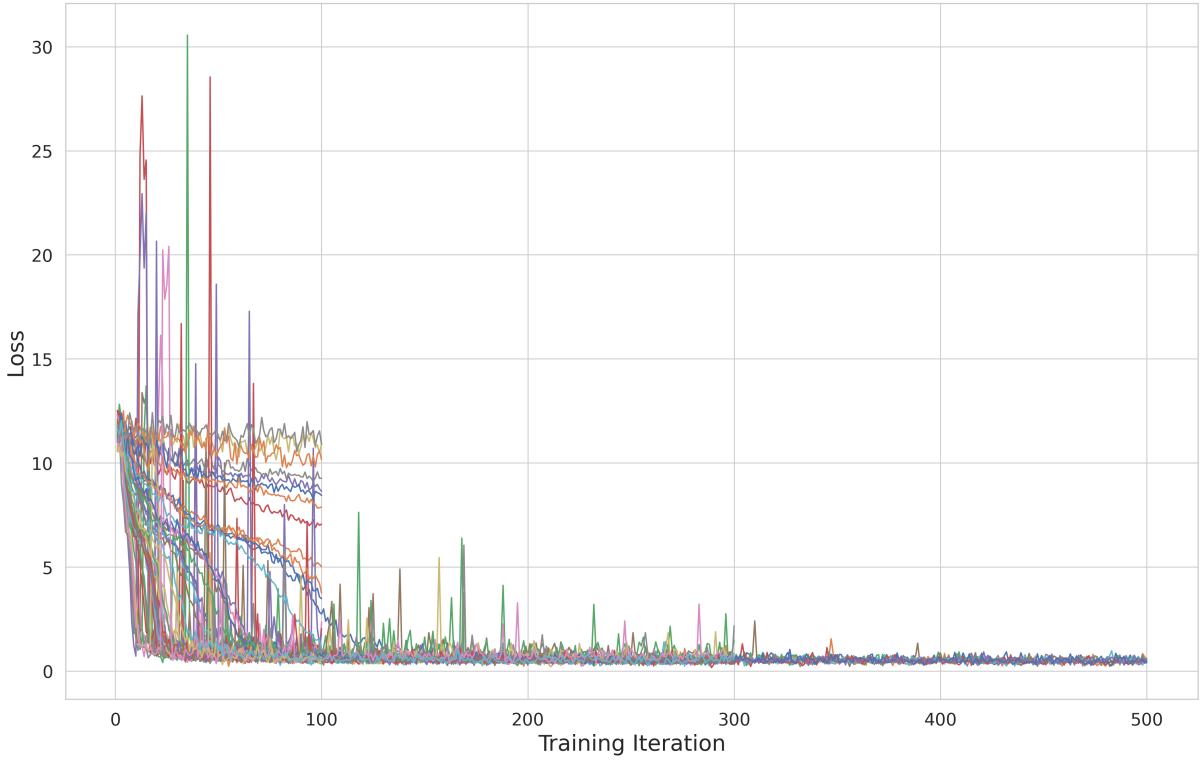


Figure C.1: Loss versus Training Iterations for 50 trials with different learning rate and momentum hyperparameters.

C.2 SGG Codebase & Open-Source

In recent years, the SGG community has relied extensively on the codebase by Tang et al. [19]. The codebase is available on GitHub[2] and has been used by several researchers to compare their models with the state-of-the-art. However, this codebase is not up-to-date, last commit was more than 2 years ago. In addition, the codebase do not incorpore latest development for the task and is limited to only a few models. We have developed a new codebase for SGG, which is available on GitHub[3]. Our codebase is the first to implement different backbones for the object detector, including YOLOV8, YOLOV9 [207], YOLOV10 [208], Yolo-World [195], RT-DETR [209], and Faster-RCNN. The YOLOV8, YOLOV9, YOLOV10, Yolo-World, and RT-DETR implementations are based on the ultralytics codebase [4]. In addition, we provide new implementations for recent approaches in SGG such as the IETrans method for internal and external data transfer [95]. Finally, our codebase can be used with different dataset such as the PSG dataset [20], VG150 [12] and IndoorVG dataset. In Table C.5, we compare our codebase with other codebases for SGG [210].

Codebase	**Datasets**	**OD Backbone**	**SGG Methods**	**Pytorch**
[38]	VG150 [12]	Faster-RCNN [139]	IMP [12], Motifs [38]	v0.3
[19]	VG150 [12]	Faster-RCNN [139]	IMP [12], Motifs [38], VCTree [37], VTransE [211], Transformer [19]	v1.2
[210]	VG150 [12], OpenImage [17]	Faster-RCNN [139]	IMP [12], MSDN [212], Motifs [38], GRCNN [213], RelDN [214]	v1.7
Ours	VG150 [12], PSG [20], GQA [15], IndoorVG [74]	Faster-RCNN [139], YOLOV8 [39], YOLOV9 [207], YOLOV10 [208], RT-DETR [209], Yolo-World [195]	IMP [12], Motifs [38], VCTree [37], VTransE [211], Transformer [19], GPS-Net [47], SHA-GCL [188], PENET [49], Squat [215], IETrans [95]	v2.2

Table C.5: Comparison of different code bases for SGG. Our codebase is the only one that provides a wide range of object detectors and SGG methods.

To our knowledge, we are the first to provide an open-source implementation of a wide range of object detector for the task of SGG, which can be used with different datasets and SGG methods. This high number of possible backbone-relation head combination makes it possible to compare different models and methods fairly. In addition, we believe that more diversity in the codebase will help the community to use SGG in a wider range of use cases. In Figure C.2,

[2] `https://github.com/KaihuaTang/Scene-Graph-Benchmark.pytorch`
[3] `https://github.com/Maelic/SGG-Benchmark`
[4] `https://github.com/ultralytics/ultralytics`

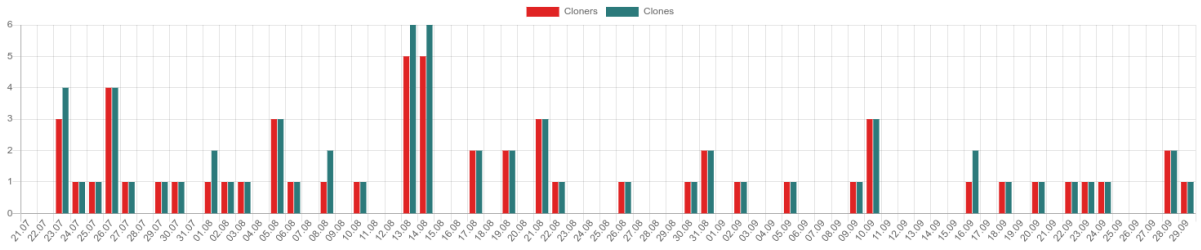


Figure C.2: GitHub analytics for the SGG-Benchmark codebase, the total number of clones is 70 from 64 different account (cloners).

we display the number of clones that our codebase has received since its open-source release the 21/07/2024. From the period between 21/07/2024 and 29/09/2024, the codebase has been visited by 643 unique visitors, and 70 clones have been made. In addition, 32 issues have been opened by the community, which shows a growing interest. Our codebase also provide a tutorial for hyperparameters tuning and a tutorial for integration in other pipelines with visualization and demos. We are committed to maintaining and updating the codebase with new models and methods.

BIBLIOGRAPHY

[1] N. Xu, A.-A. Liu, J. Liu, W. Nie, and Y. Su, "Scene graph captioner: Image captioning based on structural visual representation", *Journal of Visual Communication and Image Representation*, vol. 58, pp. 477–485, Jan. 2019, ISSN: 10473203. DOI: 10.1016/j.jvcir.2018.12.027. Accessed: Sep. 18, 2024.

[2] Y. Zhong, L. Wang, J. Chen, D. Yu, and Y. Li, "Comprehensive Image Captioning via Scene Graph Decomposition", *in Computer Vision at ECCV 2020*, A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, Eds., vol. 12359, Cham: Springer International Publishing, 2020, pp. 211–229, ISBN: 978-3-030-58567-9 978-3-030-58568-6. DOI: 10.1007/978-3-030-58568-6_13. Accessed: Dec. 3, 2023.

[3] S. He, H. R. Tavakoli, A. Borji, and N. Pugeault, "Human Attention in Image Captioning: Dataset and Analysis", *in 2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, Seoul, Korea (South): IEEE, Oct. 2019, pp. 8528–8537, ISBN: 978-1-72814-803-8. DOI: 10.1109/ICCV.2019.00862. Accessed: Dec. 4, 2023.

[4] W. Liang, Y. Jiang, and Z. Liu, "GraghVQA: Language-Guided Graph Neural Networks for Graph-based Visual Question Answering", *in Proceedings of the Third Workshop on Multimodal Artificial Intelligence*, A. Zadeh et al., Eds., Mexico City, Mexico: Association for Computational Linguistics, Jun. 2021, pp. 79–86. DOI: 10.18653/v1/2021.maiworkshop-1.12. Accessed: Nov. 27, 2023.

[5] S. Antol et al., "Vqa: visual question answering", *in Proceedings of the IEEE international conference on computer vision*, 2015, pp. 2425–2433.

[6] P. Anderson et al., "Bottom-Up and Top-Down Attention for Image Captioning and Visual Question Answering", *in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 6077–6086. Accessed: Sep. 6, 2024.

[7] C. Lu, R. Krishna, M. Bernstein, and L. Fei-Fei, "Visual Relationship Detection with Language Priors", *in Computer Vision - ECCV 2016*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds., vol. 9905, Cham: Springer International Publishing, 2016, pp. 852–869, ISBN: 978-3-319-46447-3 978-3-319-46448-0. DOI: 10.1007/978-3-319-46448-0_51. Accessed: Jan. 25, 2022.

[8] R. Krishna et al., “Visual Genome: Connecting Language and Vision Using Crowdsourced Dense Image Annotations”, *International Journal of Computer Vision*, vol. 123, *1*, pp. 32–73, May 1, 2017, ISSN: 1573-1405. DOI: 10.1007/s11263-016-0981-7. Accessed: Nov. 8, 2022.

[9] X. Li, D. Guo, H. Liu, and F. Sun, “Embodied Semantic Scene Graph Generation”, *in Proceedings of the 5th Conference on Robot Learning*, PMLR, Jan. 2022, pp. 1585–1594. Accessed: Sep. 18, 2024.

[10] K. P. Singh, J. Salvador, L. Weihs, and A. Kembhavi, “Scene Graph Contrastive Learning for Embodied Navigation”, *in 2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, Paris, France: IEEE, Oct. 2023, pp. 10 850–10 860, ISBN: 9798350307184. DOI: 10.1109/ICCV51070.2023.00999. Accessed: Sep. 18, 2024.

[11] F. Amodeo, F. Caballero, N. DÃaz-RodrÃguez, and L. Merino, “OG-SGG: Ontology-Guided Scene Graph Generationâ€”A Case Study in Transfer Learning for Telepresence Robotics”, *IEEE Access*, vol. 10, pp. 132 564–132 583, 2022, ISSN: 2169-3536. DOI: 10.1109/ACCESS.2022.3230590.

[12] D. Xu, Y. Zhu, C. B. Choy, and L. Fei-Fei, “Scene Graph Generation by Iterative Message Passing”, *in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5410–5419. Accessed: Sep. 2, 2023.

[13] R. Yu, A. Li, V. I. Morariu, and L. S. Davis, “Visual Relationship Detection with Internal and External Linguistic Knowledge Distillation”, *in 2017 IEEE International Conference on Computer Vision (ICCV)*, Venice: IEEE, Oct. 2017, pp. 1068–1076, ISBN: 978-1-5386-1032-9. DOI: 10.1109/ICCV.2017.121. Accessed: Jan. 17, 2023.

[14] K. Nguyen, S. Tripathi, B. Du, T. Guha, and T. Q. Nguyen, “In Defense of Scene Graphs for Image Captioning”, *in Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 1407–1416. Accessed: Jun. 27, 2024.

[15] D. A. Hudson and C. D. Manning, *GQA: A New Dataset for Real-World Visual Reasoning and Compositional Question Answering*, May 2019. arXiv: 1902.09506 [cs]. Accessed: Nov. 27, 2023.

[16] X. Chang, P. Ren, P. Xu, Z. Li, X. Chen, and A. Hauptmann, “A Comprehensive Survey of Scene Graphs: Generation and Application”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, *1*, pp. 1–26, Jan. 2023, ISSN: 0162-8828, 2160-9292, 1939-3539. DOI: 10.1109/TPAMI.2021.3137605. Accessed: Oct. 20, 2023.

[17] A. Kuznetsova et al., “The open images dataset v4: unified image classification, object detection, and visual relationship detection at scale”, *International journal of computer vision*, vol. 128, *7*, pp. 1956–1981, 2020.

[18] Y. Liang, Y. Bai, W. Zhang, X. Qian, L. Zhu, and T. Mei, "VrR-VG: Refocusing Visually-Relevant Relationships", *in 2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, Seoul, Korea (South): IEEE, Oct. 2019, pp. 10 402–10 411, ISBN: 978-1-72814-803-8. DOI: 10.1109/ICCV.2019.01050. Accessed: Nov. 29, 2022.

[19] K. Tang, Y. Niu, J. Huang, J. Shi, and H. Zhang, "Unbiased Scene Graph Generation From Biased Training", *in 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle, WA, USA: IEEE, Jun. 2020, pp. 3713–3722, ISBN: 978-1-72817-168-5. DOI: 10.1109/CVPR42600.2020.00377. Accessed: Oct. 10, 2022.

[20] J. Yang, Y. Z. Ang, Z. Guo, K. Zhou, W. Zhang, and Z. Liu, "Panoptic Scene Graph Generation", *in Computer Vision – ECCV 2022*, S. Avidan, G. Brostow, M. Cissé, G. M. Farinella, and T. Hassner, Eds., ser. Lecture Notes in Computer Science, Cham: Springer Nature Switzerland, 2022, pp. 178–196, ISBN: 978-3-031-19812-0. DOI: 10.1007/978-3-031-19812-0_11.

[21] K. Yoon, K. Kim, J. Moon, and C. Park. "Unbiased Heterogeneous Scene Graph Generation with Relation-aware Message Passing Neural Network". version 1. arXiv: 2212.00443 [cs], Accessed: Dec. 21, 2022, preprint.

[22] J. Yu, Y. Chai, Y. Wang, Y. Hu, and Q. Wu, "CogTree: Cognition Tree Loss for Unbiased Scene Graph Generation", *in Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence*, Montreal, Canada: International Joint Conferences on Artificial Intelligence Organization, Aug. 2021, pp. 1274–1280, ISBN: 978-0-9992411-9-6. DOI: 10.24963/ijcai.2021/176. Accessed: Jan. 2, 2023.

[23] M.-J. Chiou, H. Ding, H. Yan, C. Wang, R. Zimmermann, and J. Feng, "Recovering the Unbiased Scene Graphs from the Biased Ones", *in Proceedings of the 29th ACM International Conference on Multimedia*, Virtual Event China: ACM, Oct. 17, 2021, pp. 1581–1590, ISBN: 978-1-4503-8651-7. DOI: 10.1145/3474085.3475297. Accessed: May 24, 2024.

[24] S. Woo, J. Noh, and K. Kim, "Tackling the Challenges in Scene Graph Generation With Local-to-Global Interactions", *IEEE Transactions on Neural Networks and Learning Systems*, vol. 34, *12*, pp. 9713–9726, Dec. 2023, ISSN: 2162-2388. DOI: 10.1109/TNNLS.2022.3159990. Accessed: Sep. 19, 2024.

[25] S. V. Nuthalapati et al., "Lightweight Visual Question Answering using Scene Graphs", *in Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, Virtual Event Queensland Australia: ACM, Oct. 2021, pp. 3353–3357, ISBN: 978-1-4503-8446-9. DOI: 10.1145/3459637.3482218. Accessed: Nov. 27, 2023.

[26] T. Jin et al., "Fast Contextual Scene Graph Generation With Unbiased Context Augmentation", presented at the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 6302–6311. Accessed: Jun. 19, 2024.

[27] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection", *in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 779–788. Accessed: Jul. 11, 2024.

[28] P. Mathur, A. Gill, A. Yadav, A. Mishra, and N. K. Bansode, "Camera2Caption: A real-time image caption generator", *in 2017 International Conference on Computational Intelligence in Data Science(ICCIDS)*, Jun. 2017, pp. 1–6. DOI: 10.1109/ICCIDS.2017.8272660. Accessed: Sep. 19, 2024.

[29] S. Li, P. Zheng, Z. Wang, J. Fan, and L. Wang, "Dynamic Scene Graph for Mutual-Cognition Generation in Proactive Human-Robot Collaboration", *Procedia CIRP*, Leading Manufacturing Systems Transformation - Proceedings of the 55th CIRP Conference on Manufacturing Systems 2022, vol. 107, pp. 943–948, Jan. 2022, ISSN: 2212-8271. DOI: 10.1016/j.procir.2022.05.089. Accessed: Sep. 16, 2024.

[30] G. Jung, J. Lee, and I. Kim, "Tracklet Pair Proposal and Context Reasoning for Video Scene Graph Generation", *Sensors*, vol. 21, *9*, p. 3164, Jan. 2021, ISSN: 1424-8220. DOI: 10.3390/s21093164. Accessed: Nov. 28, 2021.

[31] Y. Teng, L. Wang, Z. Li, and G. Wu, "Target Adaptive Context Aggregation for Video Scene Graph Generation", *in Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 13 688–13 697. Accessed: Sep. 2, 2022.

[32] W. Wang, R. Wang, S. Shan, and X. Chen, "Sketching Image Gist: Human-Mimetic Hierarchical Scene Graph Generation", *in Computer Vision - ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIII*, Berlin, Heidelberg: Springer-Verlag, Aug. 23, 2020, pp. 222–239, ISBN: 978-3-030-58600-3. DOI: 10.1007/978-3-030-58601-0_14. Accessed: Jan. 11, 2023.

[33] A. Oliva, "Gist of the scene", *in Neurobiology of attention*, Elsevier, 2005, pp. 251–256.

[34] L. Fei-Fei, A. Iyer, C. Koch, and P. Perona, "What do we perceive in a glance of a real-world scene?", *Journal of vision*, vol. 7, *1*, pp. 10–10, 2007.

[35] W. Wang, R. Wang, S. Shan, and X. Chen, "Importance First: Generating Scene Graph of Human Interest", *International Journal of Computer Vision*, vol. 131, *10*, pp. 2489–2515, Oct. 1, 2023, ISSN: 1573-1405. DOI: 10.1007/s11263-023-01817-7. Accessed: Mar. 6, 2024.

[36] L. Yang et al., *Diffusion-Based Scene Graph to Image Generation with Masked Contrastive Pre-Training*, Nov. 2022. DOI: 10.48550/arXiv.2211.11138. arXiv: 2211.11138 [cs]. Accessed: Feb. 9, 2023.

[37] K. Tang, H. Zhang, B. Wu, W. Luo, and W. Liu, "Learning to Compose Dynamic Tree Structures for Visual Contexts", *in 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA: IEEE, Jun. 2019, pp. 6612–6621, ISBN: 978-1-72813-293-8. DOI: 10.1109/CVPR.2019.00678. Accessed: Jan. 20, 2023.

[38] R. Zellers, M. Yatskar, S. Thomson, and Y. Choi, "Neural Motifs: Scene Graph Parsing with Global Context", *in 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT: IEEE, Jun. 2018, pp. 5831–5840, ISBN: 978-1-5386-6420-9. DOI: 10.1109/CVPR.2018.00611. Accessed: Nov. 14, 2022.

[39] G. Jocher, A. Chaurasia, and J. Qiu, *Ultralytics YOLO*, version 8.0.0, Jan. 2023. [Online]. Available: https://github.com/ultralytics/ultralytics.

[40] M. Ghallab et al., "PDDL - The Planning Domain Definition Language", Aug. 1998.

[41] D. Beckett and B. McBride, "Rdf/xml syntax specification (revised)", *W3C recommendation*, vol. 10, *2.3*, 2004.

[42] S. Thrun, "Probabilistic robotics", *Communications of the ACM*, vol. 45, *3*, pp. 52–57, 2002.

[43] S. Lemaignan, R. Ros, L. Mösenlechner, R. Alami, and M. Beetz, "Oro, a knowledge management platform for cognitive architectures in robotics", *in 2010 IEEE/RSJ International conference on intelligent robots and systems*, IEEE, 2010, pp. 3548–3553.

[44] A. Das et al., "Visual dialog", *in Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 326–335.

[45] T.-Y. Lin et al., "Microsoft COCO: Common Objects in Context", *in Computer Vision - ECCV 2014*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds., Cham: Springer International Publishing, 2014, pp. 740–755, ISBN: 978-3-319-10602-1. DOI: 10.1007/978-3-319-10602-1_48.

[46] J. Pennington, R. Socher, and C. Manning, "GloVe: Global Vectors for Word Representation", *in Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, A. Moschitti, B. Pang, and W. Daelemans, Eds., Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 1532–1543. DOI: 10.3115/v1/D14-1162. Accessed: Jun. 20, 2024.

[47] X. Lin, C. Ding, J. Zeng, and D. Tao, "GPS-Net: Graph Property Sensing Network for Scene Graph Generation", *in 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle, WA, USA: IEEE, Jun. 2020, pp. 3743–3752, ISBN: 978-1-72817-168-5. DOI: 10.1109/CVPR42600.2020.00380. Accessed: Oct. 10, 2022.

[48] A. Vaswani et al., "Attention is All you Need", *in Advances in Neural Information Processing Systems*, vol. 30, Curran Associates, Inc., 2017. Accessed: Oct. 18, 2021.

[49] C. Zheng, X. Lyu, L. Gao, B. Dai, and J. Song, "Prototype-Based Embedding Network for Scene Graph Generation", *in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 22 783–22 792. Accessed: Apr. 19, 2024.

[50] J. M. Mandler and R. E. Parker, "Memory for descriptive and spatial information in complex pictures.", *Journal of experimental psychology: Human learning and memory*, vol. 2, *1*, p. 38, 1976.

[51] I. Biederman, "On the semantics of a glance at a scene", *in Perceptual organization*, Routledge, 1981, pp. 213–253.

[52] T. Sanocki, T. Nguyen, S. Shultz, and J. Defant, "Novel scene understanding, from gist to elaboration", *Visual Cognition*, vol. 31, *3*, pp. 188–215, Mar. 2023, ISSN: 1350-6285. DOI: 10.1080/13506285.2023.2221047. Accessed: Jan. 30, 2024.

[53] M. C. Potter, "Short-term conceptual memory for pictures.", *Journal of experimental psychology: human learning and memory*, vol. 2, *5*, p. 509, 1976.

[54] J. Johnson et al., "Image retrieval using scene graphs", *in 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, USA: IEEE, Jun. 2015, pp. 3668–3678, ISBN: 978-1-4673-6964-0. DOI: 10.1109/CVPR.2015.7298990. Accessed: Jun. 23, 2022.

[55] S. Yoon et al., "Image-to-Image Retrieval by Learning Similarity between Scene Graphs", *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, *12*, pp. 10 718–10 726, May 2021, ISSN: 2374-3468. DOI: 10.1609/aaai.v35i12.17281. Accessed: Sep. 27, 2024.

[56] J. Yang, J. Lu, S. Lee, D. Batra, and D. Parikh, "Graph R-CNN for Scene Graph Generation", *in Computer Vision – ECCV 2018*, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds., vol. 11205, Cham: Springer International Publishing, 2018, pp. 690–706, ISBN: 978-3-030-01245-8 978-3-030-01246-5. Accessed: Dec. 28, 2022.

[57] M. Khademi and O. Schulte, "Deep Generative Probabilistic Graph Neural Networks for Scene Graph Generation", *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, *07*, pp. 11 237–11 245, Apr. 2020, ISSN: 2374-3468. DOI: 10.1609/aaai.v34i07.6783. Accessed: Sep. 27, 2024.

[58] V. Damodaran et al., *Understanding the Role of Scene Graphs in Visual Question Answering*, Jan. 2021. arXiv: 2101.05479 [cs]. Accessed: Nov. 27, 2023.

[59] M. Z. Hossain, F. Sohel, M. F. Shiratuddin, and H. Laga, "A Comprehensive Survey of Deep Learning for Image Captioning", *ACM Comput. Surv.*, vol. 51, *6*, 118:1–118:36, Feb. 2019, ISSN: 0360-0300. DOI: 10.1145/3295748. Accessed: Sep. 28, 2024.

[60] J. Jia et al., "Image captioning based on scene graphs: A survey", *Expert Systems with Applications*, vol. 231, p. 120 698, Nov. 2023, ISSN: 09574174. DOI: 10.1016/j.eswa.2023.120698. Accessed: Nov. 15, 2023.

[61] D. Wang, D. Beck, and T. Cohn, "On the Role of Scene Graphs in Image Captioning", *in Proceedings of the Beyond Vision and LANguage: inTEgrating Real-world kNowledge (LANTERN)*, A. Mogadala, D. Klakow, S. Pezzelle, and M.-F. Moens, Eds., Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 29–34. DOI: 10.18653/v1/D19-6405. Accessed: Nov. 30, 2023.

[62] V. Milewski, M.-F. Moens, and I. Calixto, "Are Scene Graphs Good Enough to Improve Image Captioning?", *in Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, K.-F. Wong, K. Knight, and H. Wu, Eds., Suzhou, China: Association for Computational Linguistics, Dec. 2020, pp. 504–515. Accessed: Dec. 19, 2023.

[63] X. Yang et al., *Transforming Visual Scene Graphs to Image Captions*, May 2023. arXiv: 2305.02177 [cs]. Accessed: Dec. 7, 2023.

[64] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-Resolution Image Synthesis With Latent Diffusion Models", *in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 10 684–10 695. Accessed: Sep. 28, 2024.

[65] J. Johnson, A. Gupta, and L. Fei-Fei, "Image Generation from Scene Graphs", *in 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT: IEEE, Jun. 2018, pp. 1219–1228, ISBN: 978-1-5386-6420-9. DOI: 10.1109/CVPR.2018.00133. Accessed: Feb. 9, 2023.

[66] J. Gu, H. Zhao, Z. Lin, S. Li, J. Cai, and M. Ling, "Scene Graph Generation With External Knowledge and Image Reconstruction", *in 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA: IEEE, Jun. 2019, pp. 1969–1978, ISBN: 978-1-72813-293-8. DOI: 10.1109/CVPR.2019.00207. Accessed: Sep. 20, 2022.

[67] S. Y. Gadre, K. Ehsani, S. Song, and R. Mottaghi, "Continuous Scene Representations for Embodied AI", *in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 14 849–14 859. Accessed: Sep. 2, 2022.

[68] X. Li, D. Guo, H. Liu, and F. Sun, "Embodied Semantic Scene Graph Generation", *in Proceedings of the 5th Conference on Robot Learning*, PMLR, Jan. 2022, pp. 1585–1594. Accessed: Sep. 5, 2022.

[69] S. Amiri, K. Chandan, and S. Zhang, "Reasoning With Scene Graphs for Robot Planning Under Partial Observability", *IEEE Robotics and Automation Letters*, vol. 7, *2*, pp. 5560–5567, Apr. 2022, ISSN: 2377-3766. DOI: 10.1109/LRA.2022.3157567. Accessed: Sep. 28, 2024.

[70] C. Agia et al., *TASKOGRAPHY: Evaluating robot task planning over large 3D scene graphs*, Jul. 2022. arXiv: 2207.05006 [cs]. Accessed: Jul. 19, 2023.

[71] M. Tenorth and M. Beetz, "KnowRob: A knowledge processing infrastructure for cognition-enabled robots", *The International Journal of Robotics Research*, vol. 32, *5*, pp. 566–590, Apr. 2013, Publisher: SAGE Publications Ltd STM, ISSN: 0278-3649.

[72] M. Beetz, D. Bessler, A. Haidu, M. Pomarlan, A. K. Bozcuoglu, and G. Bartels, "Know Rob 2.0 : A 2nd Generation Knowledge Processing Framework for Cognition-Enabled Robotic Agents", en, *in 2018 IEEE International Conference on Robotics and Automation (ICRA)*, Brisbane, QLD: IEEE, May 2018, pp. 512–519, ISBN: 978-1-5386-3081-5.

[73] M. Neau, P. E. Santos, A.-G. Bosser, and C. Buche, "Fine-Grained is Too Coarse: A Novel Data-Centric Approach for Efficient Scene Graph Generation", *in Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 11–20. Accessed: Sep. 23, 2024.

[74] M. Neau, P. Santos, A.-G. Bosser, and C. Buche, "In Defense of Scene Graph Generation for Human-Robot Open-Ended Interaction in Service Robotics", *in RoboCup 2023: Robot World Cup XXVI*, C. Buche, A. Rossi, M. SimÃµes, and U. Visser, Eds., vol. 14140, Cham: Springer Nature Switzerland, 2024, pp. 299–310, ISBN: 978-3-031-55014-0 978-3-031-55015-7. DOI: 10.1007/978-3-031-55015-7_25. Accessed: Sep. 23, 2024.

[75] D. A. Chacra and J. Zelek, "The Topology and Language of Relationships in the Visual Genome Dataset", *in 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, New Orleans, LA, USA: IEEE, Jun. 2022, pp. 4859–4867, ISBN: 978-1-66548-739-9. DOI: 10.1109/CVPRW56347.2022.00533. Accessed: Feb. 28, 2023.

[76] I. Misra, C. L. Zitnick, M. Mitchell, and R. Girshick, "Seeing through the Human Reporting Bias: Visual Classifiers from Noisy Human-Centric Labels", *in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA: IEEE, Jun. 2016, pp. 2930–2939, ISBN: 978-1-4673-8851-1. DOI: 10.1109/CVPR.2016.320. Accessed: Sep. 9, 2024.

[77] C. Lu, R. Krishna, M. Bernstein, and L. Fei-Fei, "Visual Relationship Detection with Language Priors", *in Computer Vision - ECCV 2016*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds., Cham: Springer International Publishing, 2016, pp. 852–869, ISBN: 978-3-319-46448-0. DOI: 10.1007/978-3-319-46448-0_51.

[78] J. Zhang, Y. Kalantidis, M. Rohrbach, M. Paluri, A. Elgammal, and M. Elhoseiny. "Large-Scale Visual Relationship Understanding". arXiv: 1804.10660 [cs], Accessed: Jan. 12, 2023, preprint.

[79] R. Li, S. Zhang, and X. He. "SGTR+: End-to-end Scene Graph Generation with Transformer". arXiv: 2401.12835 [cs], Accessed: Jun. 6, 2024, preprint.

[80] Y.-W. Chao, Y. Liu, X. Liu, H. Zeng, and J. Deng, "Learning to Detect Human-Object Interactions", *in 2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, Mar. 2018, pp. 381–389. DOI: 10.1109/WACV.2018.00048. Accessed: Jul. 2, 2024.

[81] S. Gupta and J. Malik, *Visual Semantic Role Labeling*, May 2015. DOI: 10.48550/arXiv.1505.04474. arXiv: 1505.04474 [cs]. Accessed: Jul. 2, 2024.

[82] D. Damen et al., "Scaling Egocentric Vision: The EPIC-KITCHENS Dataset", *in Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 720–736. Accessed: Jul. 2, 2024.

[83] I. Armeni et al., "3D Scene Graph: A Structure for Unified Semantics, 3D Space, and Camera", *in Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 5664–5673. Accessed: Jul. 2, 2024.

[84] J. Wald, H. Dhamo, N. Navab, and F. Tombari, "Learning 3D Semantic Scene Graphs From 3D Indoor Reconstructions", *in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 3961–3970. Accessed: Jul. 2, 2024.

[85] J. Ji, R. Krishna, L. Fei-Fei, and J. C. Niebles, "Action Genome: Actions As Compositions of Spatio-Temporal Scene Graphs", *in 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle, WA, USA: IEEE, Jun. 2020, pp. 10 233–10 244, ISBN: 978-1-72817-168-5. DOI: 10.1109/CVPR42600.2020.01025. Accessed: Jun. 28, 2022.

[86] S. Nag, K. Min, S. Tripathi, and A. K. R. Chowdhury, *Unbiased Scene Graph Generation in Videos*, Jun. 2023. arXiv: 2304.00733 [cs]. Accessed: Jul. 2, 2024.

[87] B. Wen, J. Luo, X. Liu, and L. Huang. "Unbiased Scene Graph Generation via Rich and Fair Semantic Extraction". arXiv: 2002.00176 [cs], Accessed: Jan. 10, 2023, preprint.

[88] X. Li, L. Chen, J. Shao, S. Xiao, S. Zhang, and J. Xiao. "Rethinking the Evaluation of Unbiased Scene Graph Generation". arXiv: 2208.01909 [cs], Accessed: Jul. 18, 2023, preprint.

[89] L. Li, G. Chen, J. Xiao, Y. Yang, C. Wang, and L. Chen. "Compositional Feature Augmentation for Unbiased Scene Graph Generation". arXiv: 2308.06712 [cs], Accessed: Sep. 1, 2023, preprint.

[90] S. Yan et al., "PCPL: Predicate-Correlation Perception Learning for Unbiased Scene Graph Generation", *in Proceedings of the 28th ACM International Conference on Multimedia*, ser. MM '20, New York, NY, USA: Association for Computing Machinery, Oct. 12, 2020, pp. 265–273, ISBN: 978-1-4503-7988-5. DOI: 10.1145/3394171.3413722. Accessed: May 14, 2024.

[91] D. Liu, M. Bober, and J. Kittler, "Importance Weighted Structure Learning for Scene Graph Generation", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 46, *2*, pp. 1231–1242, Feb. 2024, ISSN: 0162-8828, 2160-9292, 1939-3539. DOI: 10.1109/TPAMI.2023.3329339. Accessed: Apr. 19, 2024.

[92] L. Gao et al. "Informative Scene Graph Generation via Debiasing". arXiv: 2308.05286 [cs], Accessed: Oct. 23, 2023, preprint.

[93] G. Zhu et al., *Scene Graph Generation: A Comprehensive Survey*, Jun. 2022. arXiv: 2201.00443 [cs]. Accessed: Sep. 5, 2022.

[94] F. Plesse, A. Ginsca, B. Delezoide, and F. Preteux, "Focusing Visual Relation Detection on Relevant Relations with Prior Potentials", *in 2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*, Snowmass Village, CO, USA: IEEE, Mar. 2020, pp. 2969–2978, ISBN: 978-1-72816-553-0. DOI: 10.1109/WACV45572.2020.9093605. Accessed: Jan. 17, 2023.

[95] A. Zhang et al. "Fine-Grained Scene Graph Generation with Data Transfer". arXiv: 2203.11654 [cs], Accessed: Feb. 4, 2023, preprint.

[96] C. Fellbaum, Ed., *WordNet: An Electronic Lexical Database* (Language, Speech, and Communication). Cambridge, MA: MIT Press, 1998, ISBN: 978-0-262-06197-1.

[97] R. Speer, J. Chin, and C. Havasi, "Conceptnet 5.5: an open multilingual graph of general knowledge", *in Thirty-first AAAI conference on artificial intelligence*, 2017.

[98] F. Ilievski, A. Oltramari, K. Ma, B. Zhang, D. L. McGuinness, and P. Szekely, "Dimensions of commonsense knowledge", *Knowledge-Based Systems*, vol. 229, p. 107 347, 2021.

[99] N. Reimers and I. Gurevych, "Sentence-bert: sentence embeddings using siamese bert-networks", *in Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, Nov. 2019. [Online]. Available: https://arxiv.org/abs/1908.10084.

[100] J. M. Henderson, T. R. Hayes, C. E. Peacock, and G. Rehrig, "Meaning and Attentional Guidance in Scenes: A Review of the Meaning Map Approach", *Vision*, vol. 3, *2*, p. 19, May 2019, ISSN: 2411-5150. DOI: 10.3390/vision3020019. Accessed: Oct. 30, 2023.

[101] R. Li, S. Zhang, B. Wan, and X. He, "Bipartite Graph Network with Adaptive Message Passing for Unbiased Scene Graph Generation", *in 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Nashville, TN, USA: IEEE, Jun. 2021, pp. 11 104–11 114, ISBN: 978-1-66544-509-2. DOI: 10.1109/CVPR46437.2021.01096. Accessed: Oct. 2, 2022.

[102] J. Ortigosa-HernÃ¡ndez, I. Inza, and J. A. Lozano, "Measuring the class-imbalance extent of multi-class problems", *Pattern Recognition Letters*, vol. 98, pp. 32–38, Oct. 2017, ISSN: 0167-8655. DOI: 10.1016/j.patrec.2017.08.002. Accessed: Jun. 20, 2024.

[103] R. Zhu, Z. Wang, Z. Ma, G. Wang, and J.-H. Xue, "LRID: A new metric of multi-class imbalance degree based on likelihood-ratio test", *Pattern Recognition Letters*, vol. 116, pp. 36–42, Dec. 2018, ISSN: 01678655. DOI: 10.1016/j.patrec.2018.09.012. Accessed: Feb. 9, 2023.

[104] S. Cai, L. Qiu, X. Chen, Q. Zhang, and L. Chen, "Semantic-Enhanced Image Clustering", *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, *6*, pp. 6869–6878, Jun. 2023, ISSN: 2374-3468. DOI: 10.1609/aaai.v37i6.25841. Accessed: Jun. 20, 2024.

[105] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne.", *Journal of machine learning research*, vol. 9, *11*, 2008.

[106] L. Li, L. Chen, Y. Huang, Z. Zhang, S. Zhang, and J. Xiao, "The Devil is in the Labels: Noisy Label Correction for Robust Scene Graph Generation", *in 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, New Orleans, LA, USA: IEEE, Jun. 2022, pp. 18 847–18 856, ISBN: 978-1-66546-946-3. DOI: 10.1109/CVPR52688.2022.01830. Accessed: Jan. 27, 2023.

[107] OpenAI, *Chatgpt: chat generative pre-trained transformer 3.5*, Large language model, 2022. [Online]. Available: https://www.openai.com/research/chatgpt.

[108] N. Bian et al., "ChatGPT Is a Knowledgeable but Inexperienced Solver: An Investigation of Commonsense Problem in Large Language Models", *in Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, N. Calzolari, M.-Y. Kan, V. Hoste, A. Lenci, S. Sakti, and N. Xue, Eds., Torino, Italia: ELRA and ICCL, May 2024, pp. 3098–3110. Accessed: Jun. 20, 2024.

[109] S. Xiao, Z. Liu, P. Zhang, and N. Muennighoff, *C-pack: packaged resources to advance general chinese embedding*, 2023. arXiv: 2309.07597 [cs.CL].

[110] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding", *in Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 4171–4186. DOI: 10.18653/v1/N19-1423. Accessed: Oct. 18, 2021.

[111] A. Zareian, S. Karaman, and S.-F. Chang. "Bridging Knowledge Graphs to Generate Scene Graphs". arXiv: 2001.02314 [cs], Accessed: May 26, 2023, preprint.

[112] M. Neau, P. E. Santos, A.-G. Bosser, A. Macvicar, and C. Buche, "Mining informativeness in scene graphs: prioritizing informative relations in scene graph generation for enhanced performance in applications", *Pattern Recognition Letters*, 2025, ISSN: 0167-8655. DOI: https://doi.org/10.1016/j.patrec.2025.01.008. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S016786552500008X.

[113] D. Paulius and Y. Sun, "A Survey of Knowledge Representation in Service Robotics", en, *Robotics and Autonomous Systems*, vol. 118, pp. 13–30, Aug. 2019, ISSN: 09218890.

[114] T. Chen, W. Yu, R. Chen, and L. Lin, "Knowledge-Embedded Routing Network for Scene Graph Generation", *in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 6163–6171. Accessed: Jul. 5, 2024.

[115] J.-H. Kim, J. Jun, and B.-T. Zhang, "Bilinear Attention Networks", *in Advances in Neural Information Processing Systems*, vol. 31, Curran Associates, Inc., 2018. Accessed: Jul. 10, 2024.

[116] J. Li, D. Li, S. Savarese, and S. Hoi, "BLIP-2: Bootstrapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models", arXiv, 2023. DOI: 10.48550/ARXIV.2301.12597. Accessed: Jul. 11, 2024.

[117] S. Schuster, R. Krishna, A. Chang, L. Fei-Fei, and C. D. Manning, "Generating Semantically Precise Scene Graphs from Textual Descriptions for Improved Image Retrieval", *in Proceedings of the Fourth Workshop on Vision and Language*, A. Belz, L. Coheur, V. Ferrari, M.-F. Moens, K. Pastra, and I. VuliÄ‡, Eds., Lisbon, Portugal: Association for Computational Linguistics, Sep. 2015, pp. 70–80. DOI: 10.18653/v1/W15-2812. Accessed: Jul. 11, 2024.

[118] Z. Li et al., *FACTUAL: A Benchmark for Faithful and Consistent Textual Scene Graph Parsing*, Jun. 2023. arXiv: 2305.17497 [cs]. Accessed: Jul. 11, 2024.

[119] P. Anderson, B. Fernando, M. Johnson, and S. Gould, "SPICE: Semantic Propositional Image Caption Evaluation", *in Computer Vision - ECCV 2016*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds., Cham: Springer International Publishing, 2016, pp. 382–398, ISBN: 978-3-319-46454-1. DOI: 10.1007/978-3-319-46454-1_24.

[120] M. Honnibal, I. Montani, S. Van Landeghem, A. Boyd, et al., "Spacy: industrial-strength natural language processing in python", 2020.

[121] S. R. Bowman, G. Angeli, C. Potts, and C. D. Manning, "A large annotated corpus for learning natural language inference", *in Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, L. MÃ rquez, C. Callison-Burch, and J. Su, Eds., Lisbon, Portugal: Association for Computational Linguistics, Sep. 2015, pp. 632–642. DOI: 10.18653/v1/D15-1075. Accessed: Jul. 16, 2024.

[122] D. Cer, M. Diab, E. Agirre, I. Lopez-Gazpio, and L. Specia, "SemEval-2017 Task 1: Semantic Textual Similarity Multilingual and Crosslingual Focused Evaluation", *in Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, S. Bethard, M. Carpuat, M. Apidianaki, S. M. Mohammad, D. Cer, and D. Jurgens, Eds., Vancouver, Canada: Association for Computational Linguistics, Aug. 2017, pp. 1–14. DOI: 10.18653/v1/S17-2001. Accessed: Jul. 16, 2024.

[123] Y. Liu et al., "Roberta: a robustly optimized bert pretraining approach", *arXiv preprint arXiv:1907.11692*, 2019.

[124] K. Song, X. Tan, T. Qin, J. Lu, and T.-Y. Liu, *MPNet: Masked and Permuted Pre-training for Language Understanding*, Nov. 2020. arXiv: 2004.09297 [cs]. Accessed: Jul. 17, 2024.

[125] P. Zhang, S. Xiao, Z. Liu, Z. Dou, and J.-Y. Nie, *Retrieve Anything To Augment Large Language Models*, Oct. 2023. DOI: 10.48550/arXiv.2310.07554. arXiv: 2310.07554 [cs]. Accessed: Sep. 22, 2024.

[126] A. Radford et al., "Learning Transferable Visual Models From Natural Language Supervision", *in Proceedings of the 38th International Conference on Machine Learning*, PMLR, Jul. 2021, pp. 8748–8763. Accessed: Jul. 16, 2024.

[127] W. H. Kruskal and W. A. Wallis, "Use of ranks in one-criterion variance analysis", *Journal of the American statistical Association*, vol. 47, *260*, pp. 583–621, 1952.

[128] H. Keselman and J. C. Rogan, "The tukey multiple comparison test: 1953–1976.", *Psychological Bulletin*, vol. 84, *5*, p. 1050, 1977.

[129] N. Kucharczuk, T. WÄ… s, and O. Skibski, "PageRank for Edges: Axiomatic Characterization", *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, *5*, pp. 5108–5115, Jun. 2022, ISSN: 2374-3468. DOI: 10.1609/aaai.v36i5.20444. Accessed: Dec. 9, 2023.

[130] L. Page, S. Brin, R. Motwani, and T. Winograd, "The pagerank citation ranking: bringing order to the web.", Stanford infolab, Tech. Rep., 1999.

[131] M. Girvan and M. E. J. Newman, "Community structure in social and biological networks", *Proceedings of the National Academy of Sciences*, vol. 99, *12*, pp. 7821–7826, Jun. 2002. DOI: 10.1073/pnas.122653799. Accessed: Jul. 12, 2024.

[132] L. C. Freeman, "A Set of Measures of Centrality Based on Betweenness", *Sociometry*, vol. 40, *1*, pp. 35–41, 1977, ISSN: 0038-0431. DOI: 10.2307/3033543. JSTOR: 3033543. Accessed: Jul. 12, 2024.

[133] X. Yang et al., "Transforming Visual Scene Graphs to Image Captions", *in Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, A. Rogers, J. Boyd-Graber, and N. Okazaki, Eds., Toronto, Canada: Association for Computational Linguistics, Jul. 2023, pp. 12 427–12 440. DOI: 10.18653/v1/2023.acl-long.694. Accessed: Jun. 24, 2024.

[134] B. Knyazev, H. De Vries, C. Cangea, G. W. Taylor, A. Courville, and E. Belilovsky, "Generative Compositional Augmentations for Scene Graph Prediction", *in 2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, Montreal, QC, Canada: IEEE, Oct. 2021, pp. 15 807–15 817, ISBN: 978-1-66542-812-5. DOI: 10.1109/ICCV48922.2021.01553. Accessed: Jun. 24, 2024.

[135] D. A. Hudson and C. D. Manning, "GQA: A New Dataset for Real-World Visual Reasoning and Compositional Question Answering", *in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 6700–6709. Accessed: Jun. 24, 2024.

[136] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models", *in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 10 684–10 695.

[137] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "Gans trained by a two time-scale update rule converge to a local nash equilibrium", *Advances in neural information processing systems*, vol. 30, 2017.

[138] T. Jin et al., "Independent Relationship Detection for Real-Time Scene Graph Generation", *in Neural Information Processing*, M. Tanveer, S. Agarwal, S. Ozawa, A. Ekbal, and A. Jatowt, Eds., ser. Communications in Computer and Information Science, Singapore: Springer Nature, 2023, pp. 106–118, ISBN: 978-981-9916-39-9. DOI: 10.1007/978-981-99-1639-9_9.

[139] K. He, G. Gkioxari, P. Dollar, and R. Girshick, "Mask R-CNN", *in Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2961–2969. Accessed: Jul. 31, 2024.

[140] T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature Pyramid Networks for Object Detection", *in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2117–2125. Accessed: Jul. 31, 2024.

[141] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, *6*, pp. 1137–1149, Jun. 2017, ISSN: 1939-3539. DOI: 10.1109/TPAMI.2016.2577031.

[142] Y. Teng and L. Wang, "Structured Sparse R-CNN for Direct Scene Graph Generation", *in 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, New Orleans, LA, USA: IEEE, Jun. 2022, pp. 19 415–19 424, ISBN: 978-1-66546-946-3. DOI: 10.1109/CVPR52688.2022.01883. Accessed: Aug. 21, 2023.

[143] Y. Cong, M. Y. Yang, and B. Rosenhahn. "RelTR: Relation Transformer for Scene Graph Generation". version 2. arXiv: 2201.11460 [cs], Accessed: Oct. 2, 2022, preprint.

[144] R. Li, S. Zhang, and X. He, "SGTR: End-to-end Scene Graph Generation with Transformer", *in 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, New Orleans, LA, USA: IEEE, Jun. 2022, pp. 19 464–19 474, ISBN: 978-1-66546-946-3. DOI: 10.1109/CVPR52688.2022.01888. Accessed: Oct. 2, 2022.

[145] S. Xie, R. Girshick, P. Dollar, Z. Tu, and K. He, "Aggregated Residual Transformations for Deep Neural Networks", *in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1492–1500. Accessed: Jul. 30, 2024.

[146] X. Lyu et al., "Fine-Grained Predicates Learning for Scene Graph Generation", presented at the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 19 467–19 475. Accessed: Sep. 19, 2023.

[147] A. K. Menon, S. Jayasumana, A. S. Rawat, H. Jain, A. Veit, and S. Kumar, "Long-tail learning via logit adjustment", *in International Conference on Learning Representations*.

[148] Z. Chen, S. Rezayi, and S. Li, "More Knowledge, Less Bias: Unbiasing Scene Graph Generation With Explicit Ontological Adjustment", *in Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2023, pp. 4023–4032. Accessed: Aug. 23, 2024.

[149] Y. Guo et al., "From General to Specific: Informative Scene Graph Generation via Balance Adjustment", *in Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 16 383–16 392. Accessed: Aug. 23, 2024.

[150] N. H. Chapman, F. Dayoub, W. Browne, and C. Lehnert, "Predicting Class Distribution Shift for Reliable Domain Adaptive Object Detection", *IEEE Robotics and Automation Letters*, vol. 8, *8*, pp. 5084–5091, Aug. 2023, ISSN: 2377-3766. DOI: 10.1109/LRA.2023.3290420. Accessed: Aug. 5, 2024.

[151] Y. Ou, L. Mi, and Z. Chen, "Object-Relation Reasoning Graph for Action Recognition", *in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 20 133–20 142. Accessed: Sep. 2, 2022.

[152] B. Chen, K. Marussy, S. Pilarski, O. SemerÃ¡th, and D. Varro, "Consistent Scene Graph Generation by Constraint Optimization", *in Proceedings of the 37th IEEE/ACM International Conference on Automated Software Engineering*, Rochester MI USA: ACM, Oct. 2022, pp. 1–13, ISBN: 978-1-4503-9475-8. DOI: 10.1145/3551349.3560433. Accessed: Aug. 2, 2023.

[153] M. Riand, P. Le Callet, and L. Dollé, "Rethinking Scene Graphs for Action Recognition", *in 2023 IEEE International Conference on Visual Communications and Image Processing (VCIP)*, Dec. 2023, pp. 1–5. DOI: 10.1109/VCIP59821.2023.10402749. Accessed: Sep. 16, 2024.

[154] A. M. Zanchettin, "Symbolic representation of what robots are taught in one demonstration", *Robotics and Autonomous Systems*, vol. 166, p. 104 452, Aug. 2023, ISSN: 0921-8890. DOI: 10.1016/j.robot.2023.104452. Accessed: Sep. 13, 2024.

[155] J. F. Allen, "Maintaining knowledge about temporal intervals", *Communications of the ACM*, vol. 26, *11*, pp. 832–843, Nov. 1983, ISSN: 0001-0782, 1557-7317. DOI: 10.1145/182.358434. Accessed: Jan. 24, 2025.

[156] M. Beetz, L. MÃ¶senlechner, and M. Tenorth, "CRAM - A Cognitive Robot Abstract Machine for everyday manipulation in human environments", *in 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct. 2010, pp. 1012–1017. DOI: 10.1109/IROS.2010.5650146. Accessed: Jan. 27, 2025.

[157] J. McCarthy and P. Hayes, "Some Philosophical Problems from the Standpoint of Artificial Intelligence", *in Readings in Artificial Intelligence*, Elsevier, 1969, pp. 431–450, ISBN: 978-0-934613-03-3. DOI: 10.1016/B978-0-934613-03-3.50033-7. Accessed: Jun. 3, 2022.

[158] M. Shanahan, "The Event Calculus Explained", *in Artificial Intelligence Today: Recent Trends and Developments*, M. J. Wooldridge and M. Veloso, Eds., Berlin, Heidelberg: Springer, 1999, pp. 409–430, ISBN: 978-3-540-48317-5. DOI: 10.1007/3-540-48317-9_17. Accessed: Jan. 26, 2025.

[159] E. Erdem, M. Gelfond, and N. Leone, "Applications of Answer Set Programming", *AI Magazine*, vol. 37, *3*, pp. 53–68, Oct. 2016, ISSN: 2371-9621. DOI: 10.1609/aimag.v37i3.2678. Accessed: Jan. 27, 2025.

[160] S. Manzoor et al., "Ontology-Based Knowledge Representation in Robotic Systems: A Survey Oriented toward Applications", *Applied Sciences*, vol. 11, *10*, p. 4324, Jan. 2021, ISSN: 2076-3417. DOI: 10.3390/app11104324. Accessed: Oct. 2, 2022.

[161] A. Saxena, A. Jain, O. Sener, A. Jami, D. K. Misra, and H. S. Koppula, *RoboBrain: Large-Scale Knowledge Engine for Robots*, Apr. 2015.

[162] T. Nanayakkara et al., "Kimera: From SLAM to spatial perception with 3D dynamic scene graphs", *Int. J. Rob. Res.*, vol. 40, *12-14*, pp. 1510–1546, Dec. 2021, ISSN: 0278-3649. DOI: 10.1177/02783649211056674. Accessed: Jan. 27, 2025.

[163] J. Strader, N. Hughes, W. Chen, A. Speranzon, and L. Carlone, "Indoor and Outdoor 3D Scene Graph Generation Via Language-Enabled Spatial Ontologies", *IEEE Robotics and Automation Letters*, vol. 9, *6*, pp. 4886–4893, Jun. 2024, ISSN: 2377-3766. DOI: 10.1109/LRA.2024.3384084. Accessed: Jan. 27, 2025.

[164] K. Ramirez-Amaro, M. Beetz, and G. Cheng, "Transferring skills to humanoid robots by extracting semantic representations from observations of human activities", *Artificial Intelligence*, Special Issue on AI and Robotics, vol. 247, pp. 95–118, Jun. 2017, ISSN: 0004-3702. DOI: 10.1016/j.artint.2015.08.009. Accessed: Jan. 24, 2025.

[165] X. Shang, T. Ren, J. Guo, H. Zhang, and T.-S. Chua, "Video Visual Relation Detection", *in Proceedings of the 25th ACM International Conference on Multimedia*, Mountain View California USA: ACM, Oct. 2017, pp. 1300–1308, ISBN: 978-1-4503-4906-2. DOI: 10.1145/3123266.3123380. Accessed: Sep. 2, 2022.

[166] R. Wang, Z. Wei, P. Li, Q. Zhang, and X. Huang, "Storytelling from an Image Stream Using Scene Graphs", *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, *05*, pp. 9185–9192, Apr. 2020, ISSN: 2374-3468. DOI: 10.1609/aaai.v34i05.6455. Accessed: Nov. 4, 2021.

[167] Y. Li, X. Yang, and C. Xu, "Dynamic Scene Graph Generation via Anticipatory Pre-Training", *in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 13 874–13 883. Accessed: Sep. 2, 2022.

[168] J. Cao, J. Pang, X. Weng, R. Khirodkar, and K. Kitani, "Observation-Centric SORT: Rethinking SORT for Robust Multi-Object Tracking", *in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 9686–9696. Accessed: Sep. 10, 2024.

[169] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple online and realtime tracking", *in 2016 IEEE International Conference on Image Processing (ICIP 2016)*, IEEE, Institute of Electrical and Electronics Engineers, Aug. 2016, pp. 3464–3468. DOI: 10.1109/ICIP.2016.7533003. Accessed: Sep. 11, 2024.

[170] R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems", *Journal of Basic Engineering*, vol. 82, *1*, pp. 35–45, Mar. 1960, ISSN: 0021-9223. DOI: 10.1115/1.3662552. Accessed: Sep. 11, 2024.

[171] H. W. Kuhn, "The Hungarian method for the assignment problem", *Naval Research Logistics Quarterly*, vol. 2, *1-2*, pp. 83–97, 1955, ISSN: 1931-9193. DOI: 10.1002/nav.3800020109. Accessed: Sep. 11, 2024.

[172] F. Battiston, V. Nicosia, and V. Latora, "Structural measures for multiplex networks", *Physical Review E*, vol. 89, *3*, p. 032 804, Mar. 2014, ISSN: 1539-3755, 1550-2376. DOI: 10.1103/PhysRevE.89.032804. arXiv: 1308.3182 [physics]. Accessed: Nov. 29, 2023.

[173] T. Zhuo, Z. Cheng, P. Zhang, Y. Wong, and M. Kankanhalli, *Explainable Video Action Reasoning via Prior Knowledge and State Transitions*, Aug. 2019. arXiv: 1908.10700 [cs]. Accessed: Mar. 27, 2024.

[174] H. Xu, A. Das, and K. Saenko, "R-C3D: Region Convolutional 3D Network for Temporal Activity Detection", *in Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 5783–5792. Accessed: Sep. 13, 2024.

[175] G. A. Sigurdsson, G. Varol, X. Wang, A. Farhadi, I. Laptev, and A. Gupta, *Hollywood in Homes: Crowdsourcing Data Collection for Activity Understanding*, Jul. 2016. arXiv: 1604.01753 [cs]. Accessed: Jun. 28, 2022.

[176] K. Kahatapitiya, Z. Ren, H. Li, Z. Wu, M. S. Ryoo, and G. Hua, "Weakly-guided self-supervised pretraining for temporal activity detection", *in Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence and Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence and Thirteenth Symposium on Educational Advances in Artificial Intelligence*, ser. AAAI'23/IAAI'23/EAAI'23, vol. 37, AAAI Press,

Feb. 2023, pp. 1078–1086, ISBN: 978-1-57735-880-0. DOI: 10.1609/aaai.v37i1.25189. Accessed: Sep. 13, 2024.

[177] F. Semeraro, A. Griffiths, and A. Cangelosi, "Human-robot collaboration and machine learning: A systematic review of recent research", *Robotics and Computer-Integrated Manufacturing*, vol. 79, p. 102 432, Feb. 2023, ISSN: 0736-5845. DOI: 10.1016/j.rcim.2022.102432. Accessed: Sep. 13, 2024.

[178] M. Diehl, C. Paxton, and K. Ramirez-Amaro, "Automated Generation of Robotic Planning Domains from Observations", *in 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sep. 2021, pp. 6732–6738. DOI: 10.1109/IROS51168.2021.9636781. Accessed: Jan. 19, 2025.

[179] T. R. Savarimuthu et al., "Teaching a Robot the Semantics of Assembly Tasks", *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 48, *5*, pp. 670–692, May 2018, ISSN: 2168-2232. DOI: 10.1109/TSMC.2016.2635479. Accessed: Sep. 22, 2024.

[180] S. Macenski, T. Foote, B. Gerkey, C. Lalancette, and W. Woodall, "Robot Operating System 2: Design, architecture, and uses in the wild", *Science Robotics*, vol. 7, *66*, eabm6074, May 2022. DOI: 10.1126/scirobotics.abm6074. Accessed: Jan. 24, 2025.

[181] G. Vaquette, A. Orcesi, L. Lucat, and C. Achard, "The DAily Home LIfe Activity Dataset: A High Semantic Activity Dataset for Online Recognition", *in 2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017)*, May 2017, pp. 497–504. DOI: 10.1109/FG.2017.67. Accessed: Sep. 13, 2024.

[182] S. Hassan, G. Mujtaba, A. Rajput, and N. Fatima, "Multi-object tracking: a systematic literature review", *Multimedia Tools and Applications*, vol. 83, *14*, pp. 43 439–43 492, 2024.

[183] E. Yadollahi et al., "Explainability for human-robot collaboration", *in Companion of the 2024 ACM/IEEE International Conference on Human-Robot Interaction*, 2024, pp. 1364–1366.

[184] X. Chang, T. Wang, S. Cai, and C. Sun. "LANDMARK: Language-guided Representation Enhancement Framework for Scene Graph Generation". arXiv: 2303.01080 [cs], Accessed: Apr. 25, 2023, preprint.

[185] C. Chen, Y. Zhan, B. Yu, L. Liu, Y. Luo, and B. Du. "Resistance Training using Prior Bias: toward Unbiased Scene Graph Generation". arXiv: 2201.06794 [cs], Accessed: Mar. 17, 2023, preprint.

[186] Y. Deng et al., "Hierarchical Memory Learning for Fine-Grained Scene Graph Generation", *in* vol. 13687, 2022, pp. 266–283. DOI: 10.1007/978-3-031-19812-0_16. arXiv: 2203.06907 [cs]. Accessed: Oct. 23, 2023.

[187] Y. Min, A. Wu, and C. Deng, "Environment-Invariant Curriculum Relation Learning for Fine-Grained Scene Graph Generation", *in 2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, Paris, France: IEEE, Oct. 1, 2023, pp. 13 250–13 261, ISBN: 9798350307184. DOI: 10.1109/ICCV51070.2023.01223. Accessed: Apr. 19, 2024.

[188] X. Dong, T. Gan, X. Song, J. Wu, Y. Cheng, and L. Nie, "Stacked Hybrid-Attention and Group Collaborative Learning for Unbiased Scene Graph Generation", *in 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, New Orleans, LA, USA: IEEE, Jun. 2022, pp. 19 405–19 414, ISBN: 978-1-66546-946-3. DOI: 10.1109/CVPR52688.2022.01882. Accessed: Jan. 17, 2023.

[189] K. Ye and A. Kovashka, "Linguistic Structures as Weak Supervision for Visual Scene Graph Generation", *in 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Nashville, TN, USA: IEEE, Jun. 2021, pp. 8285–8295, ISBN: 978-1-66544-509-2. DOI: 10.1109/CVPR46437.2021.00819. Accessed: Sep. 20, 2022.

[190] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a Method for Automatic Evaluation of Machine Translation", *in Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, P. Isabelle, E. Charniak, and D. Lin, Eds., Philadelphia, Pennsylvania, USA: Association for Computational Linguistics, Jul. 2002, pp. 311–318. DOI: 10.3115/1073083.1073135. Accessed: Sep. 21, 2024.

[191] S. Banerjee and A. Lavie, "METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments", *in Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, J. Goldstein, A. Lavie, C.-Y. Lin, and C. Voss, Eds., Ann Arbor, Michigan: Association for Computational Linguistics, Jun. 2005, pp. 65–72. Accessed: Sep. 21, 2024.

[192] R. Vedantam, C. Lawrence Zitnick, and D. Parikh, "CIDEr: Consensus-Based Image Description Evaluation", *in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 4566–4575. Accessed: Sep. 21, 2024.

[193] T. He, L. Gao, J. Song, and Y.-F. Li, "Towards Open-Vocabulary Scene Graph Generation with Prompt-Based Finetuning", *in Computer Vision - ECCV 2022*, S. Avidan, G. Brostow, M. CissÃ©, G. M. Farinella, and T. Hassner, Eds., Cham: Springer Nature Switzerland, 2022, pp. 56–73, ISBN: 978-3-031-19815-1. DOI: 10.1007/978-3-031-19815-1_4.

[194] R. Li, S. Zhang, D. Lin, K. Chen, and X. He, "From Pixels to Graphs: Open-Vocabulary Scene Graph Generation with Vision-Language Models", *in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 28 076–28 086. Accessed: Sep. 21, 2024.

[195] T. Cheng, L. Song, Y. Ge, W. Liu, X. Wang, and Y. Shan, *YOLO-World: Real-Time Open-Vocabulary Object Detection*, Feb. 2024. DOI: 10.48550/arXiv.2401.17270. arXiv: 2401.17270 [cs]. Accessed: Jun. 3, 2024.

[196] S. Liu et al., *Grounding DINO: Marrying DINO with Grounded Pre-Training for Open-Set Object Detection*, Jul. 2024. DOI: 10.48550/arXiv.2303.05499. arXiv: 2303.05499 [cs]. Accessed: Sep. 23, 2024.

[197] B. Jiang, Z. Zhuang, and C. J. Taylor. "Enhancing Scene Graph Generation with Hierarchical Relationships and Commonsense Knowledge". arXiv: 2311.12889 [cs], Accessed: Feb. 29, 2024, preprint.

[198] M. J. Khan, J. G. Breslin, and E. Curry, "Expressive Scene Graph Generation Using Commonsense Knowledge Infusion for Visual Understanding and Reasoning", *in The Semantic Web*, P. Groth et al., Eds., vol. 13261, Cham: Springer International Publishing, 2022, pp. 93–112, ISBN: 978-3-031-06980-2 978-3-031-06981-9. Accessed: Sep. 8, 2022.

[199] D. Paulius and Y. Sun, "A Survey of Knowledge Representation in Service Robotics", *Robotics and Autonomous Systems*, vol. 118, pp. 13–30, Aug. 2019, ISSN: 09218890. DOI: 10.1016/j.robot.2019.03.005. Accessed: Jun. 27, 2022.

[200] F. Liu, F. Yan, L. Zheng, C. Feng, Y. Huang, and L. Ma, *RoboUniView: Visual-Language Model with Unified View Representation for Robotic Manipulation*, Sep. 2024. arXiv: 2406.18977 [cs]. Accessed: Sep. 23, 2024.

[201] J. Gao et al., "Physically Grounded Vision-Language Models for Robotic Manipulation", *in 2024 IEEE International Conference on Robotics and Automation (ICRA)*, May 2024, pp. 12 462–12 469. DOI: 10.1109/ICRA57147.2024.10610090. Accessed: Sep. 23, 2024.

[202] C. Buche, M. Neau, T. Ung, L. Li, S. Wang, and C. L. Bono, "Robocup@ home sspl champion 2023: robobreizh, a fully embedded approach", *in Robot World Cup*, Springer, 2023, pp. 374–385.

[203] T. Wisspeintner, T. van der Zant, L. Iocchi, and S. Schiffer, "RoboCup@Home: Scientific Competition and Benchmarking for Domestic Service Robots", *Interaction Studies*, vol. 10, *3*, pp. 392–426, Jan. 2009, ISSN: 1572-0373, 1572-0381. DOI: 10.1075/is.10.3.06wis. Accessed: Sep. 15, 2024.

[204] M. Neau, P. Santos, A.-G. Bosser, N. Beu, and C. Buche, *Commonsense Reasoning for Identifying and Understanding the Implicit Need of Help and Synthesizing Assistive Actions*, Feb. 2022. arXiv: 2202.11337 [cs]. Accessed: Sep. 23, 2024.

[205] M. Sap et al., "ATOMIC: An Atlas of Machine Commonsense for If-Then Reasoning", en, *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 3027–3035, Jul. 2019, ISSN: 2374-3468, 2159-5399.

[206] A. Chiatti, *Visually intelligent agents: improving sensemaking in service robotics*. Open University (United Kingdom), 2022.

[207] C.-Y. Wang, I.-H. Yeh, and H.-Y. M. Liao, *YOLOv9: Learning What You Want to Learn Using Programmable Gradient Information*, Feb. 2024. DOI: 10.48550/arXiv.2402.13616. arXiv: 2402.13616 [cs]. Accessed: Sep. 29, 2024.

[208] A. Wang et al., *YOLOv10: Real-Time End-to-End Object Detection*, May 2024. arXiv: 2405.14458 [cs]. Accessed: May 25, 2024.

[209] Y. Zhao et al., "DETRs Beat YOLOs on Real-time Object Detection", *in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 16 965–16 974. Accessed: Sep. 29, 2024.

[210] X. Han, J. Yang, H. Hu, L. Zhang, J. Gao, and P. Zhang, *Image Scene Graph Generation (SGG) Benchmark*, Jul. 2021. DOI: 10.48550/arXiv.2107.12604. arXiv: 2107.12604 [cs]. Accessed: Sep. 29, 2024.

[211] H. Zhang, Z. Kyaw, S.-F. Chang, and T.-S. Chua, "Visual Translation Embedding Network for Visual Relation Detection", *in 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI: IEEE, Jul. 2017, pp. 3107–3115, ISBN: 978-1-5386-0457-1. DOI: 10.1109/CVPR.2017.331. Accessed: Dec. 21, 2022.

[212] Y. Li, W. Ouyang, B. Zhou, K. Wang, and X. Wang, "Scene Graph Generation from Objects, Phrases and Region Captions", *in 2017 IEEE International Conference on Computer Vision (ICCV)*, Venice: IEEE, Oct. 2017, pp. 1270–1279, ISBN: 978-1-5386-1032-9. DOI: 10.1109/ICCV.2017.142. Accessed: Nov. 3, 2021.

[213] J. Yang, J. Lu, S. Lee, D. Batra, and D. Parikh, "Graph R-CNN for Scene Graph Generation", *in Computer Vision - ECCV 2018*, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds., vol. 11205, Cham: Springer International Publishing, 2018, pp. 690–706, ISBN: 978-3-030-01245-8 978-3-030-01246-5. DOI: 10.1007/978-3-030-01246-5_41. Accessed: Nov. 4, 2021.

[214] J. Zhang, K. J. Shih, A. Elgammal, A. Tao, and B. Catanzaro, "Graphical Contrastive Losses for Scene Graph Parsing", *in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 11 535–11 543. Accessed: Sep. 29, 2024.

[215] Z. Wang, X. Xu, G. Wang, Y. Yang, and H. T. Shen, "Quaternion Relation Embedding for Scene Graph Generation", *IEEE Transactions on Multimedia*, pp. 1–12, 2023, ISSN: 1941-0077. DOI: 10.1109/TMM.2023.3239229.